

## Ezy Barcode Reader Tutorial Guide

Before decoding any barcode document, ensure that you have selected the required barcode format(s) that you wish to search within the document by selecting the relevant checkboxes as provided within the sample Forms program, 'LB\_EBR\_LNF\_DB.fmb'. See below, for the list of supported **1D** and **2D** barcodes:

By default, '**Code 128**' is the only barcode format selected.

Should the required barcode format you are trying to decode is not found; ensure that you have its checkbox selected.

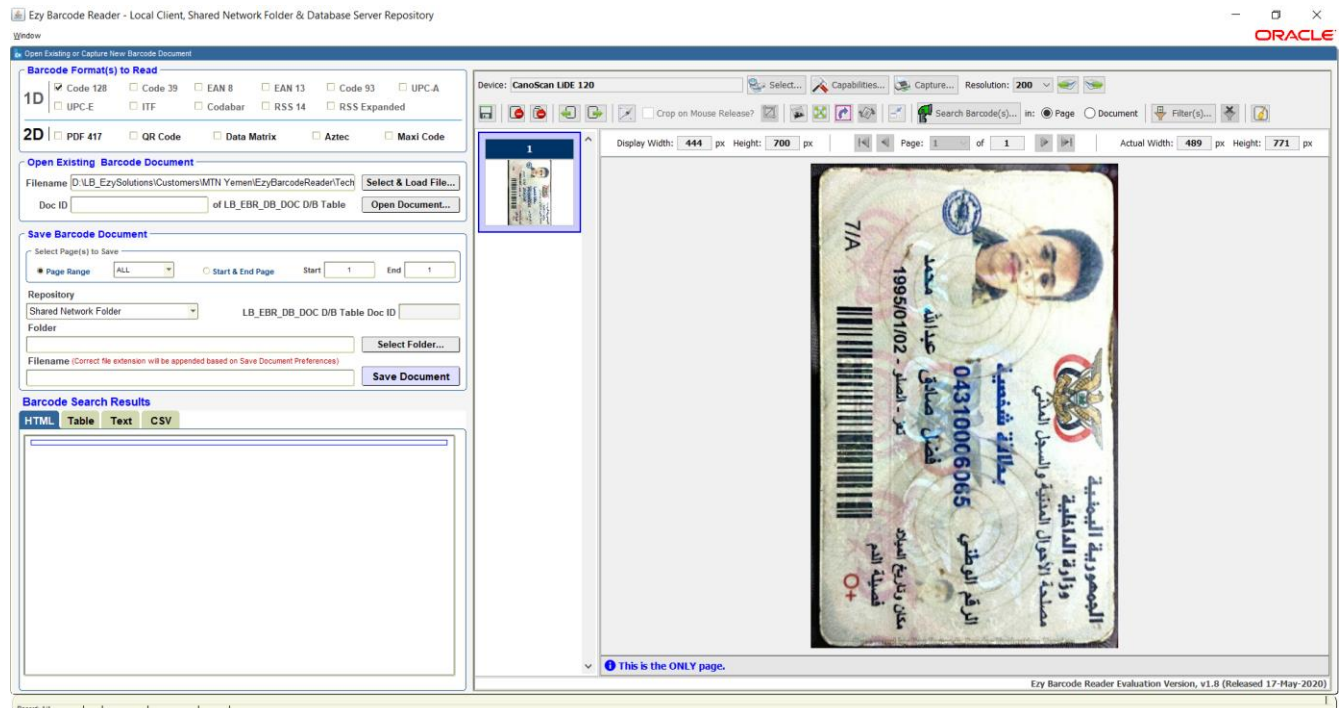
You should ALWAYS decode any existing barcode document once it has been captured from a supported imaging device or loaded from an existing document by pressing '**Search Barcode(s)...**' button as shown below:

Depending on the barcode document's file format, the radio button '**Page**' is normally selected for single page documents such as those with '**.bmp**', '**.gif**', '**.jpg**', '**.jpeg**' or '**.png**' file format, or '**Document**' is selected for multi-page documents with '**.tif**' or '**.tiff**' file format.

To open an existing document from either your Local Folder on your PC/laptop, or from a Shared Network Folder accessible from your local client, press '**Select & Load File...**' button, as shown below:

This action would display the **File Browser** dialog window as shown below, from where you can select the required barcode document to open:

Should the file be accessible, you should see your barcode document loaded successfully into the Java Bean Image Viewer. An example of a barcode document is as shown below:



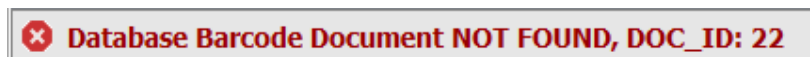
To access a barcode document from your Shared Network Folder, you may manually enter the filename, e.g., '\\Asus-bob\\c\\Users\\bob\\Documents\\A156\_ReadBarcodesSDK12.gif', in the 'File name:' field of the File Browser dialog window, and press 'Open' button to load the document.

Should a document from the Local Client or Shared Network Folder not exist within the folder location you had specified, you would get the following error in the Status Line of the Image Viewer:



To access a barcode document from your Database Server, where the sample application stores barcode documents into 'LB\_EBR\_DB\_DOCS' database table, enter a valid Document ID (DOC\_ID column) into 'Doc ID' field and press 'Open Document...' button.

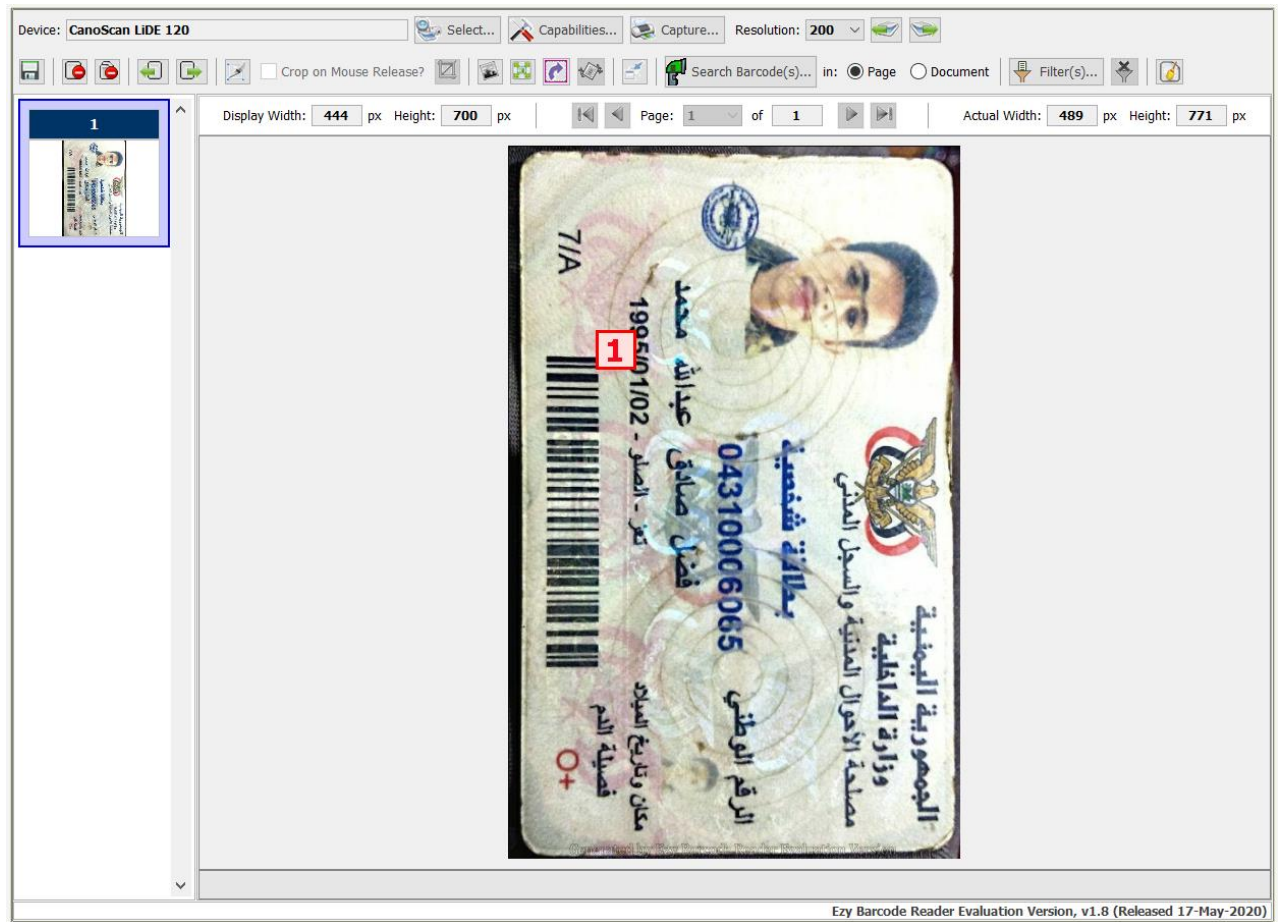
Should a document from the Database Server not exist within the database table where documents are stored, you would get the following error in the Status Line of the Image Viewer:



**ALWAYS** attempt to decode the document without requiring to use some of the provided **Image Manipulation** features within our **Ezy Barcode Reader** solution such as 'Crop Image', 'Convert to Grayscale', 'Resize, Rotate & Optimize Image', 'Rotate Image Selection', 'Blur' Filters such as 'Maximum', 'Median', 'Minimum' and 'Reduce Noise'; and 'Adjust Color Filters' such as 'Contrast' and/or 'Levels'.

You should have a good idea of the barcode(s) that exist within the document before you press 'Search Barcode(s)...' button.

In our example above, since I know that it is a 'Code 128' barcode and having its checkbox selected and pressing 'Search Barcode(s)...' button would generate the following output successfully within the Image Viewer, where the Barcode Number and its border is displayed in **RED** colour with a **PINK** background, as shown below:



The barcode results would automatically be generated and shown within the sample Oracle Forms program within the 4 relevant tab folders that shows output in 'HTML', 'Database Table', 'Text' and 'CSV' format, as shown below:

**Barcode Search Results**

HTML Table Text CSV

**Page 1**

**Code128: 1**

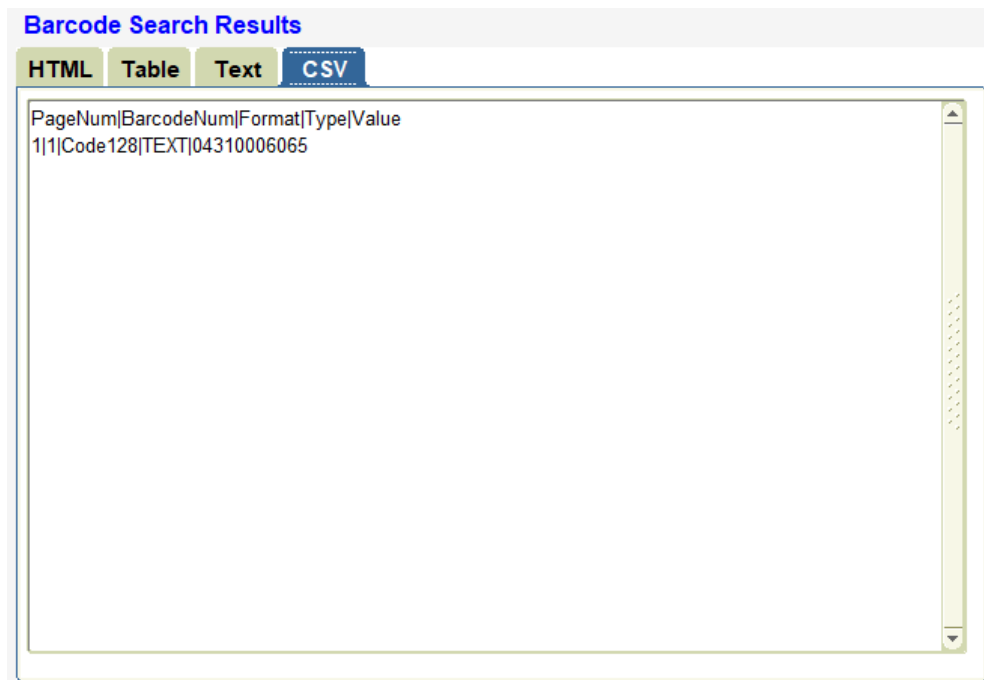
Barcode#	Type	Value
1	TEXT	04310006065

HTML Table Text CSV

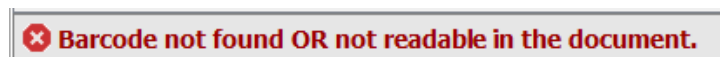
[illegible]

HTML Table **Text** CSV

Page 1  
Code128: 1  
Barcode#: 1  
Type: TEXT  
Value: 04310006065



In the case where a barcode is not recognized immediately upon pressing '**Search Barcode(s)...**' button, you would see the following message in the Status Line of the Image Viewer as shown below:



As such, you can use some of the **Image Manipulation** features as mentioned above on Page 2 to see if you can achieve the expected result that you are looking for.

You have a few options for improving barcode reading precision. Let's look at some of the most common areas you can influence.

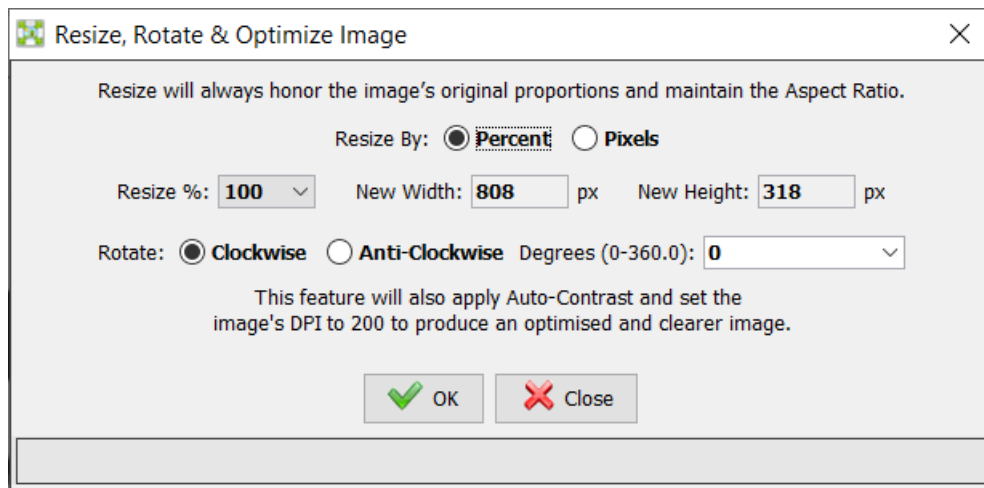
## Image Resolution

A barcode's recognition largely depends on the quality and size of the barcode image. Image resolution is the most important factor in barcode recognition accuracy. Usually, a minimum of 200 DPI is needed for acceptable barcode recognition. DPI stands for "dots per inch". Basically, the more dots per inch that make up an image, the more resolution it has. The lower the resolution, the more likely the symbol will lack the pixel density needed for good recognition.

