

# Mobile SDK

## Atalasoft MobileImage HTML5 SDK Developer's Guide

Version: 3.0.0

Date: 2016-06-14

©2013- 2016 Atalasoftware, 116 Pleasant St, Suite 321, Easthampton, MA 01027, U.S.A. All rights reserved.  
Use is subject to license terms.

Third-party software is copyrighted and licensed from Atalasoftware's suppliers.

THIS SOFTWARE CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF ATALASOFTWARE, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF KOFAX.

Atalasoftware, the Atalasoftware logo, and the Atalasoftware product names stated herein are trademarks or registered trademarks of Atalasoftware, Inc. in the U.S. and other countries. All other trademarks are the trademarks or registered trademarks of their respective owners. U.S. Government Rights Commercial software. Government users are subject to the Kofax. standard license agreement and applicable provisions of the FAR and its supplements.

You agree that you do not intend to and will not, directly or indirectly, export or transmit the Software or related documentation and technical data to any country to which such export or transmission is restricted by any applicable U.S. regulation or statute, without the prior written consent, if required, of the Bureau of Export Administration of the U.S. Department of Commerce, or such other governmental entity as may have jurisdiction over such export or transmission. You represent and warrant that you are not located in, under the control of, or a national or resident of any such country.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

# Table of Contents

<b>Preface</b> .....	<b>4</b>
Getting Help for Atalasoft Products .....	4
<b>The HTML5 SDK</b> .....	<b>5</b>
Introduction .....	5
Using the HTML SDK With Other HTML 5 Applications .....	5
HTML5 SDK External Classes .....	5
KfxWebSDK.Capture (Atalasoft MobileImage) .....	6
KfxWebSDK.ReviewController (Atalasoft ReviewControl) .....	8
KfxWebSDK Image Processor .....	9
KfxWebSDK.Utilities (Atalasoft Utilities) .....	11
JSON Definitions .....	11
Features and Limitations .....	12
Supported Devices .....	13
Supported Browsers .....	13
Limitations in the SDK .....	14
Coding Example HTML5 SDK .....	14

# Preface

This guide includes the information you need to successfully integrate HTML5 SDK components into your mobile project.

For additional details on API library properties and settings, refer to the HTML5 SDK API Reference Guide.

## Getting Help for Atalasoftware Products

Atalasoftware regularly updates the Atalasoftware Support site with the latest information about Atalasoftware products.

To access some resources, you must have a valid Support Agreement with an authorized Atalasoftware Reseller/Partner or with Atalasoftware directly.

Use the tools that Atalasoftware provides for researching and identifying issues. For example, use the Atalasoftware Support site to search for answers about messages, keywords, and product issues. To access the Atalasoftware Support page, go to <http://www.atalasoftware.com/support>.

The Atalasoftware Support page provides:

- Product information and release news
- Downloadable product documentation
- Access to product knowledge bases
- Access to support cases
- Direct contact information

Use these tools to find answers to questions that you have, to learn about new functionality, and to research possible solutions to current issues.

## Chapter 1

# The HTML5 SDK

## Introduction

This document is intended to provide a brief overview of the usage and features of the HTML5 SDK.

## Using the HTML SDK With Other HTML 5 Applications

In order to create a new HTML5 application and use/integrate HTML5 SDK the app developer needs to follow the below instructions.

1. Create an HTML5 application.
2. Include the SDK .css file in the application HTML files.  
Add the following code there: `<link rel="stylesheet" href="../../KfxWebSDK/CSS/KfxWebSDK.css">`. Be sure to change the path to `KfxWebSDK.css` according to your configuration (SDK location).
3. Include SDK java script minified file. Add the following code there: `<script src="../../KfxWebSDK/KfxWebSDK.js"></script>`. Be sure to change the path to `KfxWebSDK.js` according to your configuration (SDK location). This file contains all necessary 3<sup>rd</sup> party libraries, so there is no need to worry about any SDK dependencies.

**Note** Do not move or rename anything in the SDK folder.

There are several directories in the SDK main folder (`KfxWebSDK`) such as the CSS, Resources, Images, and so on. Do not change the directory structure of the HTML5 SDK and do not rename the files. Doing so may break the SDK.

