Web Developer's Guide to scanner.js

Last updated on Oct, 2014

ALL RIGHTS RESERVED BY LAB ASPRISE! © 1998, 2014.

Table of Contents

1 INTRODUCTION 4

1.1 ABOUT SCANNER.JS 4

1.2 WHERE TO FIND SCANNER.JS 4

1.3 THE DEMO WEB PAGE 4

1.4 SCREENSHOT 5

2 SCANNER.JS IN ACTION 6

2.1 ADD SCANNER.JS SUPPORT 6

2.2 OPENS THE SCAN DIALOG 6

2.3 THE USER SCANS AND EDITS IMAGES 7

2.4 THE DIALOG RETURNS 9

2.5 UPLOAD THE IMAGES SCANNED THROUGH FORM 11

3 ADVANCED USAGES 13

3.1 SIMPLE SCAN 13

3.2 SCAN AND UPLOAD WITH THE SCAN DIALOG 13

3.3 SCAN AND UPLOAD AS PDF 14

3.4 SUPPORT FOR LEGACY BROWSERS 15

3.5 FORCE TO RETURN IMAGE/THUMBNAIL AS URLS INSTEAD OF BASE64 16

4 SUPPORT AND PROFESSIONAL SERVICES 17

4.1 SUPPORT WEB SITE 17

4.2 BASIC SUPPORT 17

4.3 PROFESSIONAL SERVICES 17

1 Introduction

1.1 About scanner.js

Bascanner.js enables any web page to acquire images from scanners using JavaScript in most desktop browsers like IE, Chrome, Firefox and more. In most cases, software install is not required and the user can enjoy a great scanning experience.

1.2 Where to find scanner.js

http://direct.asprise.com/scan/javascript/base/scanner.js

You may copy this file to any host including any CDN (content delivery network) if you have a valid license.