To achieve this criterion, you can use '**Resize, Rotate & Optimize Image**' feature by pressing this button:

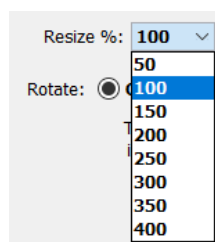


A popup dialog window as shown below will be displayed to allow you to enter the relevant data to manipulate the image to the desired size and resolution to try and decode the document again:

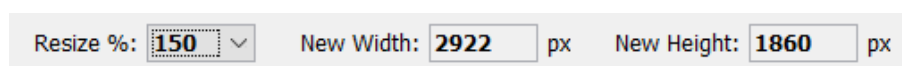


Should your original image size be too small when NO barcode was recognized, you can try increasing the size by resizing it to select one of the available options for **'Resize By'**, which is either by **'Percent'** or **'Pixels'**.

**'Resize By' Percent** provides you with a dropdown list of values as follows:

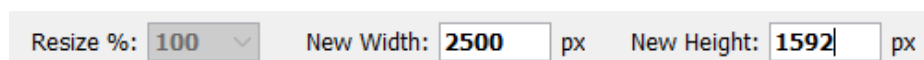


The current width and height of your barcode document will automatically be adjusted depending on the selection you have made. For example, if you choose 150%, then the new Width and Height will be as follows:



**'Resize By' Pixels** will enable you to change either the New Width or Height to your desired image dimension that you need. Bear in mind that on changing any one of Width or Height, the other value is automatically calculated for you to ensure that the image dimension's aspect ratio is maintained to generate a well-balanced image once you have made the required changes.

In my example below, I have entered '2500' as the New Width and upon clicking the mouse in the New Height field, its new value will be displayed, as shown below:

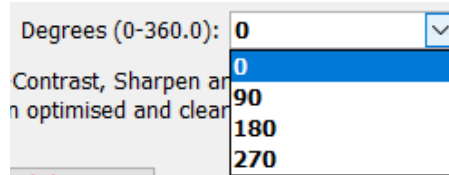


Should you wish to leave the image dimension as it is since its size is adequate enough to allow the document to be decoded correctly, you may optionally decide to rotate the image from its current position since it may not be horizontally aligned, which is normally the preferred alignment for a successful decode operation of your document.


To rotate your document and ensure that your barcode is horizontally aligned, which is preferred for faster recognition, you can either choose one of the radio buttons as shown below:

Rotate: ☒ **Clockwise** ☐ **Anti-Clockwise**

Upon making the required choice above, you can then manually enter the value for '**Degrees**' field, which should be a value between '**0**' to '**360.0**' with 1 decimal place only. Instead of manually entering the value, you could just press the 'down arrow' button and select one of '**0**', '**90**', '**180**' or '**270**' from the list.



Should you enter an invalid value which does not meet the validation requirements, i.e., a value between '**0**' to '**360.0**', and pressing '**OK**' button, you will see the following message within the Status Line of the popup window, as shown below:

 **Enter a maximum of 4 digits between 0 to 360.0 with 1 decimal place ONLY.**

Once you have made the relevant changes to either '**Resize**' and/or '**Rotation**', you can then press '**OK**' button to perform the Image Processing operation.

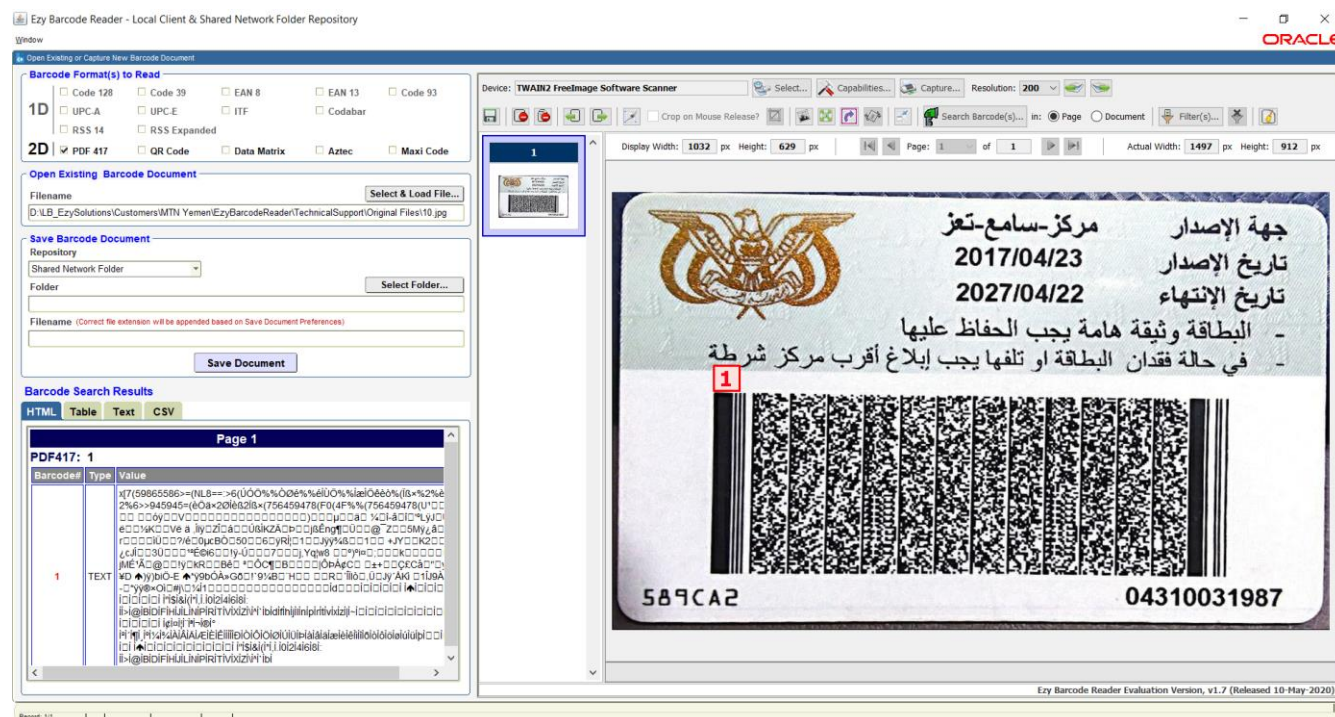
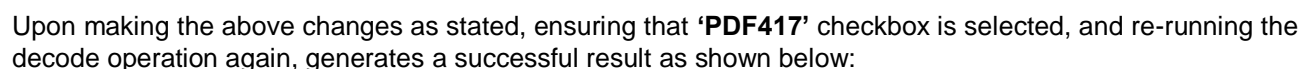
This action would also automatically generate an image where '**Auto Contrast**' and setting its **DPI** to '**200**' would be applied to the new image, and immediately replace the previous image within the Image Viewer.

You would see a message of this operation taking place, as per shown below, before the popup window is automatically closed to display the new image:

 **Image processing in progress. Please Wait...**

An example here demonstrates how a small image that cannot be initially read with the barcode being vertically aligned, and rotated anti-clockwise 90 degrees with its size being increased to 300%, and then decoded again, successfully generates the required results.

The original image, as shown below, upon pressing '**Search Barcode(s)...**' button, initially displays an error on its Status Line.



## Contrast

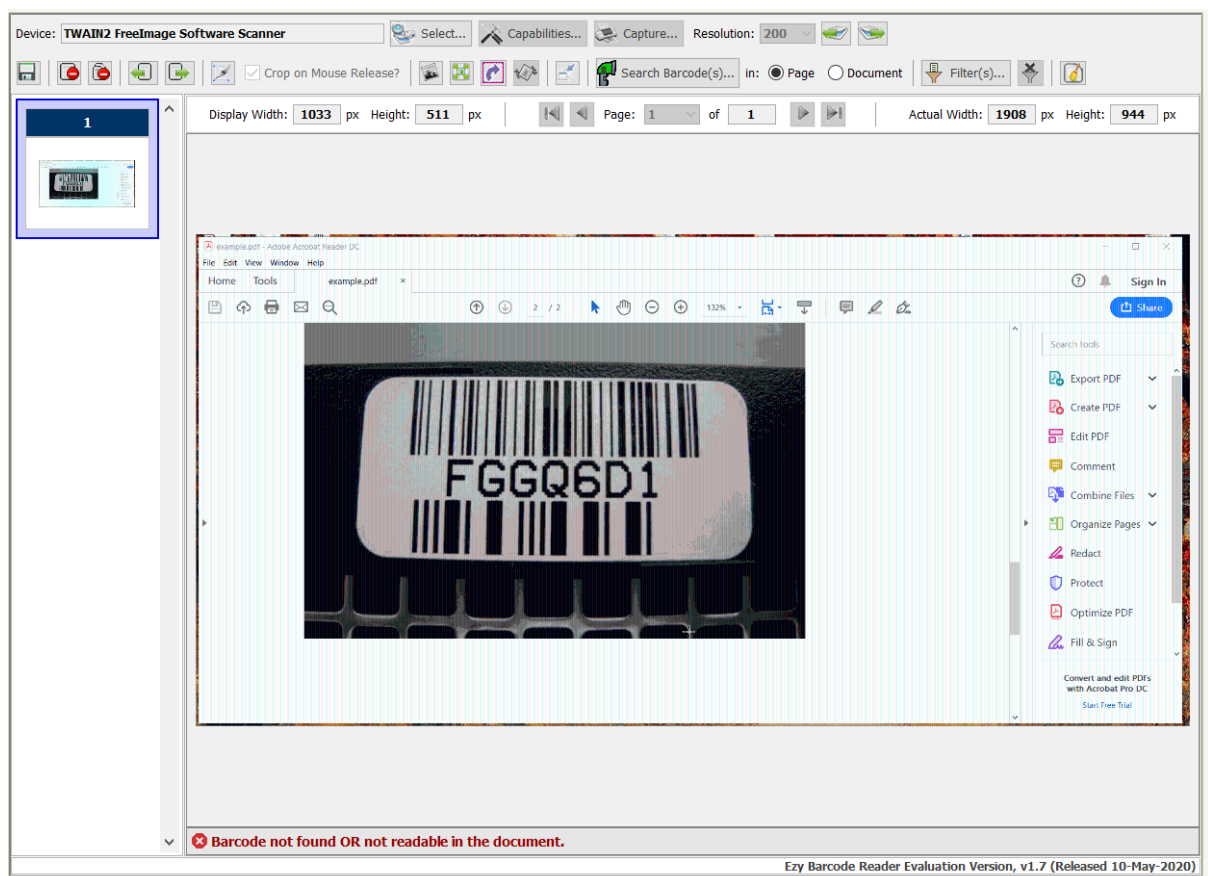
Symbol contrast is a must, where **Grayscale** or **'Black and White'** is BEST. Inaccurate recognition will normally be the result of a barcode which does not have the correct contrast within its document.

To achieve this criterion, you can use **'Convert to Grayscale'** feature by pressing this button:

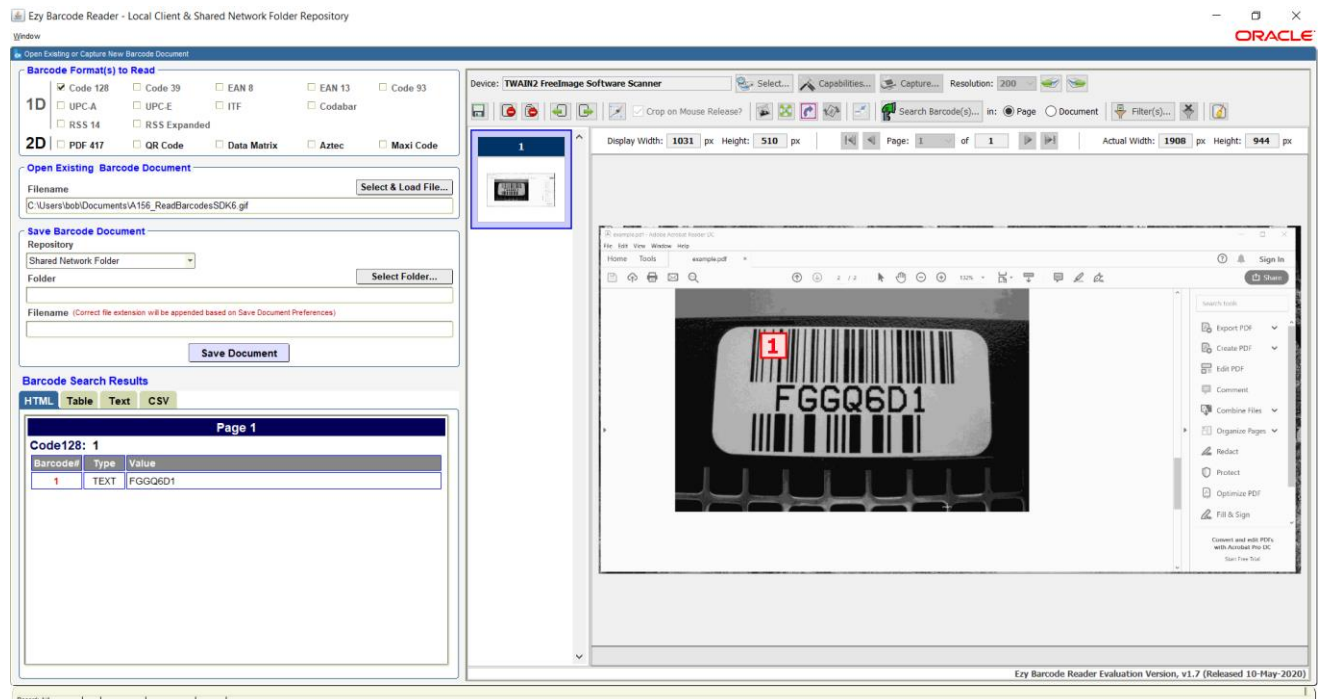


An example here demonstrates how a color image that cannot be initially read with the contrast being poor, and converting it to grayscale using the above button, and then decoded again, successfully generates the required results.

The original image, as shown below, upon pressing **'Search Barcode(s)...**' button, initially displays an error on its Status Line.