4. To ensure the SDK content is loaded successfully, or to debug any issues, please use the Web Developer Tools and console. You can find this view in most popular browsers. You can also debug remotely on a device. Please refer to the browser's user guide. For example, here is the link to a description of the Chrome remote debugging process: <https://developer.chrome.com/devtools/docs/remote-debugging>.

## HTML5 SDK External Classes

HTML5 SDK has four external classes:

- `KfxWebSDK.Capture`
- `KfxWebSDK.DocumentExtractor`

- KfxWebSDK.ReviewControl
- KfxWebSDK.Utilities
- KfxWebSDK.ImageProcessor

The following sections describe these classes in detail.

## KfxWebSDK.Capture (Atalasoft MobileImage)

This class provides methods to capture a document either from a camera or photo library. It enhances the user experience by adding feedback while the user captures a document. This guidance makes it easier to capture high quality images.

### Native

Package name: `com.kofax.capture`

Global Namespace: KfxWebSDK

Class Name: Capture

### JavaScript Closure

KfxWebSDK.Capture

### APIs

API	Parameters	Description
create	options successCallBack errorCallBack	<p>Creates a Capture control based on given options. It will always use the rear camera.</p> <p><code>Options.containerId</code>: Empty divId, where the application developer wants to see a camera preview along with capture guidance. The div container must exist and be empty, otherwise an error will be thrown. The application developer has to properly set the size and position of the div. The SDK doesn't check the size and position or any other container css properties, this is a developer responsibility.</p> <p><code>Options.preference</code>: camera/gallery, from where the developer would like to capture a document.</p> <p><code>Options.preview</code>: Boolean value representing whether or not to review the captured image using the SDK review control. In case of FALSE, the developer needs to implement its own review functionality. This option effects only web capture, when the captured image is from the gallery via the native camera there is no review screen available.</p> <p><code>Options.videoStream</code>: Boolean representing to follow either the standard capture or document capture process.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p><b>Note</b> The requirement to choose the gallery is a limitation in both Android and iPhone. Camera only is a limitation in iPhone.</p> </div>

API	Parameters	Description
setOptions	options successCallback errorCallback	Sets various capture criteria. <pre>{     frameAspectRatio: 0.628,     framePadding: 5,     frameCornerHeight: 15,     frameCornerWidth: 70,     frameCornerColor: '#00FF00',     outOfFrameTransparency: 0.5,     showEdges: false,     edgesColor: '#FFFF00',     edgesWidth: '4',     guidanceSize: 150,     criteria: {         captureTimeout: 1700         centerToleranceFraction: 0.15         maxFillFraction: 1.8         minFillFraction: 0.65     },     lookAndFeel: {         documentSample: 'http://example.com /images/document_sample.jpg',         forceCapture: 10,         gallery: true     } }</pre>
getOptions	successCallback errorCallback	Returns current capture control options for capture criteria, capture guidance messages and other configurable ui options. successCallback: callback with JSON object representing capture control options. errorCallback: callback with error message to be invoked when something goes wrong.
getDefault Options	successCallback errorCallback	Returns default capture control options for capture criteria, capture guidance messages, and other configurable UI options. successCallback: callback with JSON object representing capture control options. errorCallback: callback with error message to be invoked when something goes wrong.
takePicture	successCallback errorCallback	Starts the Auto Capture process successCallback: callback with uint8 representation of the captured image. errorCallback: callback with the error message to be invoked when something goes wrong.
takePicture Continually	successCallback errorCallback	Starts Continuous Auto Capture process. successCallback: callback with uint8 representation of the captured image. errorCallback: callback with the error message to be invoked when something goes wrong.
forceTake Picture	successCallback errorCallback	Captures document while ignoring capture criteria. successCallback: callback with uint8 representation of the captured image. errorCallback: callback with the error message to be invoked when something goes wrong.

API	Parameters	Description
stopCapture	successCallback errorCallback	Stops the capturing of images (works both in single capture and continuous capture).  successCallback: callback with no data. errorCallback: callback with the error message to be invoked when something goes wrong.
destroy	None	Cleans up internal the resources allocated by the create API call. Capturing must be stopped by the stopCapture API call before using destroy.