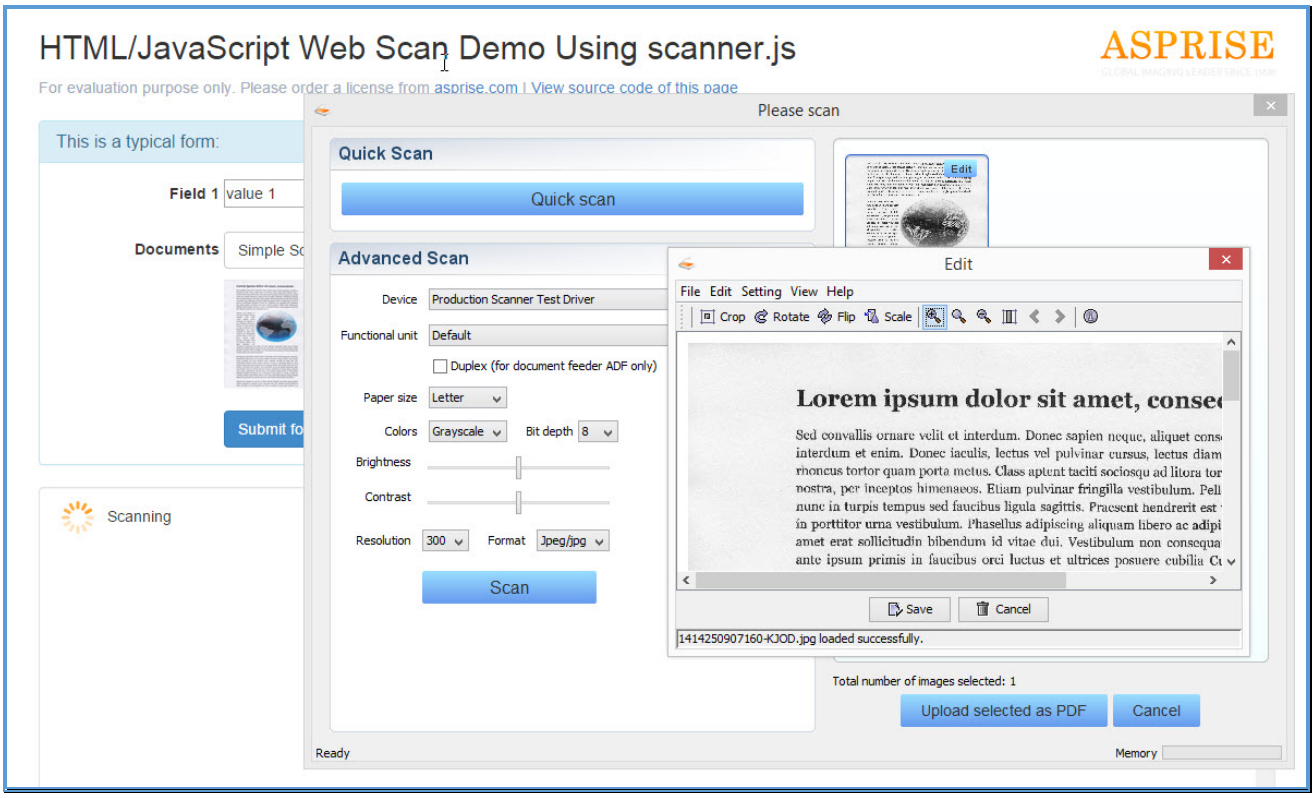
1.3 The demo web page

http://direct.asprise.com/scan/javascript/base/scanner.js-demo.html

Source code of the demo can be found at:

https://github.com/Asprise/scanner.js/blob/master/scanner.js-demo.html

1.4 Screenshot



2 scanner.js In Action

scanner.js allows any HTML/JavaScript developer to easily add document scanning, image editing and uploading functionality to add web page.

We'll use the demo web page to explain how it works. The complete source code of the demo page is available at:
https://github.com/Asprise/scanner.js/blob/master/scanner.js-demo.html

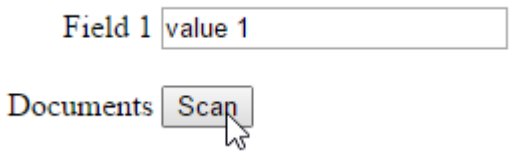
2.1 Add scanner.js support

All pages that need to access scanners must import scanner.js:

```
<script type="text/javascript"
src="http://direct.asprise.com/scan/javascript/base/scanner.js"></sc
ript>
```

2.2 Opens the Scan Dialog

The user clicks on a button, which triggers the *scan* function in JavaScript:



<button type="button" onclick="scan();">Scan</button>

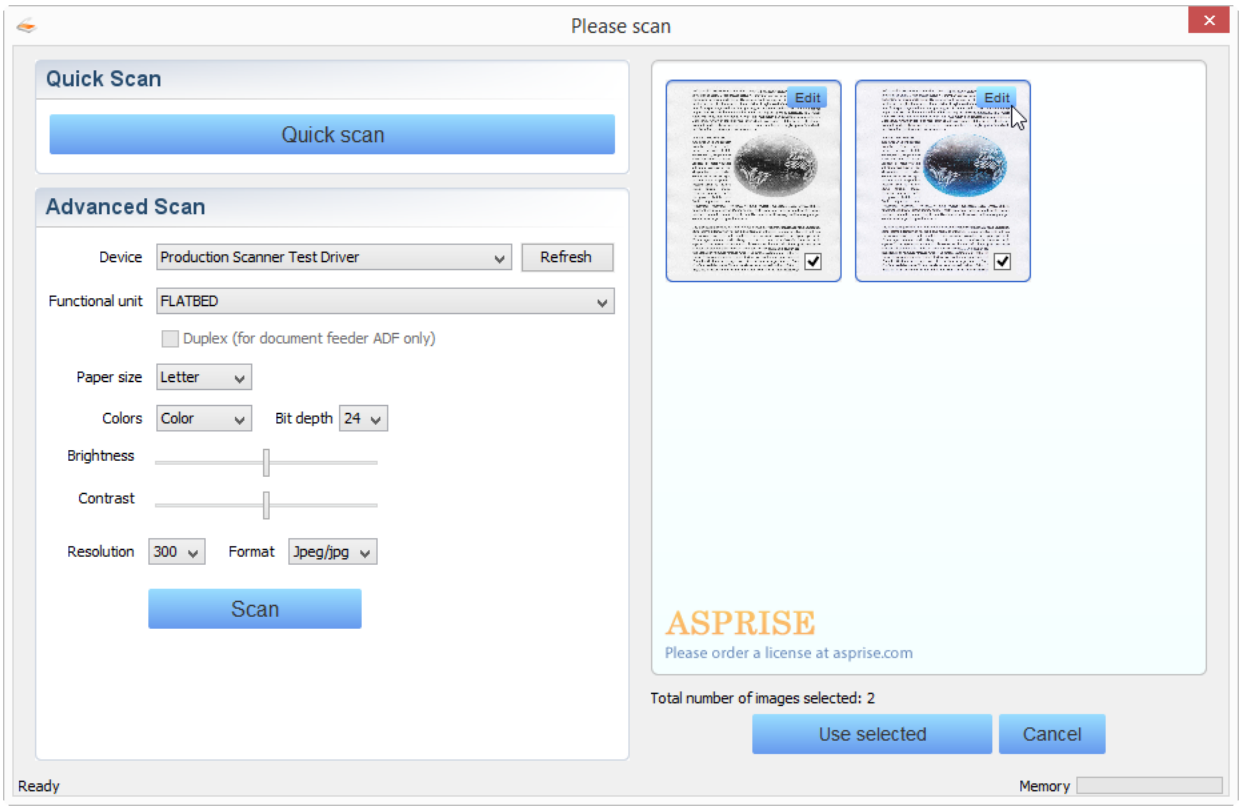
The JavaScript *scan* method opens the scan dialog:

```
function scan() {
  com_asprise_scan_request(myCallBackFunc, // callback on dialog closed
    com_asprise_scan_cmd_method_SCAN, // action: SCAN
    com_asprise_scan_cmd_return_IMAGES_AND_THUMBNAILS, // return types
    {'wia-version': 2} // options
  );
}
```