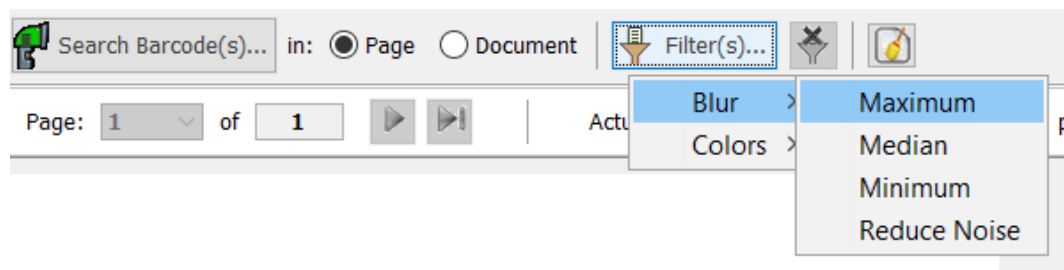
Upon pressing '**Convert to Grayscale**' button, ensuring that '**Code 128**' checkbox is selected, and re-running the decode operation again, generates a successful result as shown below:



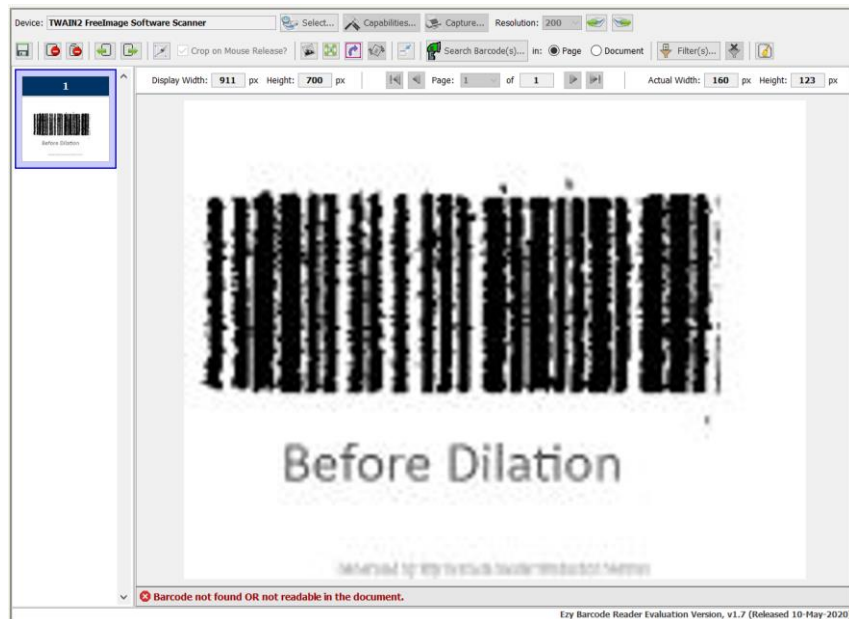
Other features within our **Ezy Barcode Reader** solution that can help to generate the correct contrast required includes '**Blur**' Filters such as '**Maximum**', '**Median**', '**Minimum**' and '**Reduce Noise**'; and '**Adjust Color Filters**' such as '**Contrast**' and/or '**Levels**'.

The '**Maximum**' filter, known as Dilation filter, replaces each pixel by the maximum of itself and its eight neighbours. This is normally used should the original pixels of your barcode be too dense or consists of thick bars, which results in less dense or thinner bars. The maximum filter is typically applied to an image to remove negative outlier noise.

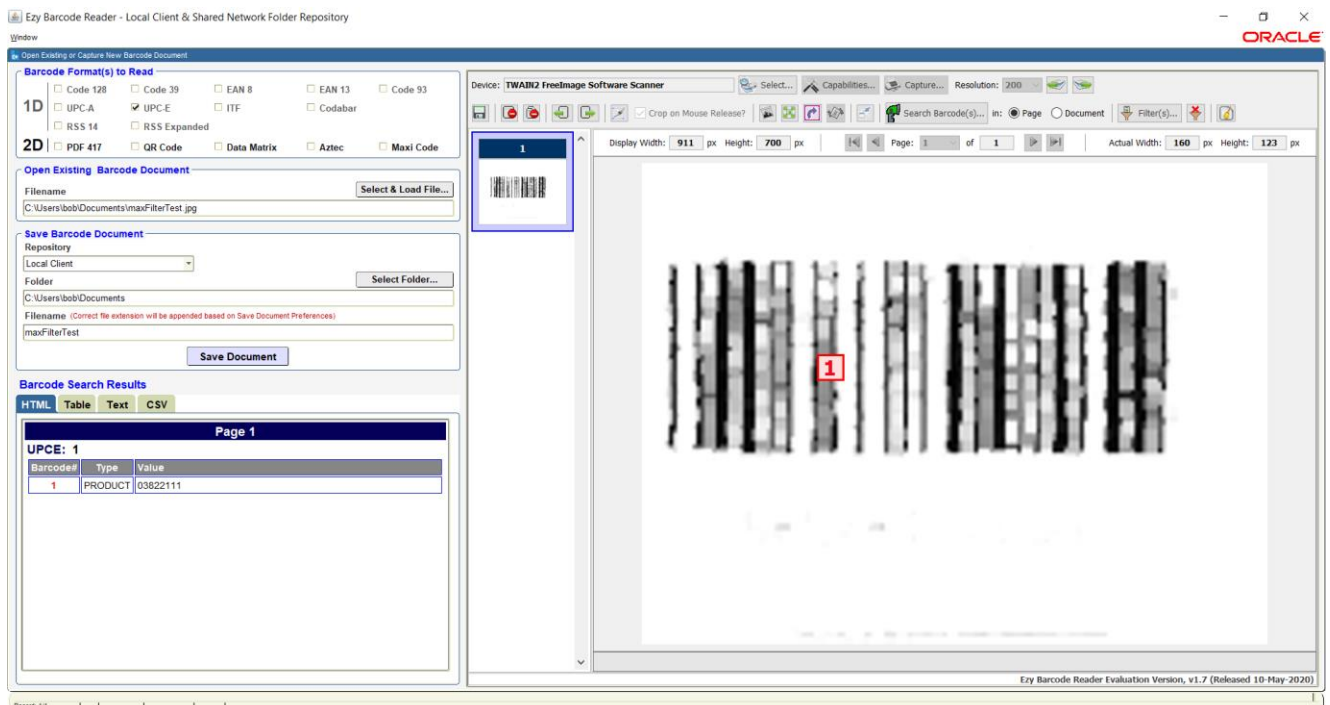
An example here demonstrates how an image that cannot be initially read with the contrast being poor, and applying the **Maximum Filter**, as shown below on how to do this, and then decoded again, successfully generates the required results.



The original image, as shown below, upon pressing **'Search Barcode(s)...**' button, initially displays an error on its Status Line.



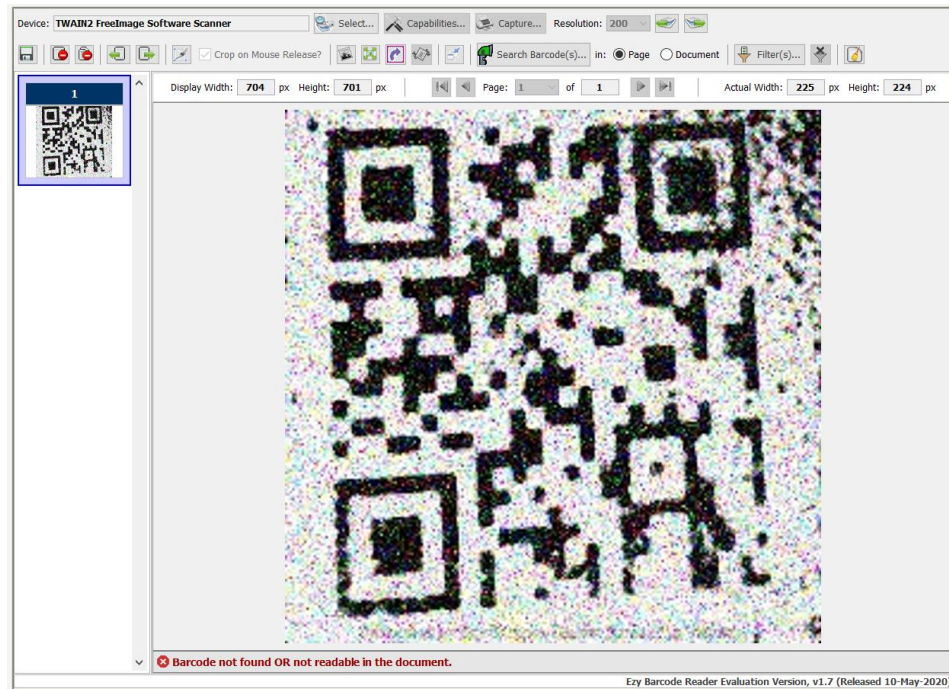
Upon pressing **'Maximum'** sub-menu item from **'Blur'** menu item, as shown above, ensuring that **'UPC-E'** checkbox is selected, and re-running the decode operation again, generates a successful result as shown below:



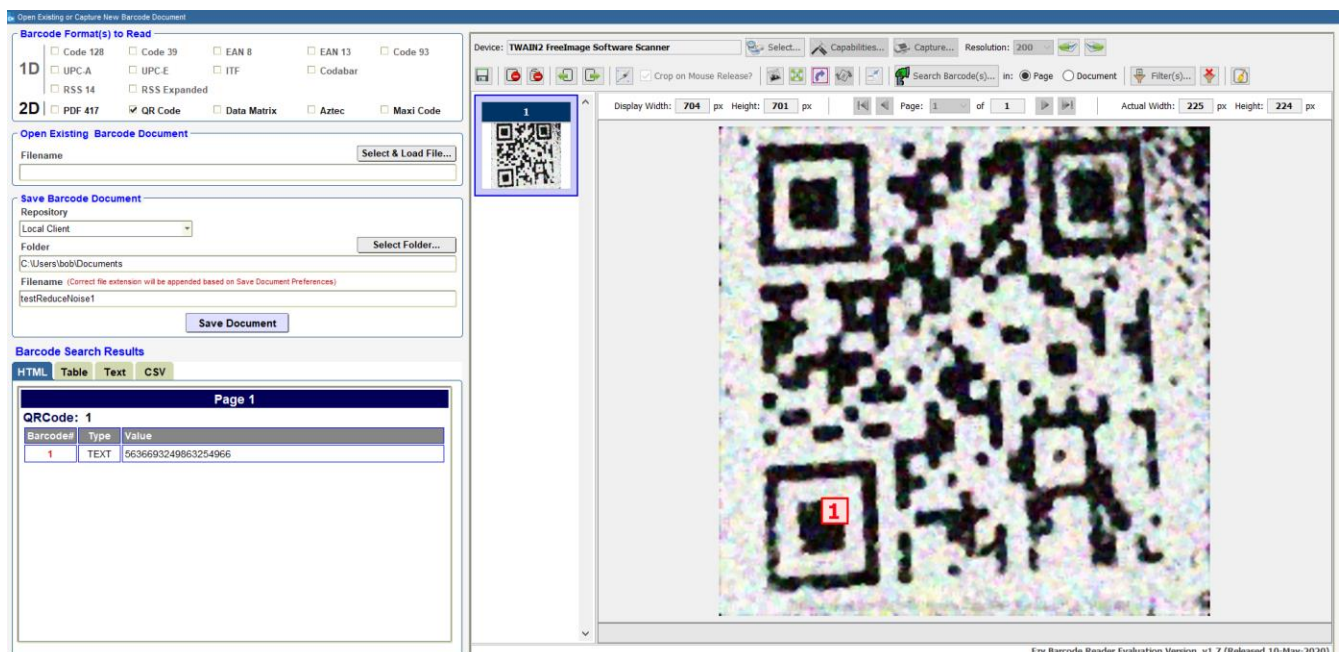
The '**Median**' filter performs a 3x3 median operation. It is useful for removing dust and noise. Should an image have specks of dots all over the image that interferes with the decoding operation, it is best that we apply the Median filter to see if the image can then be decoded successfully.

An example here demonstrates how an image that cannot be initially read with the contrast being poor, and applying the **Median Filter**, and then decoded again, successfully generates the required results.

The original image, as shown below, upon pressing '**Search Barcode(s)...**' button, initially displays an error on its Status Line.



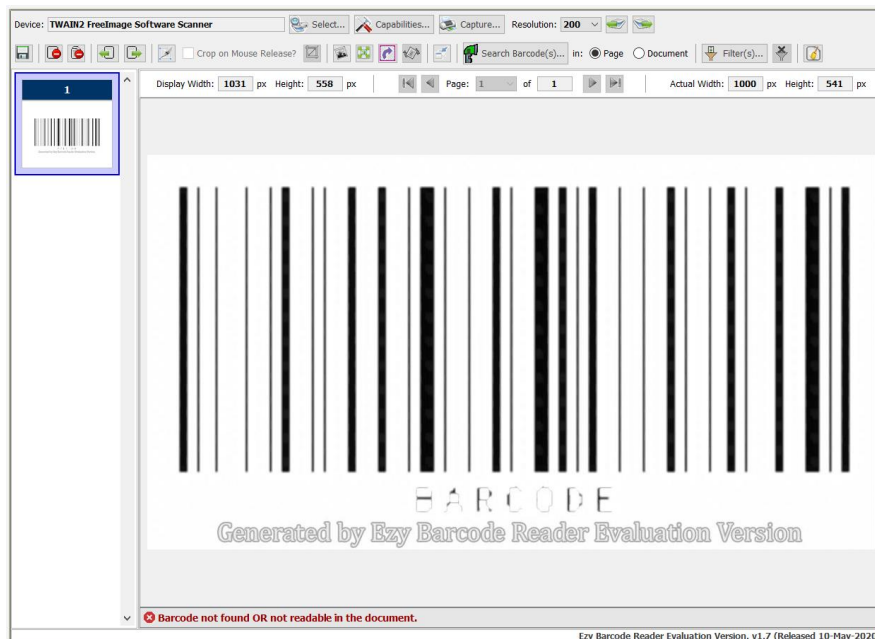
Upon pressing '**Median**' sub-menu item from '**Blur**' menu item, ensuring that '**QR Code**' checkbox is selected, and re-running the decode operation again, generates a successful result as shown below:



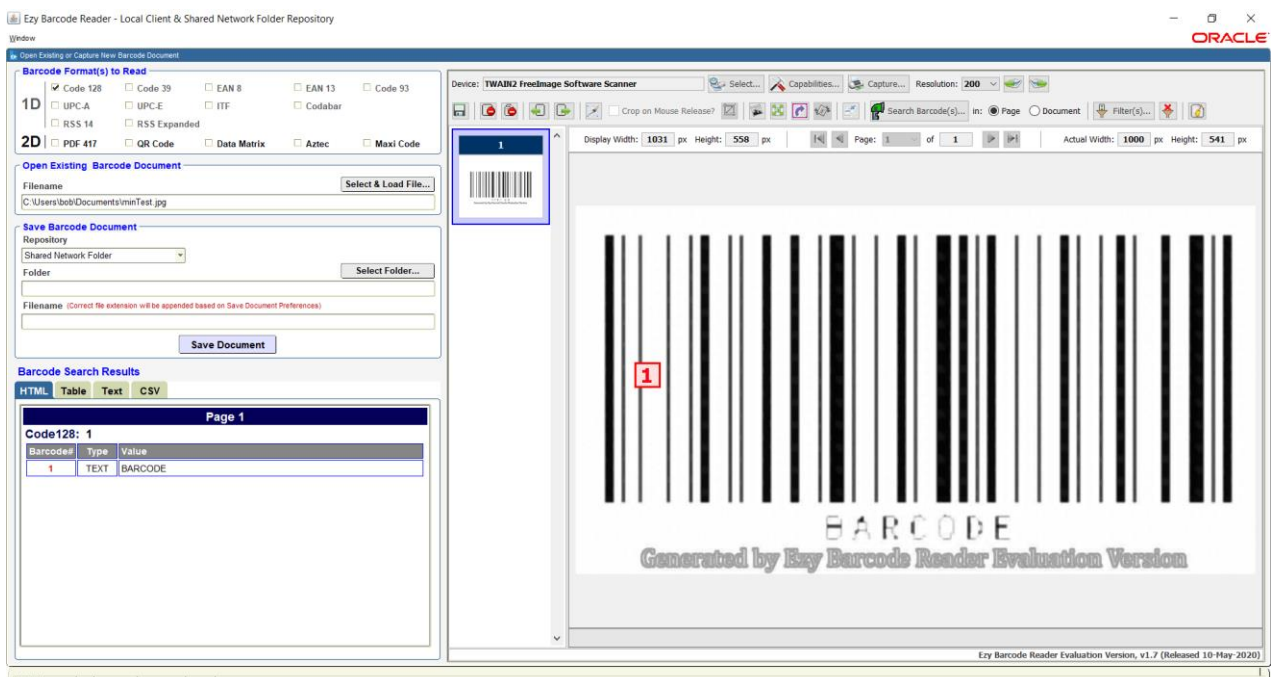
The **'Minimum'** filter, known as Erosion filter, replaces each pixel by the minimum of itself and its eight neighbours. This is normally used should the original pixels of your barcode be less dense or consists of thin bars, which results in more dense or thicker bars. The minimum filter is typically applied to an image to remove negative outlier noise.

An example here demonstrates how an image that cannot be initially read with the contrast being poor, and applying the **Minimum Filter**, and then decoded again, successfully generates the required results.

The original image, as shown below, upon pressing **'Search Barcode(s)...**' button, initially displays an error on its Status Line.

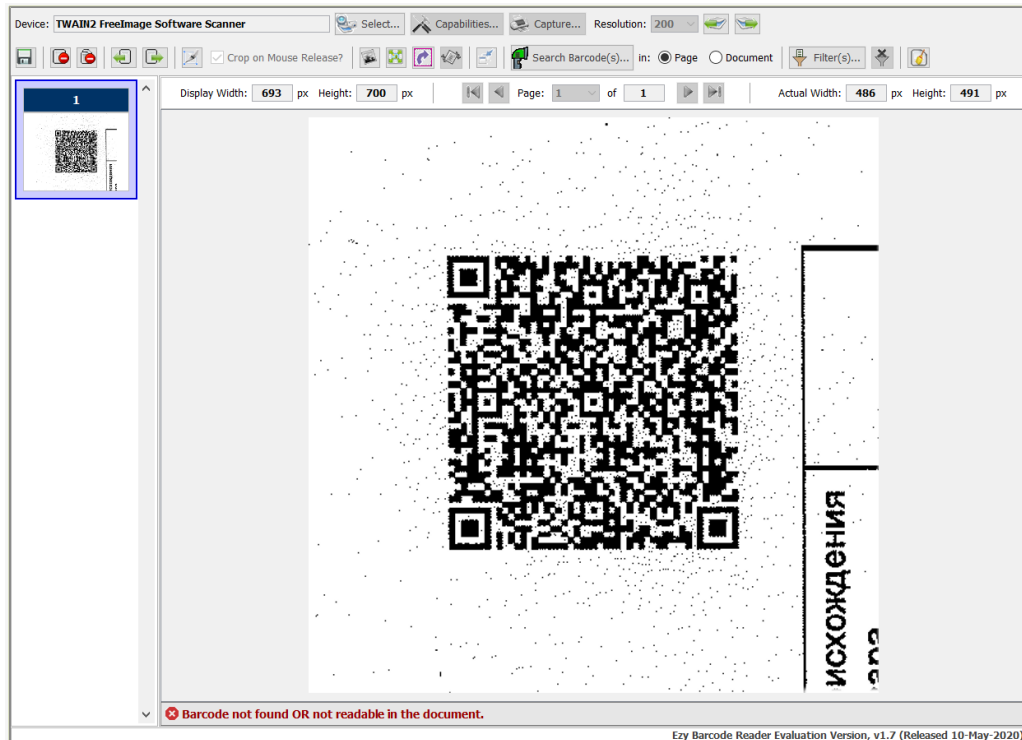


Upon pressing **'Minimum'** sub-menu item from **'Blur'** menu item, ensuring that **'QR Code'** checkbox is selected, and re-running the decode operation again, generates a successful result as shown below:

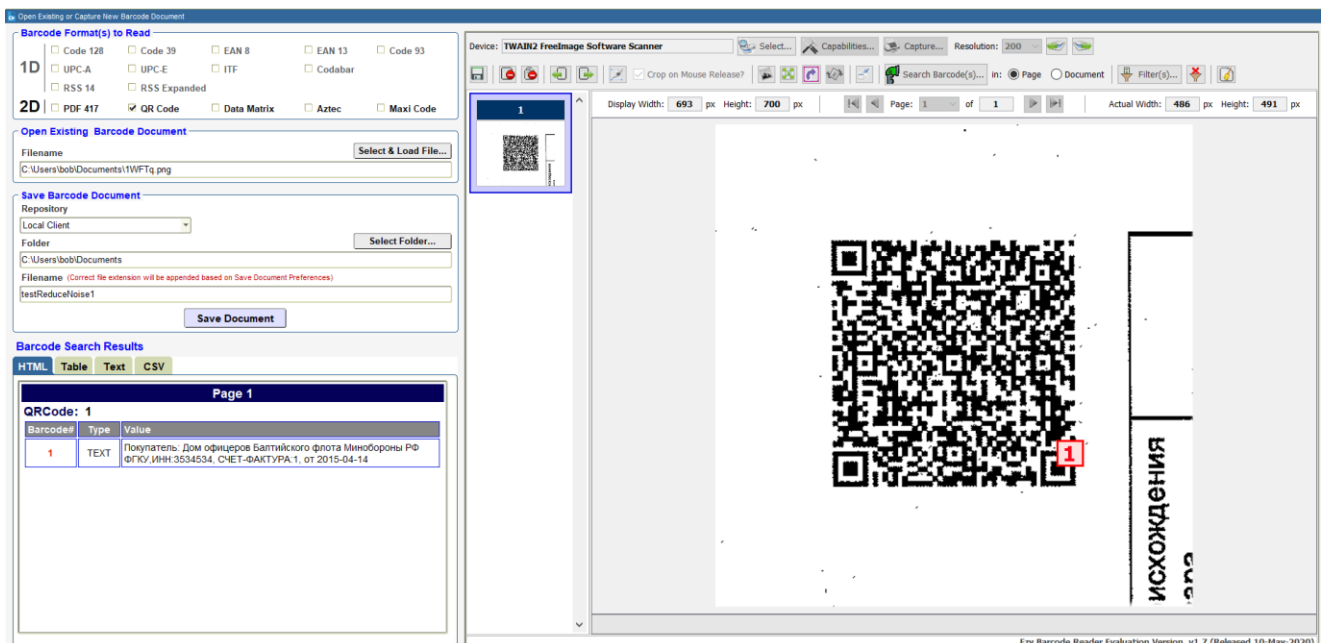


The **'Reduce Noise'** filter reduces or removes noise by looking at each pixel's 8 neighbours, and if it's a minimum or maximum, replacing it by the next minimum or maximum of the neighbours. An example here demonstrates how an image that cannot be initially read with the contrast being poor, and applying the **Reduce Noise Filter**, and then decoded again, successfully generates the required results.

The original image, as shown below, upon pressing **'Search Barcode(s)...**' button, initially displays an error on its Status Line.

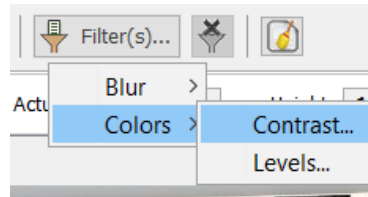


Upon pressing **'Reduce Noise'** sub-menu item from **'Blur'** menu item, ensuring that **'QR Code'** checkbox is selected, and re-running the decode operation again, generates a successful result as shown below:

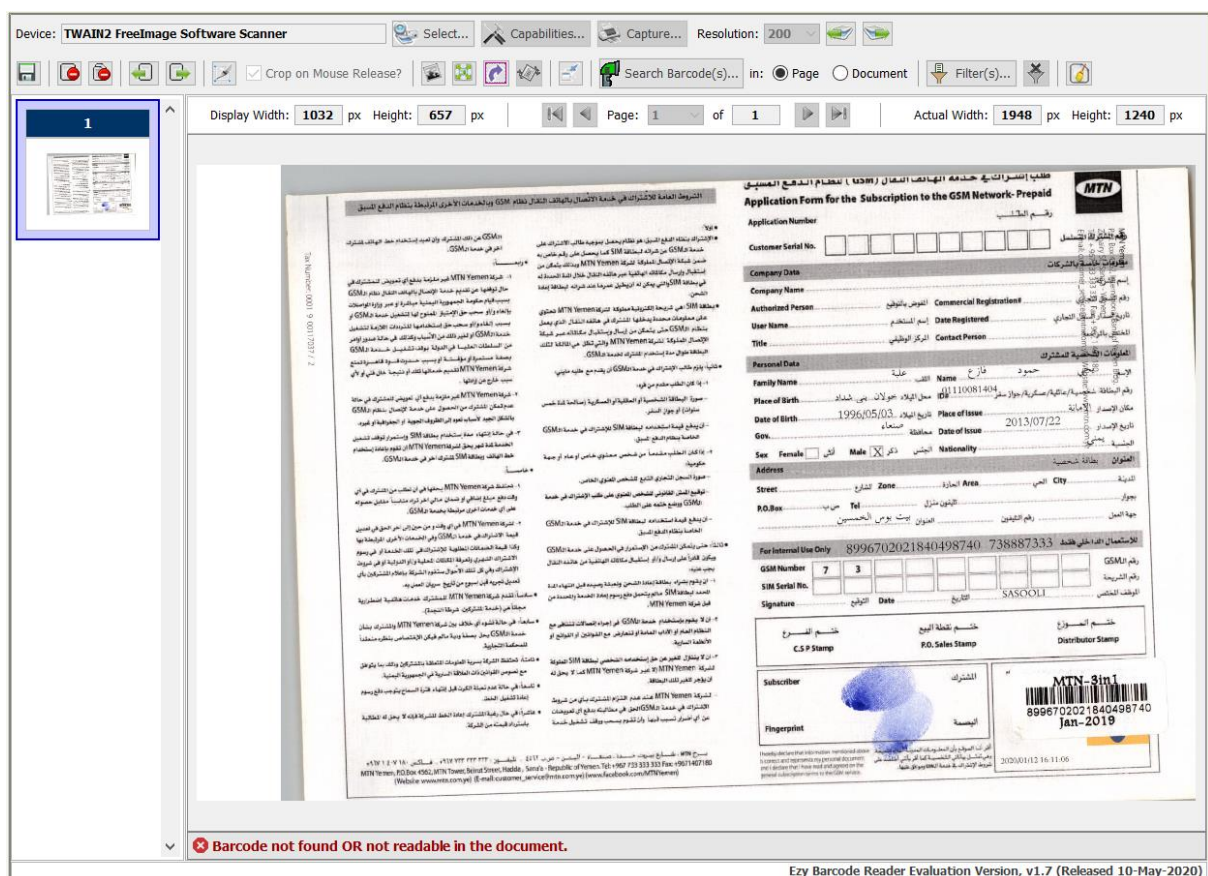


The **'Colors / Contrast'** filter allows you to adjust the Brightness and Contrast of an image should the initial decode operation fail to detect the presence of a barcode within your document.

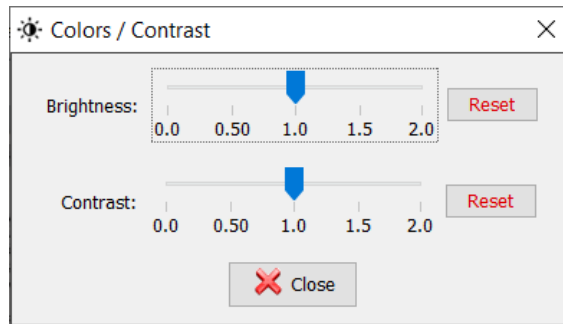
An example here demonstrates how an image that cannot be initially read with the contrast being poor, and applying the **'Colors / Contrast' Filter**, as shown below on how to do this, and then decoded again, successfully generates the required results.



The original image, as shown below, upon pressing **'Search Barcode(s)...'** button, initially displays an error on its Status Line.