## Example Code Snippet

```
//Initialize Capture singleton to work with video capturing
KfxWebSDK.Capture.create({
  useVideoStream: true,
  containerId: 'ID_CAMERA_DIV',
  preview: false
}, function() {
  console.info('Done');
},
function(e) {
  console.info(e);
});

//Invokes method 'takePicture' on the singleton
KfxWebSDK.Capture.takePicture(function() {}, function(e) { console.info(e);});
```

## KfxWebSDK.ReviewController (Atalasoftware ReviewControl)

The Review Control has APIs used to create a review screen with Accept and Retake buttons.

This can optionally be used by the developer to manage the reviewing process. The ReviewControl is also embedded in the Capture module and can be enabled by setting options.preview to TRUE.

### Native

Global Namespace: KfxWebSDK

Class Name: ReviewControl

### JavaScript Closure

KfxWebSDK.ReviewController

This class contains methods you can use to create a review screen and set the accept - retake buttons handler.



## APIs

API	Parameters	Description
ReviewControl	containerId	<p>Creates a review screen entity with the canvas and toolbar with Accept &amp; Retake buttons.</p> <p>Options.containerId: divId, where the developer wants to see a review screen. The div container must exist, otherwise an error will be thrown. The developer has to properly set the size and position of the div. The SDK doesn't check size and position or any other container css properties; this is a developer responsibility. The div container can be either empty or not. If the container is not empty, the review control will hide all nested child elements until Accept or Retake is pressed.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p><b>Note</b> This method (constructor) just creates the entity and prepares html elements. The review screen is not shown after this call.</p> </div>
review	imageData, acceptCallback, retakeCallback	<p>Show the review screen with the imageData and the toolbar with Accept &amp; Retake buttons.</p> <p>ImageData: image to be reviewed. This image is expected to have valid dimensions.</p> <p>acceptCallback: callback to be invoked when the user press accept button.</p> <p>retakeCallback: callback to be invoked when the user press retake button.</p>

## Example Code Snippet

```
//Call to show review screen
var reviewControl = new KfxWebSDK.ReviewControl(options.containerId;
reviewControl.review(imageData, acceptCallback, retakeCallback);
```

## KfxWebSDK Image Processor

This class provides methods to convert an image to Bitonal, crop, scale, and setDPI.

API	Parameters	Description
bitonal	imageData successCallback errorCallback	<p>Processes the image into a bitonal image.</p> <p>imageData: the raw bytes of the image typically from a canvas, for example: context.getImageData().</p> <p>successCallback: this would contain the resulting bitonal image.</p> <p>errorCallback: this would contain the appropriate error messages.</p>

API	Parameters	Description
autoCrop	imageData options successCallback errorCallback	<p>Crops, deskews (and performs rectangularization if needed) on the input image after detecting the edges of the document.</p> <p>imageData: the raw bytes of the image typically from a canvas, for example ex: context.getImageData().</p> <p>Options.type - One of the following values:</p> <ul style="list-style-type: none"> <li>▪ KfxWebSDK.document.MOBILE_ID: 0</li> <li>▪ KfxWebSDK.document.CHECK_DEPOSIT: 1</li> <li>▪ KfxWebSDK.document.BILL_PAY: 2</li> </ul> <p>For now, only the MOBILE_ID type is supported. Other types are reserved for possible future use and research.</p> <p>The Type option helps the edge detector choose optimal processing parameters and defines the aspect ratio of the original document. If a check or bill type is specified, the success callback will return the original input image.</p> <p>successCallback : this would contain cropped image</p> <p>errorCallback: this would contain the appropriate error messages</p>
scale	imageData options successCallback errorCallback	<p>Scales the input image as per the specified scaleMegapixel value in the options.</p> <p>imageData: the raw bytes of the image typically from a canvas, for example: context.getImageData()</p> <p>options: a JSON object for scaleMegapixels. Currently only one option (scaleMegapixels) is supported. The image is scaled to the specified megapixel value (width * height)</p> <p>For example, var options = {scaleMegapixels: 1.2}.</p> <p>The scaleMegapixels value should generally be greater than zero and should be only used to downscale the image and not upscale. If the scaleMegapixels value higher than the input image size is given, the original image is returned without any scaling. No error is thrown in this case.</p> <p>successCallback: this would contain the scaled image</p> <p>errorCallback: this would contain the appropriate error messages</p>
setDPI	imageData dpi successCallback errorCallback	<p>Update the dpi for an image.</p> <p>imageData: the raw bytes of the image typically from a canvas ex: context.getImageData()</p> <p>dpi: dpi value which we want to update for an image</p> <p>successCallback: this would contain jpeg binary as dataurl</p> <p>errorCallback: this would contain the appropriate error messages</p>

