ABBYY®

ABBYY® Real-Time Recognition SDK®

Cordova Plugin

Table of Contents

Introduction

ABBYY Real-Time Recognition SDK Cordova Plugin allows to use the Text Capture and Data Capture features of ABBYY Real-Time Recognition SDK in apps based on the Apache Cordova framework.

This plugin requires the ABBYY Real-Time Recognition SDK native library which is available for Android and iOS. You can download its free version from the ABBYY RTR SDK website. Extended versions of the native libraries are also available, with more recognition languages, more Data Capture features, and full Text Capture scenario support. If you are interested in the extended version, please contact ABBYY sales.

This manual describes the **AbbyyRtrSdk** JavaScript module. More information, including the library usage details, is available in the *ABBYY Real-Time Recognition SDK 1 Developer's Guide* found in the library download packages.

Getting Started

To start developing with the ABBYY Real-Time Recognition SDK Cordova Plugin, you need to add target platforms and the plugin to your project, and copy the ABBYY Real-Time Recognition SDK assets and native libraries for Android and iOS, as described below.

1. Create a project:

```
cordova create path/to/ProjectDir com.example.abbyyrtrsdk MyRTRProject
```

2. Change to the project directory to add platforms and the plugin:

```
cd path/to/ProjectDir
cordova platform add ios
cordova platform add android
cordova plugin add cordova-plugin-abbyy-rtr-sdk
```

3. Add ABBYY Real-Time Recognition SDK files to the following project subdirectories:
   a. Copy RTR SDK assets (patterns and dictionaries) and your license file (**AbbyyRtrSdk.license**) to **www/rtr_assets**.
   b. Copy the Android library (**abbyy-rtr-sdk-1.0.aar**) to **libs/android**.
   c. Copy the iOS framework (**AbbyyRtrSDK.framework**) to **libs/ios**.
4. Add **libs/android** and **libs/ios** to the linker search paths.
   a. For Android, add the following settings to **platforms/android/build.gradle**:

```
allprojects {
  repositories {
    flatDir {
      dirs '../../libs/android'
    }
  }
}
```

   b. For iOS, add the following settings to **platforms/ios/cordova/build.xcconfig**:

```
FRAMEWORK_SEARCH_PATHS = "../../libs/ios"
```

5. Use **AbbyyRtrSdk** module methods to add text and data capture functionality to your app.

To build and run your app for android, do:

```
cordova build android
cordova run android
```

When building your app for iOS, specify the DEVELOPMENT_TEAM build flag:

```
cordova build ios --buildFlag="DEVELOPMENT_TEAM=<YOUR_TEAM>"
cordova run ios --buildFlag="DEVELOPMENT_TEAM=<YOUR_TEAM>"
```

AbbyyRtrSdk module

ABBYY Real-Time Recognition SDK plugin module.

Methods

Name	Description
startTextCapture	Opens a modal dialog for the Text Capture scenario.
startDataCapture	Opens a modal dialog for the Data Capture scenario.

startTextCapture method of AbbyyRtrSdk module

Opens a modal dialog with controls for the Text Capture scenario.

```
AbbyyRtrSdk.startTextCapture(callback, options)
```

Parameters

callback

The callback function which receives the text capture operation result. Your callback should expect a single JSON object as an argument (see Result).

options

JSON object specifying text capture parameters (see Options).

Options

The table below describes the JSON object that you can pass as the *options* argument to change text capture settings. All keys are optional. Omitting a key means that a default setting will be used.

Key	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **www/rtr_assets/** directory in your project. Default: "AbbyyRtrSdk.license".

Key	Value type	Description
selectableRecognitionLanguages	string array	Recognition languages which can be selected by the user, for example: ["English", "French", "German"]. Empty array disables language selection. ***Note:*** *For the list of supported languages and their identifiers, see Specifications — Available Languages in the ABBYY Real-Time Recognition SDK 1 Developer's Guide.* Default: [] (empty array, language selection disabled).
recognitionLanguages	string array	Recognition language selected by default. Default: ["English"].
areaOfInterest	string	Width and height of the recognition area, separated by a whitespace — for example: "0.8 0.3". The area of interest is centered in the preview frame, its width and height are relative to the preview frame size and should be in the [0.0, 1.0] range. Default: "0.8 0.3" (intended to capture a few lines of text in portrait mode).
stopWhenStable	boolean	Whether to stop the plugin as soon as the result status is "Stable" (see Result status). When enabled (true), the recognition process can be stopped automatically. When disabled (false), recognition can be stopped only manually by user. Default: true (automatic stop enabled).
isFlashlightVisible	boolean	Show (true) or hide (false) the flashlight button in the text capture dialog. Default: true (flashlight available).
isStopButtonVisible	boolean	Show (true) or hide (false) the stop button in the text capture dialog. When the user taps stop, RTR SDK returns the latest recognition result. Default: true (manual stop available).