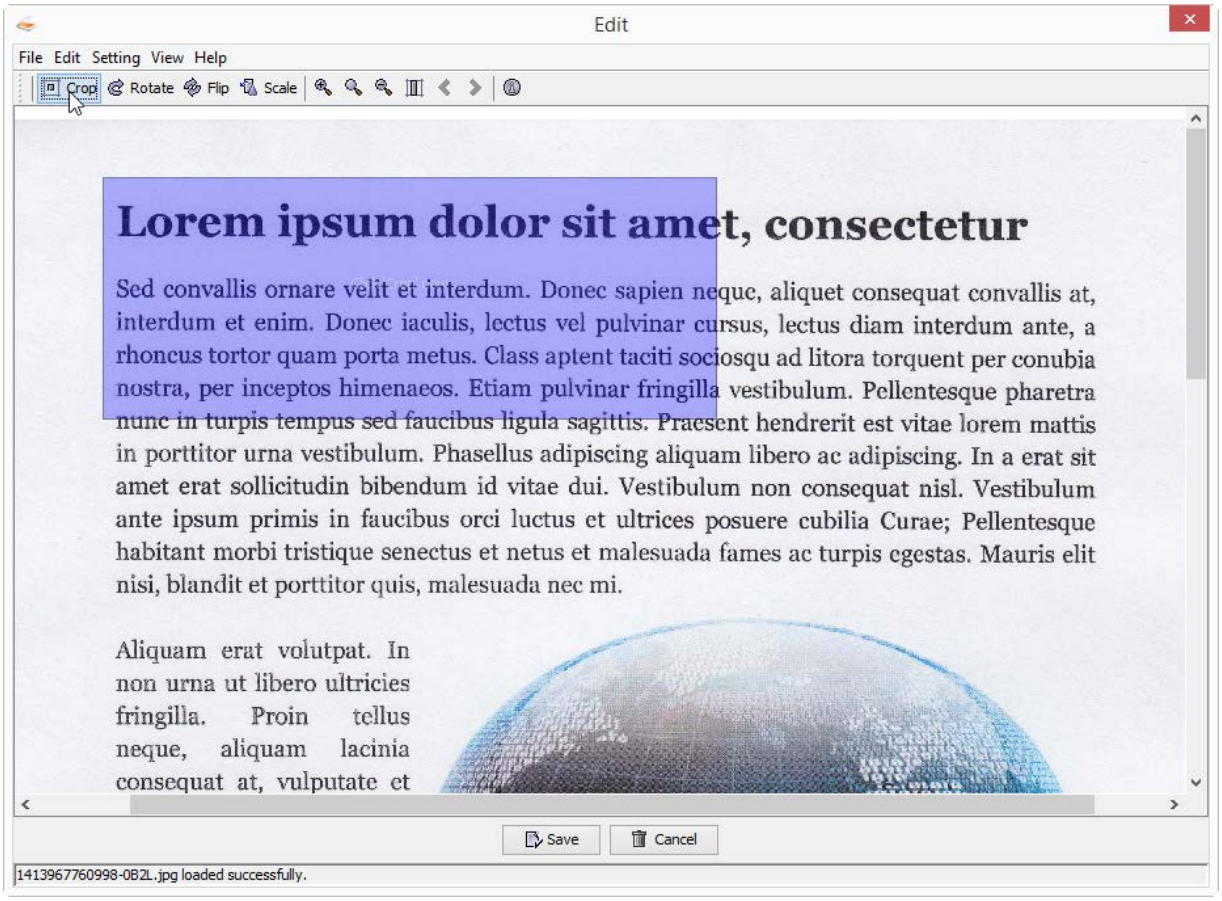
Note the above code specifies a callback function to accept images and/or thumbnails when the scan dialog closes. The source code of the callback function will be shown later.