### Example Code Snippet

```

KfxWebSDK.ImageProcessor.autoCrop(image, {
    type: KfxWebSDK.document.MOBILE_ID
}),
function(imageData) {

```

```

        // Do something with image data here
    }, function(e) {
        console.info(e);
    });

```

## KfxWebSDK.Utilities (Atalasoftware Utilities)

Utilities contains the API to check if Web capture is supported or not, depending on the browser type and device model. Developers can use it to decide what capture create options to use.

### Native

Global Namespace: KfxWebSDK

Class Name: Utilities

### JavaScript Closure

KfxWebSDK.Utilities

This singleton class contains the method you should use to check if web capture supported. The first call may be slower, but once the result is returned, it is cached and subsequent calls return the cached result.

### APIs

API	Parameters	Description
supportsAdvancedCapture	None	Checks browser and device model support for Web capture. This is useful for checking compatibility for the advanced document detection based capture experience. Based on this, the developer can configure capture experience options while creating a capture control.
GetDefaultOptions	successCallback errorCallback	Returns default capture control options for capture criteria, capture guidance messages, and other configurable UI options.  successCallback: callback with JSON object representing capture control options. errorCallback: callback with error message to be invoked when something goes wrong.

## Example Code Snippet

```

//Call to make bitonal image
if(!KfxWebSDK.Utilities.supportsAdvancedCapture()){
    doStandardCapture();
}else{
    $.mobile.navigate( "Capture.html",{transition:"none"} );
}

```

## JSON Definitions

The following sections provide definitions and examples of the JSON data used by this API.

## Capture Set Options

The following JSON definitions consist of options to set for the Capture module.

```
{
  frameAspectRatio: 0.628,
  framePadding: 5,
  frameCornerHeight: 15,
  frameCornerWidth: 70,
  frameCornerColor: '#00FF00',
  outOfFrameTransparency: 0.5,
  showEdges: false,
  edgesColor: '#FFFF00',
  edgesWidth: '4',

  criteria: {
    captureTimeout: 1700
    centerToleranceFraction: 0.15
    maxFillFraction: 1.8
    minFillFraction: 0.65
  },

  lookAndFeel: {
    documentSample: 'http://example.com/images/document_sample.jpg',
    forceCapture: 10,
    gallery: true
  }
}
```

## Features and Limitations

The HTML5 SDK includes the following features.

### Capture

The user captures a document through this feature. The capturing of a document can be done via the HTML5 SDK camera, device camera or the user can select an already existing image from the photo library.

HTML5 SDK capture enhances the user experience by giving feedback while the user captures a document. This feedback requires browser WebRTC support.

The following options can be set by the user:

- **Aspect Ratio:**

The aspect ratio defines the rectangular frame that should match the document being searched. It defines aspect ratio used to draw the Static frame with respect to the effective camera area.

- **Padding:**

Sets the minimum padding (in percent of length) to be maintained between the edges of the target frame and the image.

- **Demo Image:**

The demo image acts as a sample image that can be captured.

- **Manual Capture Time:**

After the set time (in seconds) has elapsed, the user is allowed to manually capture the image without regard to any capture constraints.

- **Use Video Frame:**  
A flag which allows the user to choose between advanced capture (HTML5 SDK camera) or standard capture (device camera or gallery).
- **Show edges:**  
When enabled shows the edges of the detected document.
- **Preference:**  
When advanced capture is turned off, you can choose between the gallery and the camera.

## Supported Devices

Advanced Capture only supports devices with a suitable camera resolution (1920(w) x 1080(h) and 1280(w) x 720(h)) are supported. Refer to the cross compatibility matrix at [www.kofax.com](http://www.kofax.com) for additional details.

All KfxWebSDK methods will report an error when used with an unsupported camera. See the API Reference guide for details on individual methods and their error handling.