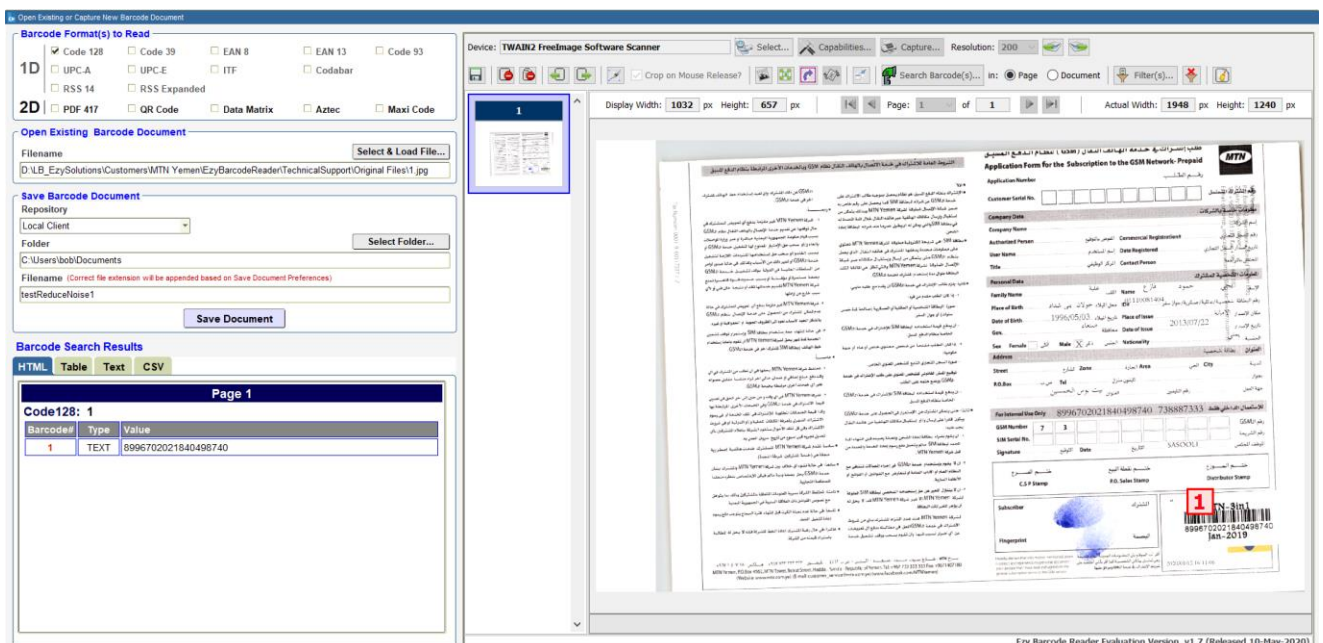
Upon pressing **'Contrast'** sub-menu item from **'Colors'** menu item, the following pop-up window will be displayed:



By adjusting the Brightness by moving the slider 1 mouse click towards right of 1.0, this will allow you to increase the brightness to generate a clearer whiter background. It is best to try in small increment of adjusting either the Brightness and/or Contrast until you achieve the desired results.

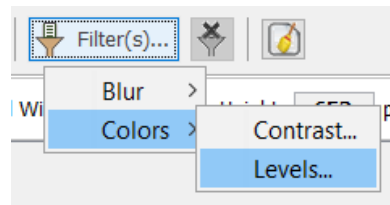
The main objective of manipulating the image's contrast is to be able to differentiate between the light and dark elements of the symbol. Both element types are essential for proper decoding, enabling the barcode reader engine to obtain the precise patterns of barcode elements that represent encoded data in the symbol.

Ensuring that **'Code 128'** checkbox is selected and re-running the decode operation again, generates a successful result as shown below:

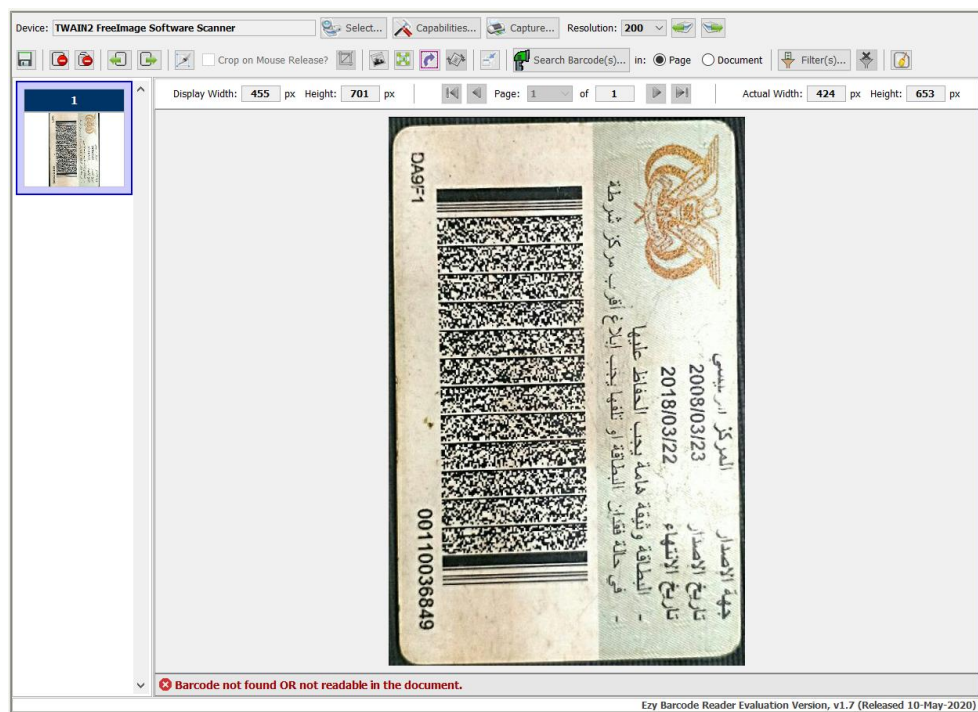


The **'Colors / Levels'** filter allows you to correct the tonal range and color balance of an image by adjusting intensity levels of image shadows, midtones, and highlights, and highlights, should the initial decode operation fail to detect the presence of a barcode within your document.

An example here demonstrates how an image that cannot be initially read with the contrast being poor, and applying the **'Colors / Levels' Filter**, as shown below on how to do this, and then decoded again, successfully generates the required results.

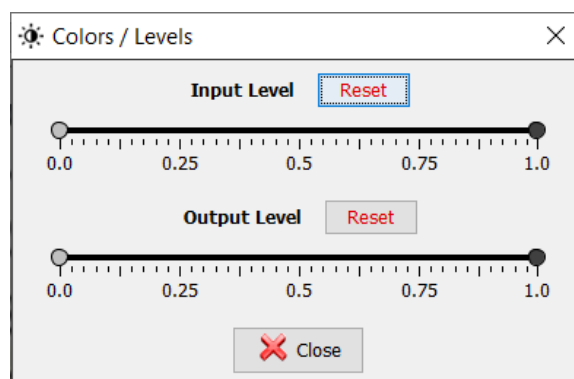


The original image, as shown below, upon pressing '**Search Barcode(s)...**' button, initially displays an error on its Status Line.



As specified earlier, especially for **PDF417** barcodes, it is best that we first rotate this image to be in a horizontal alignment, where we will use the '**Resize, Rotate & Optimize**' image manipulation feature first to do this.

Having done the above, upon pressing '**Levels**' sub-menu item from '**Colors**' menu item, the following popup window will be displayed:



As seen above there are 2 slider knobs for the '**Input Level**' and '**Output Level**'.

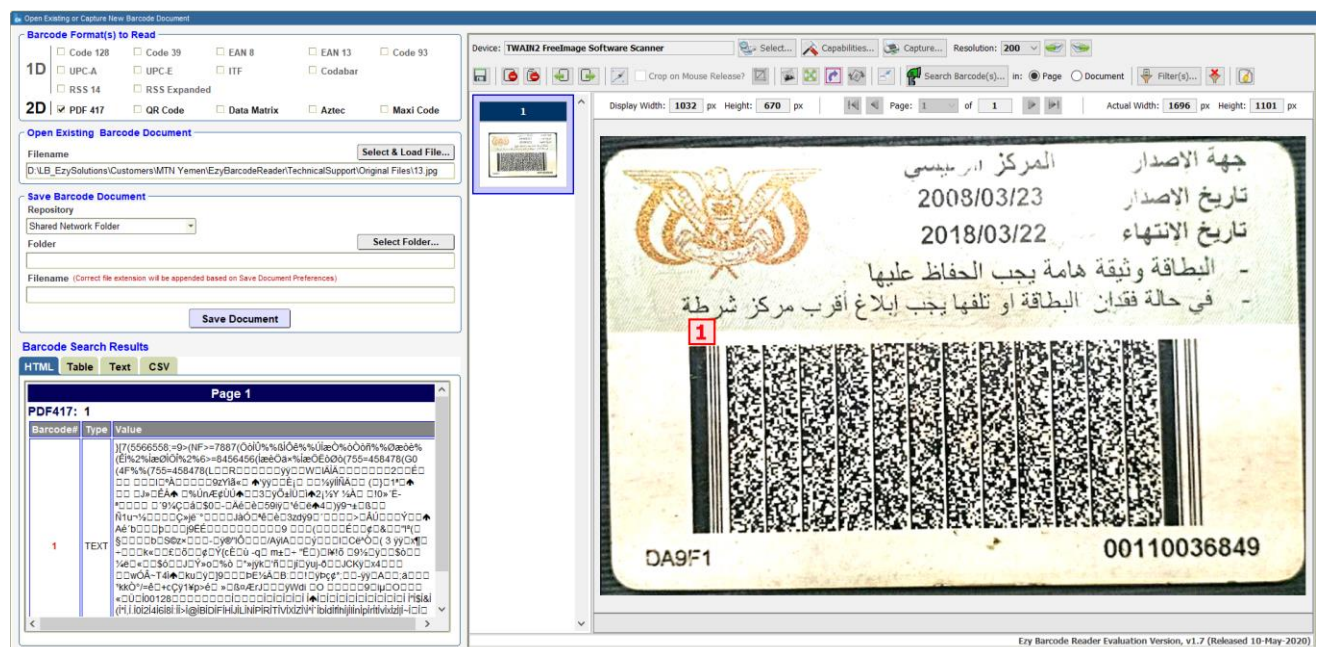
The outer two Input Levels sliders map the black point and white point to the settings of the Output Level sliders.

By default, the Output Level sliders are at level 0.0, where the pixels are black, and level 1.0, where the pixels are white. With the Output Level sliders in the default positions, moving the black input slider maps the pixel value to level 0.0 and moving the white point slider maps the pixel value to level 1.0. The remaining levels are redistributed between levels 0.0 and 1.0. This redistribution increases the tonal range of the image, in effect increasing the overall contrast of the image.



Most of the time, you would only be adjusting the Input Level slider knobs. Moving the left slider from 0.0 towards your right, increases the black pixel value (adjusts contrast to darken the black points), and moving the right slider from 1.0 towards your left increases the white pixel value (adjusts brightness to provide a whiter background).

Adjusting the right-hand side Input Level slider by moving the slider slightly towards left of 1.0 to midway between 0.75 and 1.0 will make the background whiter. It is best to try in small increment of adjusting either the 2 Input Level slider knobs until you achieve the desired results.

Ensuring that **'PDF 417'** checkbox is selected and re-running the decode operation again, generates a successful result as shown below:




The main objective of using this filter is to adjust the brightness, contrast, and tonal range to generate a clear distinction between the black and white points of your image so that the barcode can be easily decoded.

In the event that any of your **Filter** operations, such as, **'Maximum'**, **'Median'**, **'Minimum'**, **'Reduce Noise'**, **'Contrast'** and/or **'Levels'**, which ALL are available by pressing the **'Filter(s)...**' button, , did not generate the desired result upon running the **'Search Barcode(s)'** operation, you will notice that the **'Undo <xxxxx> Filter'** button, , will automatically be enabled to allow you to undo the most recent Filter operation that you had performed.

'<xxxxxx>' will contain the name of the Filter that you most recently used.

Example, if you had just used 'Levels' Filter, then this button's Tooltip would say **'Undo Levels Filter'**. By pressing this 'Undo Filter' button, the previous image within the Image Viewer just prior to your 'Levels' Filter operation will automatically be updated within the viewer area. You can then try the same or another filter operation, with different values (especially for Contrast and/or Levels filter) that you believe would allow you to successfully decode the barcode image to generate the desired result.




Should you wish to reload the original image within the current page of the loaded document at any time to restart the entire set of operations you wish to perform, you can always press the **'Restore Original Image'** button, .

For different images, you will need to adjust the Brightness and/or Contrast several times using whichever of the features mentioned so far, until you generate the desired results.

Naturally if the image is really very poor where you can visually see that it is not well created or generated, then there is no need to waste any time trying to figure out different ways of trying to generate a successful result. It is better to request for a clearer image or re-generate the barcode image correctly so that it can be readily decoded without the need to use any of the Image Manipulation features.

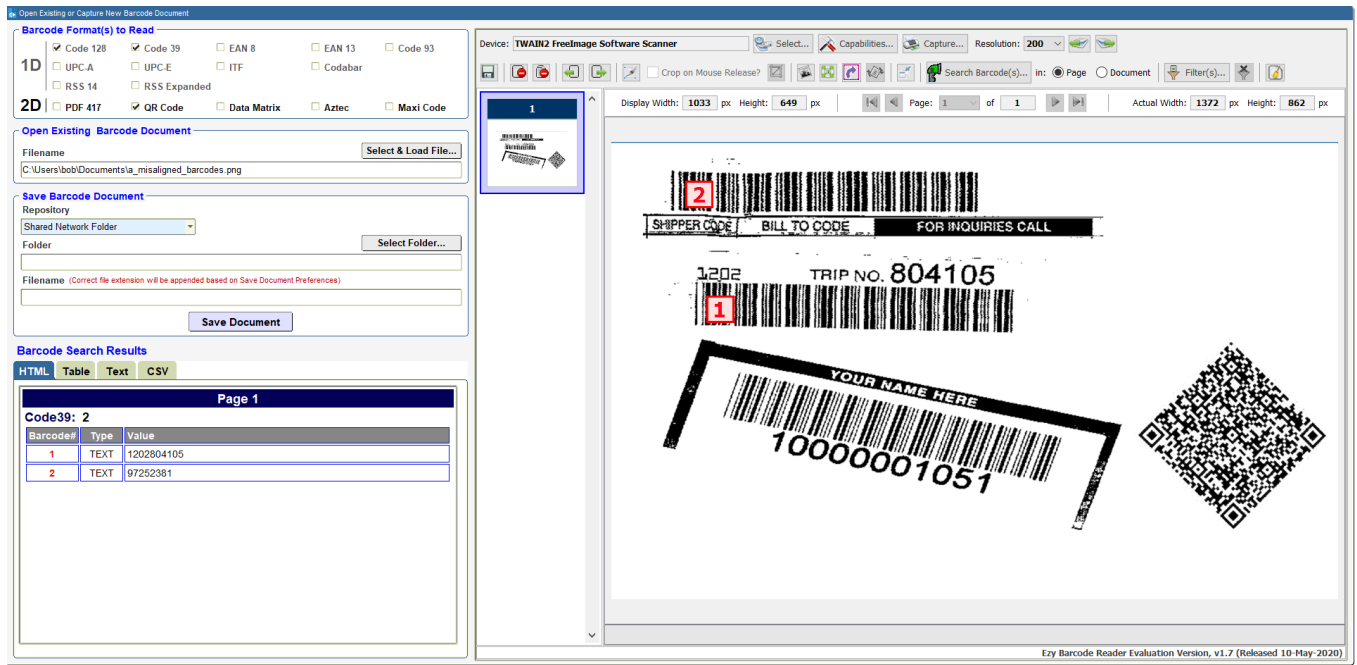
Another useful Image Manipulation feature that Ezy Barcode Reader provides is **'Rotate Selection'** feature. This is mainly going to be used to rotate a certain barcode within your document that may be deskewed, misaligned or placed at a certain angle that makes it difficult for the Barcode Decode Engine to easily read the barcode image.