2.3 The User Scans and Edits Images

The following dialog will appear and the user can then use 'Quick scan' or 'Scan' button to acquire images from scanners:



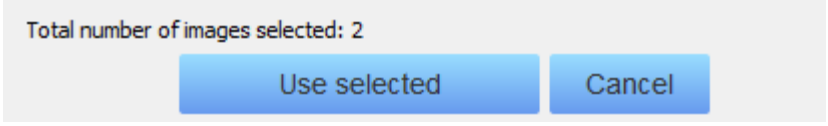
Once an image is acquired, the user may edit it by click on the 'Edit' button on the top left of the image thumbnail:



The following image editing actions are support: crop, rotate, flip and scale.

2.4 The Dialog Returns

The user may click 'Use selected' or 'Cancel' button to close the scan dialog:



Immediately after the dialog is closed, the callback function will be called:

```
function myCallBackFunc(success, mesg, thumbs, images) {
  for(var i = 0; (images instanceof Array) && i < images.length; i++){
    addImage(images[i], document.getElementById('images'));
  }
```

```
}

var imagesScanned = []; // global var to store all images scanned

function addImage(imgObj, domParent) {
  imagesScanned.push(imgObj);
  var imgSrc = imgObj.datatype == com_asprise_scan_datatype_BASE64 ?
    'data:' + imgObj.mimetype + '; base64,' + imgObj.data : imgObj.data;
  var elementImg = createDomElementFromModel({
    'name': 'img',
    'attributes': {
      'class': 'scanned',
      'src': imgSrc,
      'height': '100',
      'class': 'zoom'
    }
  });
  domParent.appendChild(elementImg);
}
```

The *addImage* function creates a new <img> element for every image scanned:

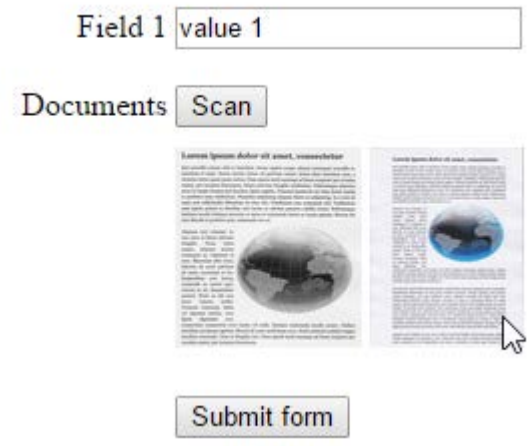


Image and thumbnail objects returned from the scan dialog have the following properties:

Name	Values	Remarks

datatype	com_asprise_scan_datatype_BASE64 com_asprise_scan_datatype_URL	Either base64 or url
mimetype	image/jpeg image/tiff application/pdf	
data	*The actual data of the datatype*	

The HTML code:

```
<form id="form1"
action="http://asprise.com/scan/applet/upload.php?action=dump"
method="POST" enctype="multipart/form-data" target="_blank">
    <label for="field1" style="display: inline-block; width: 150px;
text-align: right;">Field 1</label>
    <input id="field1" name="field1" type="text" value="value 1"/>
    <br><span style="height: 4px;"></span><br>

    <label style="display: inline-block; width: 150px; text-align:
right;">Documents</label>
    <button type="button" onclick="scan();">Scan</button>

    <div id="images" style="margin-left: 154px; margin-top: 10px;">
    </div>

    <input type="button" value="Submit form" onclick="submitForm1();"
style="margin-left: 154px; margin-top: 20px;">
</form>
```

2.5 Upload the images scanned through form

When the user clicks on the 'Submit form' button, the following JavaScript function will be invoked:

```
function submitForm1() {
  com_asprise_scan_submit_form_with_images('form1', imagesScanned,
    function(xhr) {
```

```
      if(xhr.readyState == 4) { // 4: request finished
        // todo process response: xhr.responseText
      }
    });
}
```

Remember **imagesScanned** contains all the images that have been scanned. All fields in the form with id 'form1' are submitted together with the scanned images.

3 Advanced Usages

3.1 Simple scan

When to use: In this mode, the Swing based dialog will not be shown and only a system UI scan will show up. Once the scan is done, the user does not have an opportunity to edit the image as it will be returned directly to the callback function.