Result

This section describes the JSON object that represents text recognition results. The callback you implement should parse this object and show results to user.

Key	Value type	Description
textLines	object array	An array of objects representing recognized lines of text. These objects have the following keys: • **text** (string): the recognized text. • **quadrangle** (string): vertex coordinates of the bounding quadrangle, a string of 8 integers separated with whitespaces ("x1 y1 ... x4 y4"), goes clockwise starting from the bottom left. • **rect** (string): position and size of the bounding rectangle, a string of 4 integers separated with whitespaces ("x y width height"). ***Note:*** *The **rect** key exists in the iOS version only.* If an error occurs during processing, the **textLines** key is not present in the result.
resultInfo	object	Additional information. This object has the following keys: • **stabilityStatus** (string): result stability status. See Result status for details. • **userAction** (string): the user's action which stopped the plugin, if any. Can be "Manually Stopped" if the stop button has been used, and "Canceled" if the user canceled processing. If the plugin has stopped automatically, the **userAction** key is not present in **resultInfo**. • **frameSize** (string): full size of the preview frame, a string with 2 integers separated with a whitespace ("720 1280"). • **recognitionLanguages** (string array): languages used for recognition, the array contains language identifiers (["English", "French", "German"]).
error	object	Error details. This key is present only if an error occurs. The value is an object which has a single key: • **description** (string): human-readable error description.

Below is an example of a result JSON when text capture succeeds.

```
{
  textLines : [
    {
      text : "Welcome to ABBYY RTR SDK",
      quadrangle : "100 780 100 600 620 600 620 780",
      rect : "100 600 520 180"
```

```
    }
  ],
  resultInfo : {
    stabilityStatus : "Available",
    userAction : "Manually Stopped",
    frameSize : "720 1280",
    recognitionLanguages : ["English", "French"]
  }
}
```

On error, you will receive a JSON which does not contain recognized text but provides an error message.

```
{
  error : {
    description : "Something has gone wrong."
  },
  resultInfo : {
    stabilityStatus : "Tentative",
    userAction : "Canceled",
    frameSize : "720 1280",
    recognitionLanguages : ["English", "French"]
  }
}
```

startDataCapture method of AbbyyRtrSdk module

Opens a modal dialog with controls for the Data Capture scenario.

```
AbbyyRtrSdk.startDataCapture(callback, options)
```

Parameters

callback

The callback function which receives the text capture operation result. Your callback should expect a single JSON object as an argument (see Result).

options

JSON object specifying data capture parameters (see Options).

Options

The table below describes the JSON object that you can pass as the *options* argument to change data capture settings. All keys are optional. Omitting a key means that a default setting will be used, except the **profile** and **customDataCaptureScenario** keys: you must specify either one of them, but not both at the same time.

Key	Value type	Description
profile	string	The predefined data capture profile to use, for example: "MRZ". **Note:** *For the list of supported documents, see Specifications — Data Capture Profiles in the ABBYY Real-Time Recognition SDK 1 Developer's Guide.*
customDataCaptureScenario	object	Custom data capture settings. This object has the following keys: • **name** (string): the name of your custom data capture scenario, required. • **description** (string): a more detailed description. This key is optional; if not given, it will be assigned the same value as **name**. • **recognitionLanguages** (string array): recognition languages to use. Default is ["English"]. • **fields** (object array): describes data fields to capture. Each object in this array has a single **regEx** key; its value is a string containing the regular expression that should be matched when capturing a field. **Note:** *For details, see the How to Capture a Custom Data Field guide in the ABBYY Real-Time Recognition SDK 1 Developer's Guide.* **Note:** *Currently a custom data capture profile supports one recognized field only (**fields** must contain only 1 element).* See below for an example of a custom data capture scenario definition.
licenseFileName	string	The name of the license file. This file must be located in the **www/rtr_assets/** directory in your project. Default: "AbbyyRtrSdk.license".
areaOfInterest	string	Width and height of the recognition area, separated by a whitespace — for example: "0.8 0.3". The area of interest is centered in the preview frame, its width and height are relative to the preview frame size and should be in the [0.0, 1.0] range. Default: "0.8 0.3" (intended to capture a few lines of text in portrait mode).
stopWhenStable	boolean	Whether to stop the plugin as soon as the result status is "Stable" (see Result status). When enabled (true), the

Key	Value type	Description
		recognition process can be stopped automatically. When disabled (false), recognition can be stopped only manually by user. Default: true (automatic stop enabled).
isFlashlightVisible	boolean	Show (true) or hide (false) the flashlight button in the text capture dialog. Default: true (flashlight available).
isStopButtonVisible	boolean	Show (true) or hide (false) the stop button in the text capture dialog. When the user taps stop, RTR SDK returns the latest recognition result. Default: true (manual stop available).