Within the Toolbar, these buttons would be used to allow you to carry out the required action and operation, and they are as follows:

	This is the <b>'Enable Rotate Selection'</b> button that will allow you to draw the rectangular area of a barcode image that you wish to rotate. As soon as you press this button, the icon will change to the <b>'Cancel Rotate Selection'</b> icon as shown below and the tooltip text accordingly.
	This is the <b>'Cancel Rotate Selection'</b> button that will allow you to exit the <b>'Enable Rotate Selection'</b> mode back to Normal mode, in case you decide to cancel the <b>'Rotate Selection'</b> operation. Pressing this button will change its icon back to <b>'Enable Rotate Selection'</b> icon as shown above and its tooltip text accordingly.
	This is the <b>'Rotate Selection'</b> button that will allow you to rotate your selected area to your desired position, normally horizontally or vertically, though you may also set it to whichever angle so that the Barcode Decode Engine can clearly detect it for its operation.

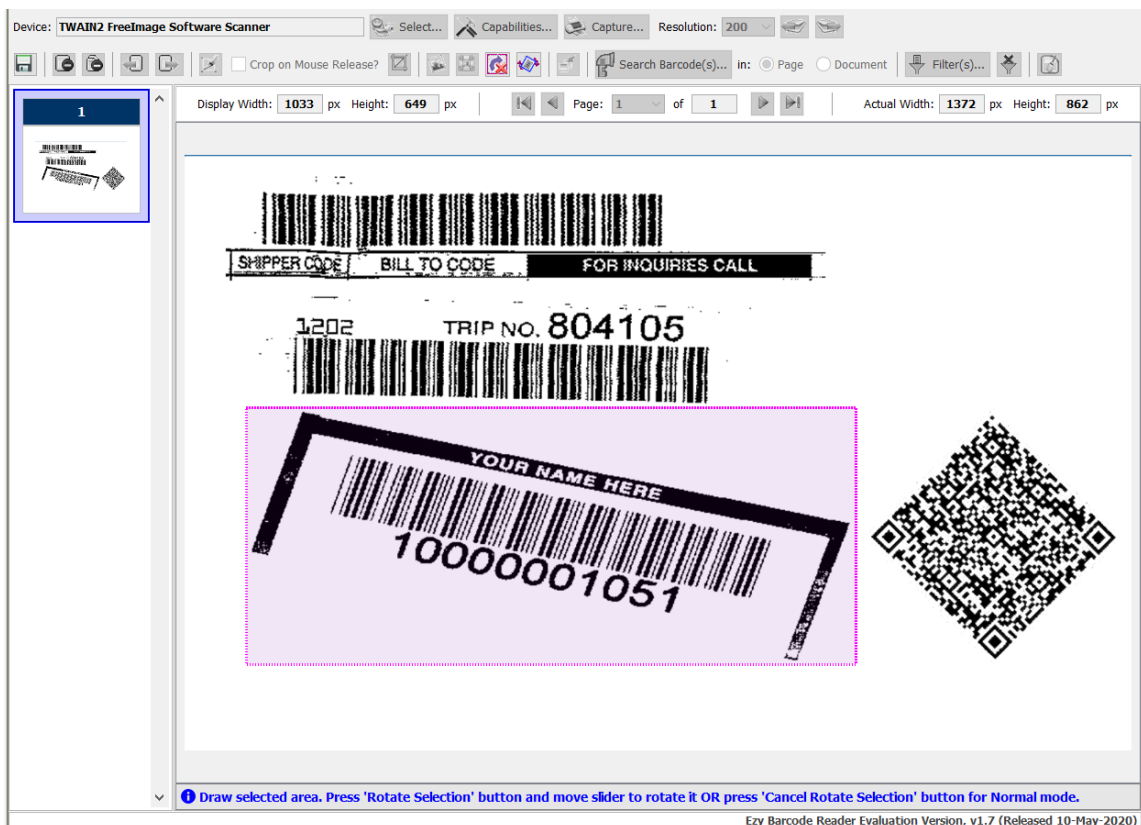
An example here demonstrates how certain barcodes cannot be initially read due to its images being deskewed or placed at a certain angle. Only 2 **'Code 39'** barcode images were readily detected and read by pressing **'Search Barcode(s)...**' button initially.

By using the **'Rotate Selection'** image manipulation feature, and then decoded again, successfully generates the required results.



From the above document, you will notice that 2 barcode images are deskewed or placed at a certain angle that makes it difficult for the Barcode Decode Engine to read it successfully.

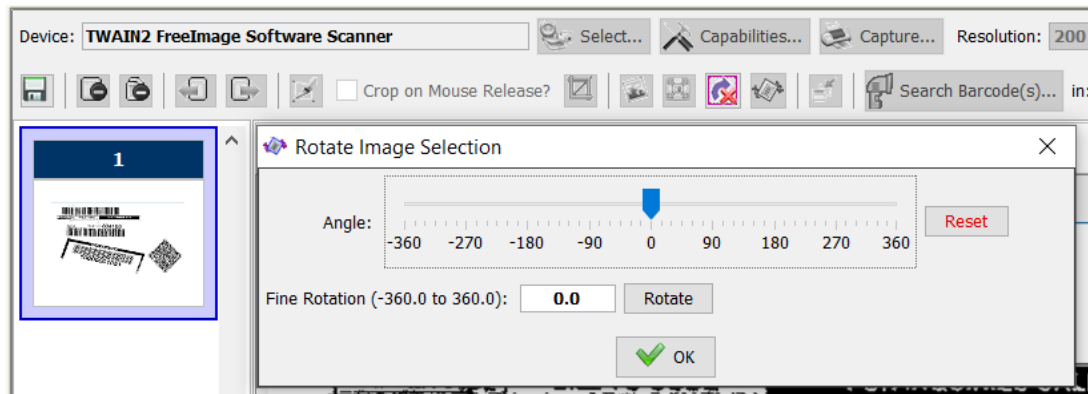
Let's start by pressing the **'Enable Rotate Selection'** button to allow us to draw the rectangular area around the **'Code 128'** barcode image so that we can rotate it to a horizontal position. See the screen below of how this is achieved:



The selected area as highlighted in a purple outline needs to encompass the entire barcode image that you wish to rotate. It is not required to include any unwanted areas that actually do not represent the barcode itself. It is also best to carry out this task one barcode image at a time that was not initially recognized, if you had done an initial decode operation.

Please take note that **ONLY** these 2 buttons related to this '**Rotate Selection**' task are actually enabled, while all the other buttons are automatically disabled.

Once we have created the selected area correctly, go ahead and press the '**Rotate Selection**' button. This will display the following popup window, as follows:



The '**Angle**' **Slider** as shown above has a range from **-360** to **360** to allow you to do a full rotation, either clockwise or anti-clockwise. You can see the effect of just moving the slider knob from one end to the other end to appreciate the beauty and efficiency of this feature. The '**Reset**' button will set your image back to its original location and set the location of the slider knob back to **0**.

As you move the slider knob leftward or rightward, the value will be automatically updated within the '**Fine Rotation**' field. In case you wish to change the value, as long as it is validated correctly to 1 decimal place (preferably), you can do so and press '**Rotate**' button to update the new location of the selected image.

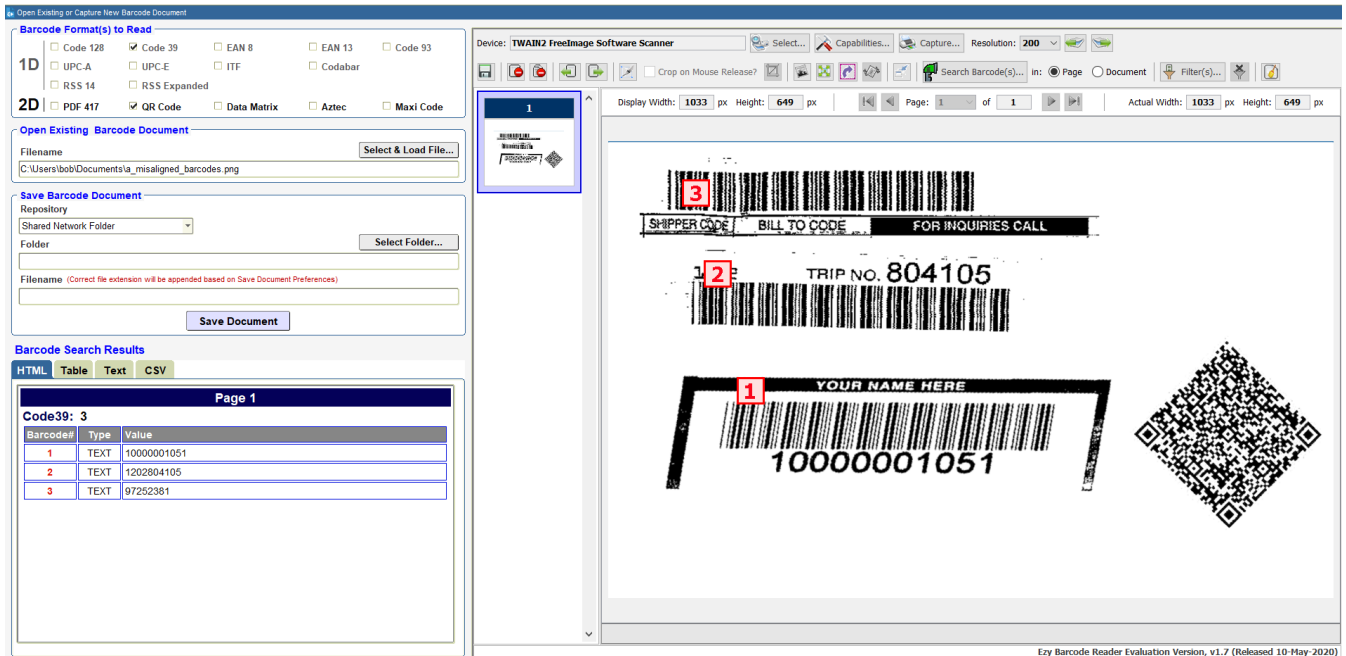
An invalid value will generate the following error on the Status Line within the Image Viewer, as shown below:

**✖ Enter a maximum of 4 digits between -360.0 to 360.0 with 1 decimal place ONLY.**

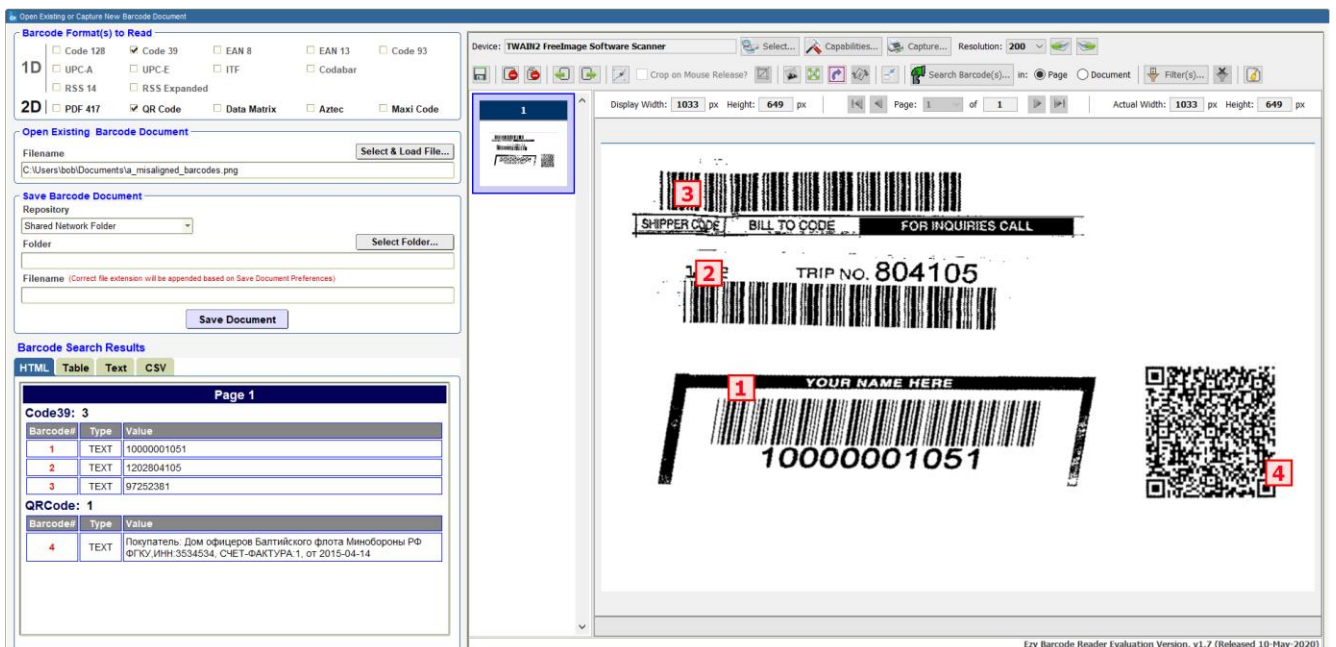
Once you are satisfied that your new updated barcode image has been rotated as per your desired location, press '**OK**' button to regenerate an updated document that automatically replaces the previous image loaded within the Image Viewer.

Ensure that both '**Code 39**' and '**QR Code**' checkboxes are selected, then press '**Search Barcode(s)...**' button again to check if your rotation operation on the selected image can now be recognized.