Following is a list of devices currently known to be unsupported by Advanced Capture, despite the fact that these devices have a supported resolution. As a result, the method `{supportAdvanceCapture}` will report these devices as "unsupported":

- Samsung S2 (GT-I9100), S3, S4, S4 mini, S5, Galaxy Tab S
- Motorola Moto X 2nd Gen
- LG G2, G3
- HTC One M8, M9
- Nexus Nexus 4
- Sony Xperia Tablet Z

**Note** Although the above devices cannot be used with Advanced Capture, the native camera can still be used for the other types of capture.

## Supported Browsers

KfxWebSDK is targeted for mobile webkit based browsers. HTML5 features/specifications are slowly being adopted by most browsers, however as of now none of the browsers support all HTML5 features. Hence the degree of KfxWebSDK support varies from browser to browser.

The method `{supportsAdvancedCapture}` will allow a developer to check for browser and device support. For the `{Create}` method, that means a developer can choose to use Advanced Capture a.k.a Capture experience for supported browsers, or the device's native camera for unsupported/partially supported browsers. All KfxWebSDK methods will report an error when used with a partially supported browser. See the API Reference guide for details on how individual methods and their error handling.

Following is a list of browsers and which base versions KfxWebSDK are supported:

	Opera Mobile	iOS Safari	Android Browser	Firefox for Android	Chrome for Android
base version	30	7.1	4.4.4	38	47 and up

	Opera Mobile	iOS Safari	Android Browser	Firefox for Android	Chrome for Android
Partial Support	NO	YES	YES(4.1, 4.2, 4.3)	YES	NO
Full Support	YES	NO	YES	NO	YES

## Limitations in the SDK

1. As KfxWebSDK is part of the HTML5 framework, its support depends on underlying webkit HTML5 support and security permissions.
2. Choose gallery only is a limitation in both Android & iPhone.
3. Choose camera only is a limitation in iPhone.
4. Not comparable to the native SDK in 1.0
5. Developers must not rename minified SDK file KfxWebSDK.j
6. The supported browsers for iPhone and iPad which can load the captured image (either from the gallery or the native camera) into the image blob are iOS Safari 7.1, 8.1, 8.4, 9.0. For other versions, the user will not be able to see a preview of the captured image.
7. With Android devices, for the best HTML 5 Web capture experience, we recommend using Chrome version 47 or later.
8. SDK Guidance Capture is only supported over an HTTPS connection, and then only with supported browsers and devices. Native Capture will work with both HTTP and HTTPS connections.
9. As a general rule, do not attempt to capture documents that have been placed on a surface with complex patterns, shapes, or colors. A plain, contrasting surface is recommended.
10. For best results with HTML 5 Web capture, ensure that the background is simple and has a strong contrast with the document (for example white document on a black background). Also, there should be no glare and no shadows on the document itself.
11. The server IPP profile should always contain the following properties ( `_DocDimSmall_2.123_DocDimLarge_3.363`) which helps to resize images to 1009 \* 637 resolution for Acuant OCR. Without it some images may fail to be extracted. For example when cropping occurs on the client side, the IPP profile should be:

```
{
  "imageperfectionsettings": "_Do90DegreeRotation_4_DoScaleImageToDPI_300_DocDimSmall_2.123
  _DocDimLarge_3.363"
}
```

## Coding Example HTML5 SDK

The following section provides code snippets for the HTML5 SDK. For details on the classes, methods, parameters, and so on, refer to the reference guide that ships with the product.

### Initiate SDK Capture with Default Options

```
var cameraOptions = { containerId : "",
  preference : "camera",
  useVideoStream : true};

KfxWebSDK.Capture.create(cameraOptions, function(createSuccess){
```

```
KfxWebSDK.Capture.takePicture(function(imageData){
//success, user get the captured image in the ImageData format .
    },function(error){
        // error while taking the picture
    });
},function(error){
// error while creating the capture control
});
```

**containerId**

Specifies the DIV on which the camera will be launched.

**preference**

When advanced capture is turned off, you can choose between the gallery and the camera.

**useVideoStream**

A flag which allows the user to choose between advanced capture (HTML5 SDK camera) or standard capture (device camera or gallery).