An example of a custom text capture scenario definition in the options JSON.

```
{
  ...

  customDataCaptureScenario : {
    name : "Code",
    description : "Alphanumeric code, for example: X6YZ64 32VPA zyy777.",
    recognitionLanguages : ["English"],
    fields : [ {
      regEx : "([a-zA-Z]+[0-9]+|[0-9]+[a-zA-Z]+)[0-9a-zA-Z]*"
    } ]
  },

  ...

}
```

Result

This section describes the JSON object that represents data recognition results. The callback you implement should parse this object and show results to user.

Key	Value type	Description
dataScheme	object	The data scheme which was applied during data capture. The value is an object which has two keys: • **id** (string): the internal scheme identifier.

Key	Value type	Description
		• **name** (string): the scheme name. If you had defined a custom data capture scenario in options, both the **id** and **name** will be the same as the scenario name you specified. If you selected a predefined profile, the **id** and **name** are specified by the profile. If an error occurs during processing, the **dataScheme** key is not present in the result.
dataFields	object array	Recognized data fields. Each object in the array represents a separate data field. The data field objects have the following keys: • **id** (string): the internal identifier of the field. • **name** (string): the field name. Similarly to **dataScheme**, in custom scenarios both **id** and **name** are the same as the scenario name you specified (currently custom scenarios allow only 1 recognized field). • **text** (string): full text of the field. • **quadrangle** (string): vertex coordinates of the bounding quadrangle, a string of 8 integers separated with whitespaces ("x1 y1 ... x4 y4"), goes clockwise starting from the bottom left. • **components** (object array): an array of objects representing field components, that is, the text fragments found on the image, which constitute the field. In the **components** array each element is an object with the following keys: • **text** (string): text of this fragment. • **quadrangle** (string): vertex coordinates of the bounding quadrangle of this fragment, similar to the field's quadrangle. • **rect** (string): position and size of the bounding rectangle, a string of 4 integers separated with whitespaces ("x y width height"). ***Note:*** *The **rect** key exists in the iOS version only.* If an error occurs during processing, the **dataFields** key is not present in the result.
resultInfo	object	Additional information. This object has the following keys: • **stabilityStatus** (string): result stability status. See Result status for details. • **userAction** (string): the user's action which stopped the plugin, if any. Can be "Manually Stopped" if the stop button has been used, and "Canceled" if the user canceled processing. If the plugin has stopped

Key	Value type	Description
		automatically, the **userAction** key is not present in **resultInfo**. • **frameSize** (string): full size of the preview frame, a string with 2 integers separated with a whitespace ("720 1280").
error	object	Error details. This key is present only if an error occurs. The value is an object which has a single key: • **description** (string): human-readable error description.

Below is an example of a result JSON when data capture succeeds.

```
{
  dataScheme : {
    id : "Hello",
    name : "Hello"
  },
  dataFields : [
    {
      id : "Hello",
      name : "Hello",
      text : "Hello world!",
      quadrangle : "100 780 100 600 620 600 620 780",
      components : [
        {
          text : "Hello",
          quadrangle : "100 780 100 600 350 600 350 780",
          rect : "100 600 250 180"
        },
        {
          text : "world!",
          quadrangle : "360 780 360 600 620 600 620 780",
          rect : "360 600 260 180"
        }
      ]
    }
  ],
  resultInfo : {
    stabilityStatus : "Available",
    userAction : "Manually Stopped",
    frameSize : "720 1280"
  }
}
```

On error, you will receive a JSON which does not contain recognized data but provides an error message.

```
{
  error : {
    description : "Something unexpected have happened."
  }
  resultInfo : {
```

```
    stabilityStatus : "Tentative",
    userAction : "Canceled",
    frameSize : "720 1280"
  }
}
```

Result status

Result status in ABBYY Real-Time Recognition SDK is an estimate of how stable the result is, and whether it is likely to be improved by recognizing more frames. It is not recommended to use the recognition result in any way if its status is below "Available".

Name	Description
NotReady	No content available.
Tentative	Content detected on a single frame.
Verified	Content verified: matching content found in at least two frames.
Available	Matching content found in three or more frames. The content is recognized and the result is available, though the result can still vary with the addition of new frames.
TentativelyStable	The result has been stable in the last two frames.
Stable	The result has been stable in the last three or more frames.