For my example above, it produces the following results that show that my rotation operation was successful:

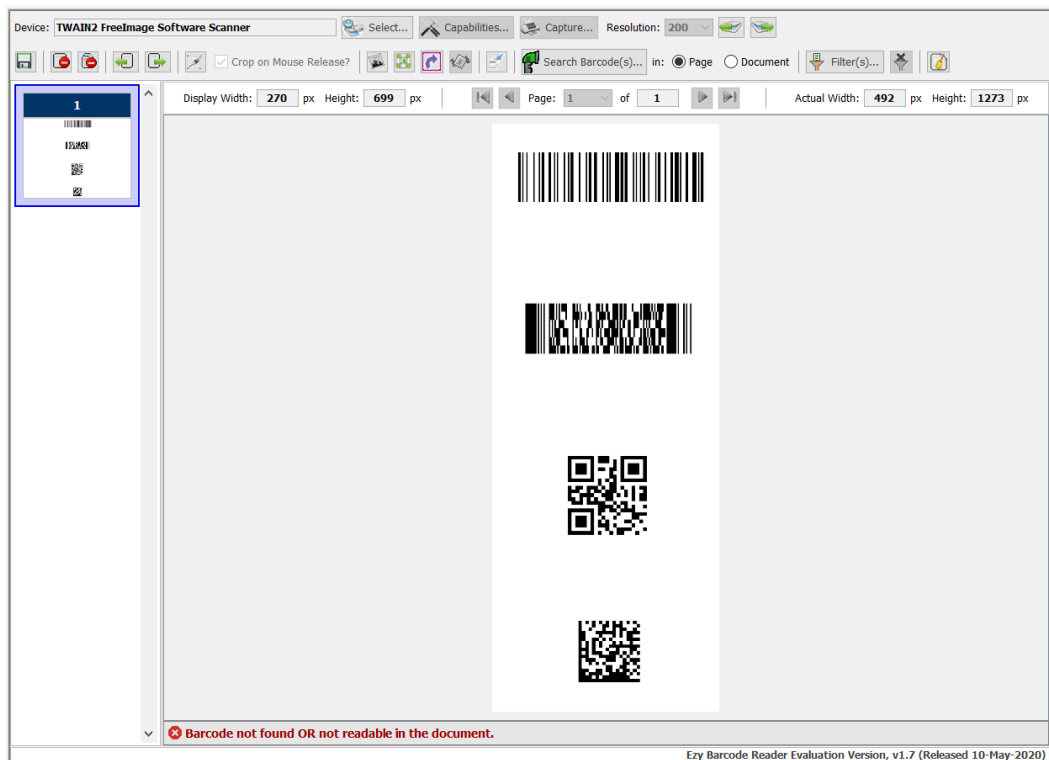


Repeat the same procedure against the 'QR Code' barcode image to align it horizontally, and re-run the decode operation. If you have done it successfully, this will generate the following results:



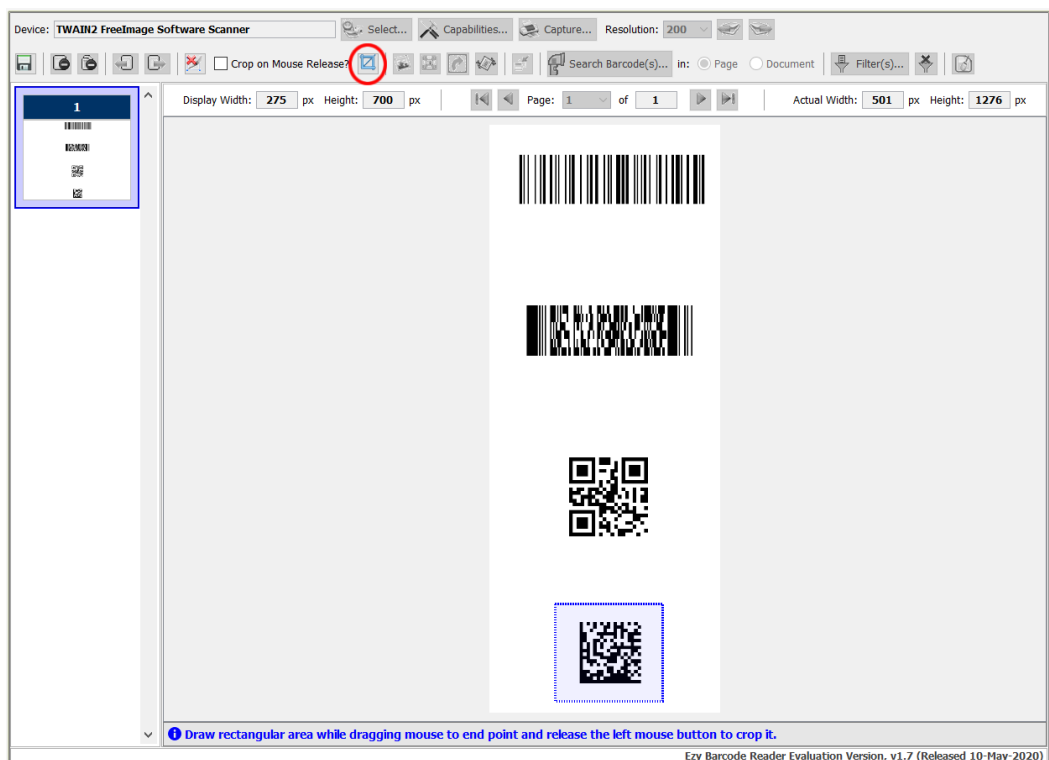
Another useful feature that is provided in Ezy Barcode Reader is 'Image Crop' feature. This allows you to select the barcode image that you are interested to isolate from any other barcode image(s) within your document, so that you can read this successfully after the cropping process.

An example here demonstrates our interest in reading ONLY the 'Data Matrix' barcode from the 4 barcode images within our sample document, but the initial search upon pressing the 'Search Barcode(s)...' button did not manage to decode it successfully, as shown below:

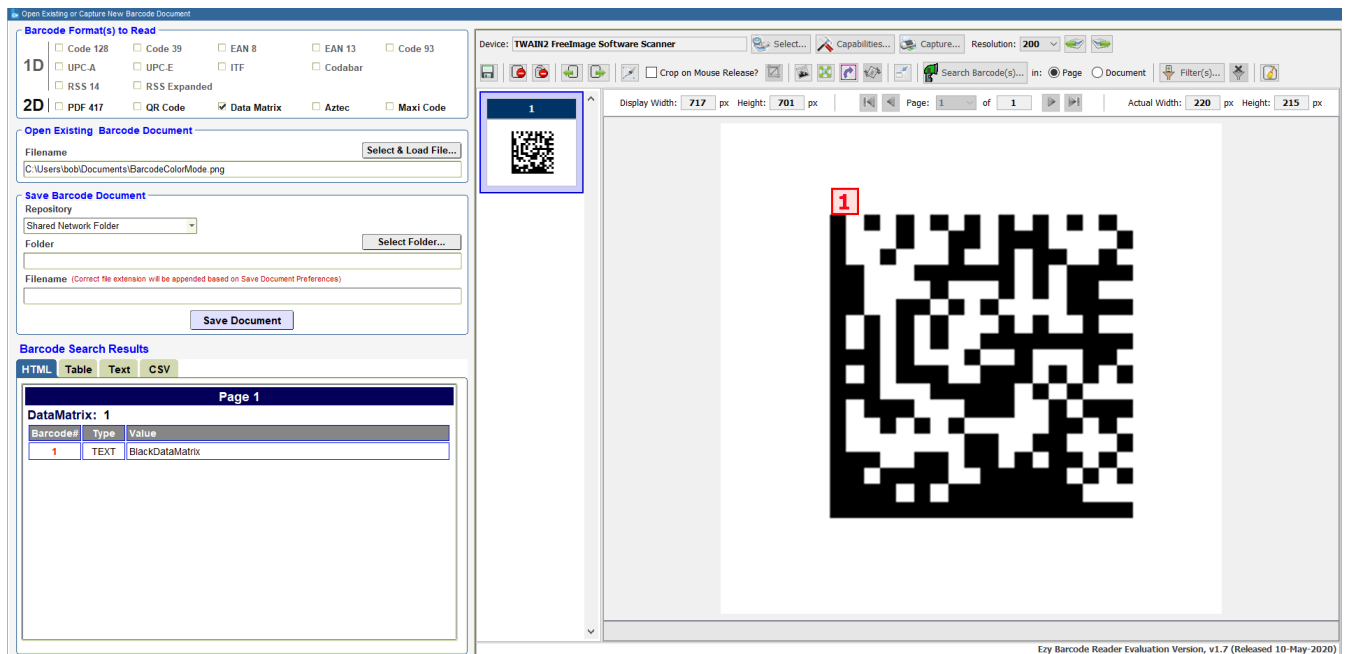


In such a situation, since we are only interested in decoding the '**Data Matrix**' barcode, which is the 4<sup>th</sup> one at the bottom of the document, as shown in the diagram below, it is best for us to use the '**Image Crop**' feature to only have this barcode document within the Image Viewer.

By pressing '**Enable Crop**' button and selecting the barcode image that we wish to be redisplayed within the Image Viewer, as shown below, and thereafter pressing '**Crop Image**' button as circled in **RED** within the diagram, we will then re-run the decode operation.



Upon pressing '**Crop Image**' button and re-running the decode operation, we will successfully get the result as shown below:

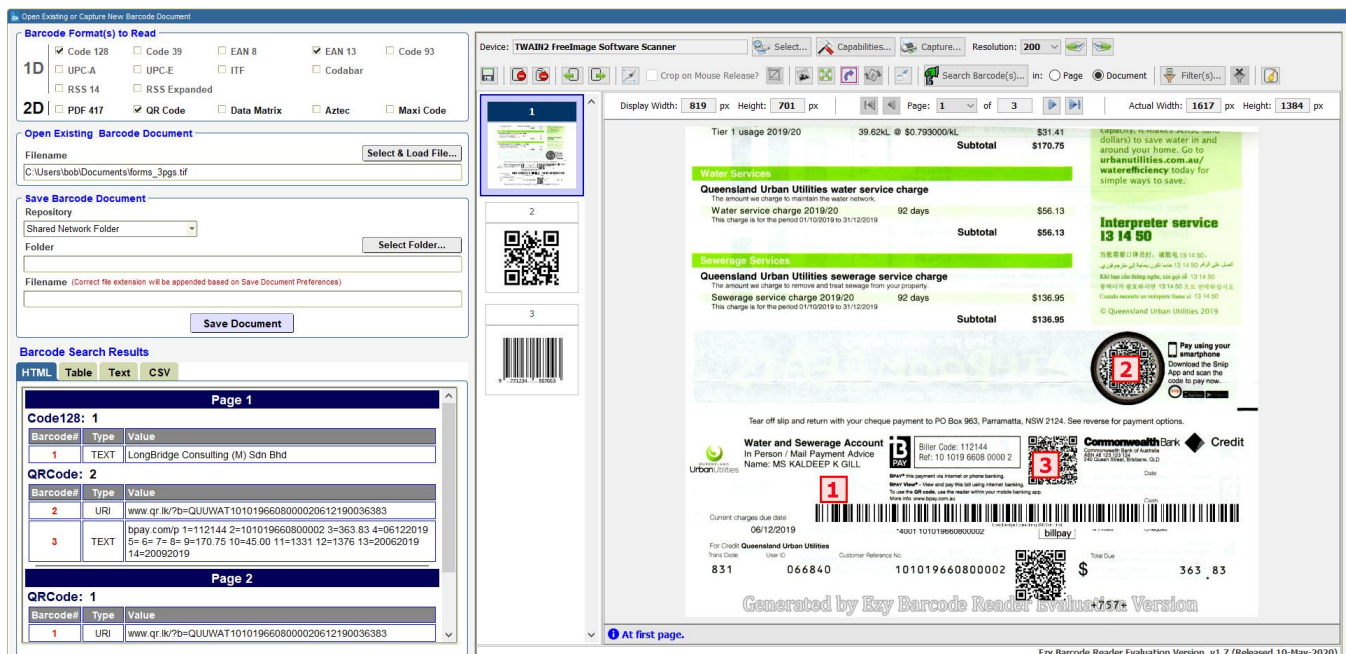


Ezy Barcode Reader is also smart enough to automatically select '**Page**' or '**Document**' radio button option upon loading a barcode document.

'**Page**' only allows a single page or the current page being displayed in the Image Viewer to be processed when a decode process is performed.

'**Document**' allows the Barcode Decode Engine to start from the 1<sup>st</sup> page, navigating page by page until the end of the document to generate the results found for all the readable barcode(s) within all the pages of the document.

Here's an example demonstrating a 3 pages TIF barcode document with its initial results generated, after having selected '**Code 128**', '**EAN13**' and '**QR Code**' checkboxes as required:



As can be seen on the 1<sup>st</sup> page, the Barcode Decode Engine did not detect the bottom 'QR Code' barcode. As such, we may have to do some Image Manipulation for this page to ensure that ALL 4 barcodes can be read successfully.

Since we are aware that we need a slightly whiter background to always ensure full detection of all barcodes, we can apply the 'Color / Levels' filter on this page and retry with 'Page' radio button selected before we press 'Search Barcode(s)...' button.

As you can see, after having adjusted the right-hand side slider knob to set it midway between 0.75 and 1.0 to whiten the background slightly, and re-running the decode process, we now have all 4 barcodes successfully read, as shown on the next page.