How to use:

```
com_asprise_scan_request(myCallBackFunc,
  com_asprise_scan_cmd_method_SIMPLE_SCAN, // method: simple scan
  com_asprise_scan_cmd_return_IMAGES_AND_THUMBNAILS, null);
```

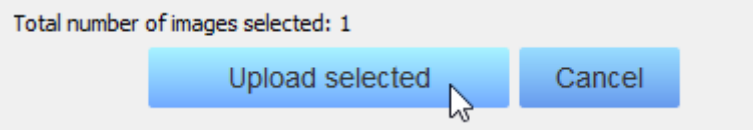
3.2 Scan and upload with the scan dialog

When to use: you need to have scanned images upload directly to a webserver or the user uses a legacy browser that does not support BASE64 as image source.

How to use:

```
com_asprise_scan_request(myCallBackFunc,
  com_asprise_scan_cmd_method_SCAN_THEN_UPLOAD, // scan and upload
  com_asprise_scan_cmd_return_IMAGES_AND_THUMBNAILS, {
   'upload-url': 'YOUR_URL_WHICH_CAN_BE_FORMED_DYNAMICALLY'
  }
);
```

The user will see the following buttons:



Once the user clicks on 'Upload selected', the response returned upon uploading will be available in the *mesg* parameter of the callback function:

```
function myCallBackFunc(success, mesg, thumbs, images)
```

Tips:

- Consider to return and display the URLs of the upload images so that the user can click to verify;
- You may request the dialog to return thumbnails to be displayed on your web page;
- You may use the following URL to test uploading: http://asprise.com/scan/applet/upload.php?action=upload (It will save any file posted and returns the URL as the response. The source code of this file is available at http://asprise.com/scan/applet/upload.php?action=source)

By default, images will be uploaded as JPG files.

3.3 Scan and upload as PDF

When to use: you need to have multiple images uploaded as a single PDF file to a web server.

How to use:

```
com_asprise_scan_request(myCallBackFunc,
```

```
  com_asprise_scan_cmd_method_SCAN_THEN_UPLOAD, // scan and upload
  com_asprise_scan_cmd_return_IMAGES_AND_THUMBNAILS, {
   'upload-url': 'YOUR_URL_WHICH_CAN_BE_FORMED_DYNAMICALLY',
   'format': 'PDF'
  }
);
```

3.4 Support for legacy browsers

When to use: you have customers running on legacy browsers, for example IE 7.

IE 7 does not support base64 as image source and IE 8 has a size limit for base64 data of 32kb. To support those legacy browsers, one can use an intermediate image server so that the scan dialog can upload images a to the image server and return URL of the images/thumbnails to JavaScript instead of base64.

How to use:

```
com_asprise_scan_request(myCallBackFunc,
  com_asprise_scan_cmd_method_SCAN_THEN_UPLOAD, // scan and upload
  com_asprise_scan_cmd_return_IMAGES_AND_THUMBNAILS, {
   'image-server-url': 'YOUR_URL_WHICH_CAN_BE_FORMED_DYNAMICALLY'
  }
);
```

If you do not specify one, the following URL will be used:

http://asprise.com/scan/applet/upload.php?action=upload

The source code of sample implementation of the image server is available at

http://asprise.com/scan/applet/upload.php?action=source

The image server accepts image files posted and output URLs in plain text with URLs separated by line break.

3.5 Force to return image/thumbnail as URLs instead of base64

When to use: you want to handle images as URLs instead of base64 in JavaScript.

For non-legacy browsers, images and thumbnails will be passed to JavaScript in base64 data format. If you prefer to get images and thumbnails as URLs, you can use the following options:

How to use:

```
com_asprise_scan_request(myCallBackFunc,
  com_asprise_scan_cmd_method_SCAN, // scan
  com_asprise_scan_cmd_return_IMAGES_AND_THUMBNAILS, {
   'image-server-url': 'YOUR_URL_WHICH_CAN_BE_FORMED_DYNAMICALLY',
   'preferred-datatype': 'url'
  }
);
```

If `image-server-url` is omitted, the following URL will be used:

http://asprise.com/scan/applet/upload.php?action=upload

4 Support and professional services

4.1 Support Web Site

http://www.asprise.com

4.2 Basic Support

Our team provides basic support for general Asprise products. Email your technical questions to support@asprise.com

Advice: You are strongly recommended to subscribe our premium support service in order to get your problems solved quickly.

4.3 Professional Services

Our team are ready to help you to develop various applications, components. Please send your query to sales@asprise.com