The screenshot shows the Ezy Barcode Reader interface. On the left, the 'Barcode Format(s) to Read' section has 'QR Code' selected. The 'Open Existing Barcode Document' section shows a file named 'forms\_3pgs.tif'. The 'Barcode Search Results' section shows the following data:

Barcode#	Type	Value
1	TEXT	LongBridge Consulting (M) Sdn Bhd
2	URI	https://si.ne/p/5ea43630-2c21-41df-8d36-f5e8a2a3e06
3	URI	www.qr.lk/7b=QUUWAT1010196608000020612190036383
4	TEXT	bpay comp 1=112144 2=101019660800002 3=363.83 4=06122019 5=6=7=8=9=170.75 10=45.00 11=1331 12=1376 13=20062019 14=20092019

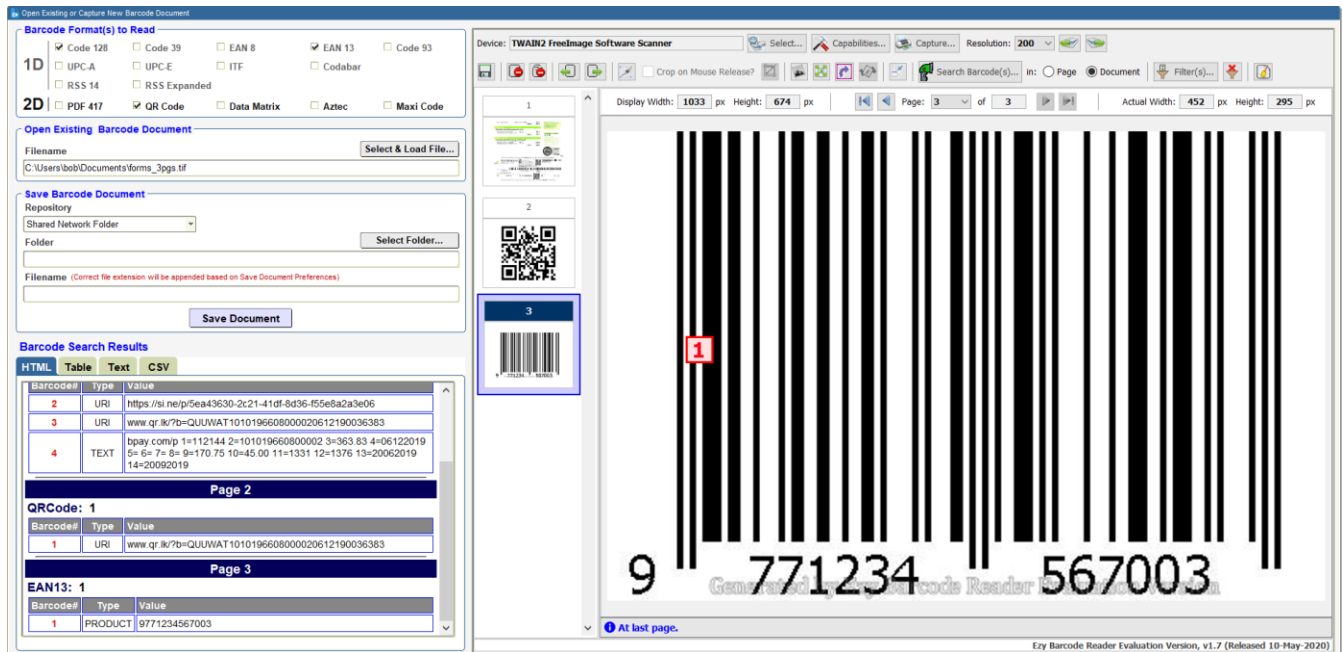
The main document area shows a utility bill for 'Queensland Urban Utilities' with a subtotal of \$170.75. It includes a QR code and a barcode at the bottom.

We can now set 'Document' as the radio button option, and press 'Search Barcode(s)...' button to read all the barcodes within the document again. Since the above already shows the results of the 1<sup>st</sup> page, below are the results of the barcodes detected and read successfully for the 2<sup>nd</sup> and 3<sup>rd</sup> pages.

The screenshot shows the Ezy Barcode Reader interface. On the left, the 'Barcode Format(s) to Read' section has 'QR Code' selected. The 'Open Existing Barcode Document' section shows a file named 'forms\_3pgs.tif'. The 'Barcode Search Results' section shows the following data:

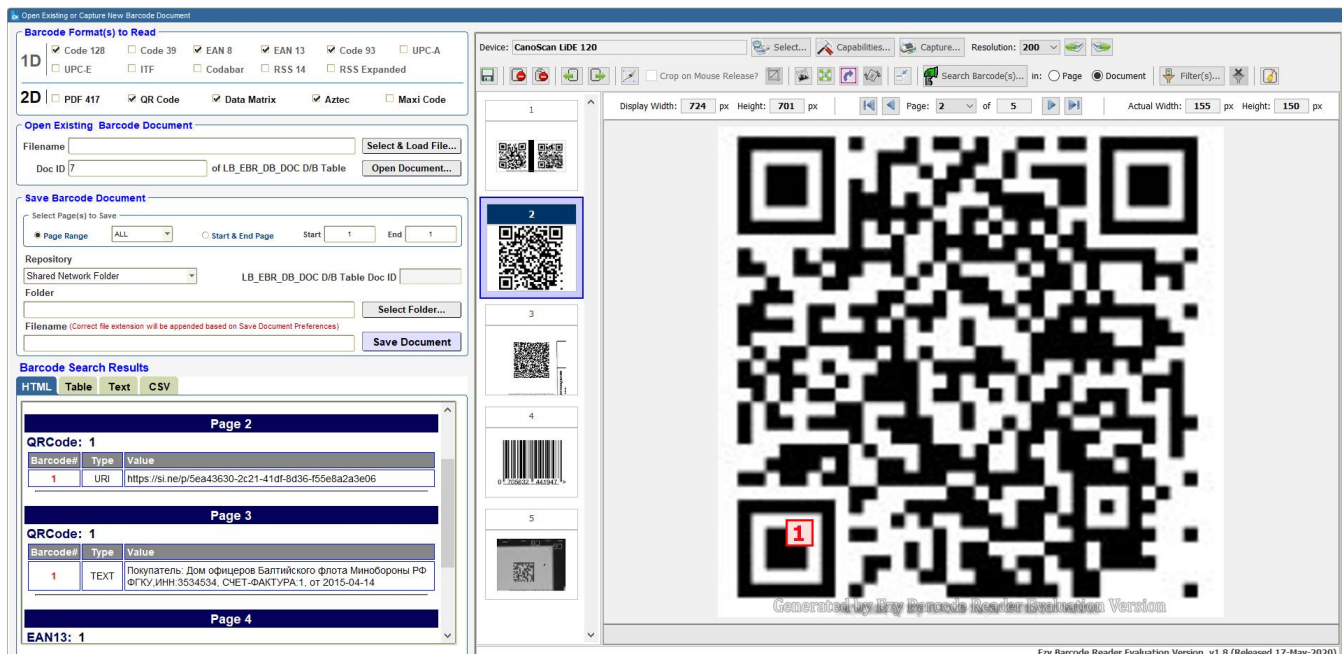
Barcode#	Type	Value
2	URI	https://si.ne/p/5ea43630-2c21-41df-8d36-f5e8a2a3e06
3	URI	www.qr.lk/7b=QUUWAT1010196608000020612190036383
4	TEXT	bpay comp 1=112144 2=101019660800002 3=363.83 4=06122019 5=6=7=8=9=170.75 10=45.00 11=1331 12=1376 13=20062019 14=20092019

The main document area shows a utility bill for 'Queensland Urban Utilities' with a subtotal of \$170.75. It includes a QR code and a barcode at the bottom.



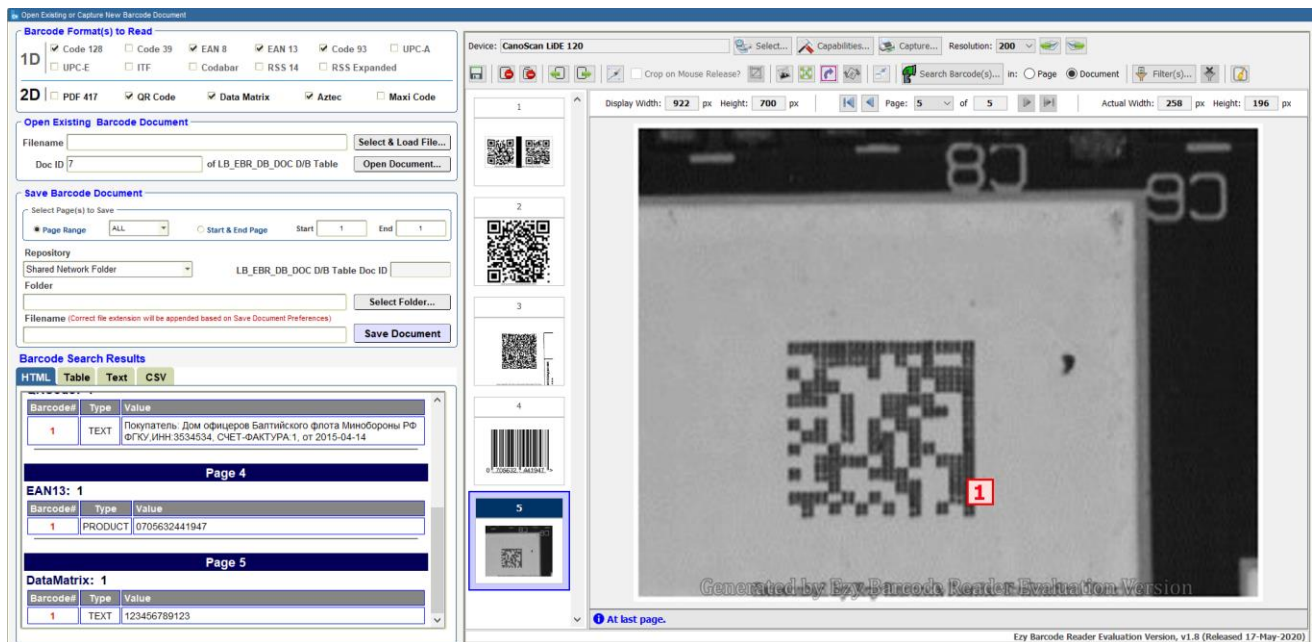
You may navigate to any page within your document to view the results of the decode operation, and the Image Viewer will always show the Barcode Number highlighted in **RED** outline for each barcode detected, as long as you have not performed any other process that automatically clears the recent barcode results.

For a multi-page document with multiple barcodes within some of the pages of the document, for example as shown below, we have provided the ability to **automatically scroll to the relevant HTML Page**, so that it is always in view, whenever navigation to a certain page within the Image Viewer is triggered.



For example, clicking the Thumbnail for Page 2, or using any of the other Page Navigation methods available, to display its page in the Image Viewer, it would also automatically scroll the HTML Search Results to bring Page 2 into view. This is a huge time saver, instead of the user needing to manually scroll to the required page.

Pressing 'End' keyboard key (accelerator shortcut key), will display Page 5 of the barcode document in the Image Viewer, while bring the page into view within the HTML Search Results tab folder, as shown below:



Finally, once we have achieved the correct results that we are looking for within a barcode document, especially after using certain Image Manipulation features, we could always save the new updated document that contains the correct barcode image(s) with any changes that we have done by using all the above mentioned features available. By doing so, we do not need to redo all the image manipulation actions again as it would be a time saver to now load the newly updated document to speed up your barcode recognition process.

To do this for the above example, we need to enter the relevant data into the 'Save Barcode Document' region within the sample Oracle Forms program provided, as shown below.

**Save Barcode Document**

Select Page(s) to Save

☒ Page Range  ☐ Start & End Page Start  End

Repository

Filename (Correct file extension will be appended based on Save Document Preferences)

The region above provides 2 options to define which page(s) that should be saved into the relevant repository. In this sample application, database barcode documents will be saved into **LB\_EBR\_DB\_DOCS** table, if you choose 'Database Server' from the Repository dropdown list. As for 'Shared Network Folder' or 'Local Client', it will be saved within the specified folder as per correct values specified within 'Folder' and 'Filename' fields.

'Page Range' option has the following selections:

- 'ALL' to allow you to save all the pages
- 'FIRST' to save only the first page
- 'LAST' to save only the last page

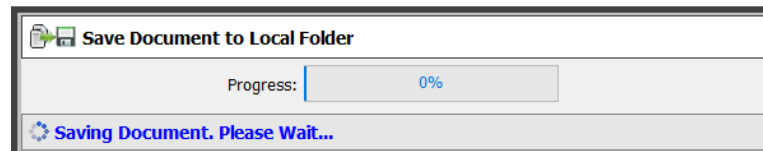
**‘Start & End Page’** option allows you to specify the Start Page Number and End Page Number from the list of page(s) that are within the displayed document.

The 2 options provide you with the flexibility of storing all the pages in one parse and then to selectively save different page range into different repository locations with a different image format, if required.

Our solution also allows you to save into other supported repositories such as **‘Application Server URL Folders’**, and/or **‘FTP Server Folders’**. Kindly request whenever required.

To save the current document into the **Local Client** folder, as an example, select or type in the Folder and the Filename (do not enter file format as it will be automatically derived based on your **‘Save Document Preferences’** setup values.

Pressing the **‘Save Document’** button will then validate the data to ensure of its completeness before a Progress Bar showing the save operation taking place to save your new document into the designated folder location.



Should your document be successfully saved into the designated folder location, you would see the following message in the Status Line of the Image Viewer:

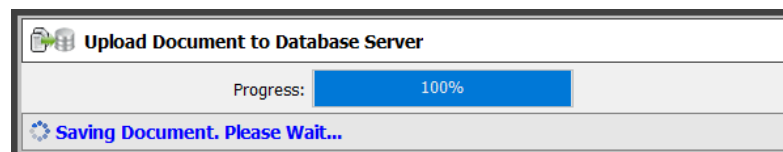


Should there be an error if you had wrongly specified the folder name, you should see the following error displayed in the Status Line of the Image Viewer:



To save the current document into the **Database Server table, LB\_EBR\_DB\_DOCS**, having selected it from the Repository dropdown list, where the non-relevant fields such as ‘Folder’ and ‘Filename’, as well as ‘Select Folder...’ button are disabled, just press **‘Save Document...’** button.

The progress dialog as shown below will be displayed:



On completion of the above process, you would see the value of **‘DOC\_ID’** database column of **LB\_EBR\_DB\_DOCS** displayed within its field, as shown below:

Confirmation that the document has been successfully saved into the database server table would be displayed in the Status Line of the Image Viewer, as follows:

**i Document has been successfully uploaded into your Database Server Repository.**

## **SUMMARY**

**Ezy Barcode Reader** is the ONLY Oracle Forms Java Bean solution in the marketplace currently that has comprehensive features that handle the capture and loading of barcode documents to detect and read most of the common 1D and 2D barcodes.

The fact is that our solution is very unique compared to other SDK solutions in the marketplace where you would need to create your own Java User Interface as part of the Java Bean component, and also integrate its APIs with your Oracle Forms applications. This makes our solution far more superior and far easier to implement in a very short time, as we provide both the ready-made Java Bean components with its User Interface as well as the sample Oracle Forms application from where a customer can use it as a baseline to copy and paste the relevant code to fully implement the necessary features needed within your own applications.

Coupled with all the Image Manipulation features that no other vendor readily provides in their solution to ensure that most barcodes, as long as they are not poor quality images of it, can be read successfully after performing the relevant tasks, it makes our solution as the most economical and provides the benefit of huge time savings as compared to having to develop similar features that our solution provides.

Our solution helps to improves workflow efficiencies, reducing human-error factors and enhances the robustness of captured data, which ultimately helps to improve barcode recognition accuracy to meet the business objectives that your organization is looking for within such a comprehensive, but yet a simple solution.