DATALOGIC™

# OCR

## User's Manual

DATALOGIC™

Datalogic Automation S.r.l.
Via Lavino, 265
40050 - Monte S. Pietro
Bologna - Italy

OCR User's Manual

Ed.: 09/2011

Datalogic shall not be liable for technical or editorial errors or omissions contained herein, nor for incidental or consequential damages resulting from the use of this material.

13/09/11

# CONTENTS

# REFERENCES

## REFERENCE DOCUMENTATION

This manual is designed to give a general overview to using the OCR feature with Datalogic Matrix Family readers. It covers application types, general and specific setup procedures and configuration steps.

For further details refer to the supplementary documentation provided on Mini-DVD:

- VisiSet™ OCR Help On Line
- OCR Software Configuration Parameters Guiide (pdf version of Help)
- DLReader User's Manual
- Matrix Host Mode Programming

## SERVICE AND SUPPORT

Datalogic provides several services as well as technical support through its website. Log on to **www.automation.datalogic.com** and click on the links indicated for further information:

- **PRODUCTS**

  Search through the links to arrive at your product page which describes specific Info, Features, Applications, Models, Accessories, and Downloads including the VisiSet™ utility program, which allows device configuration using a PC. It provides RS232 and Ethernet interface configuration.

- **SERVICE**

  - **Overview** **-** Warranty Extensions and Maintenance Agreements
  - **Sales Network-** Listing of Subsidiaries, Repair Centers, Partners
  - **Helpdesk**
  - **Material Return Authorization**

# 1 CHOOSING A CONFIGURATION

## 1.1 APPLICATION TYPES

| Application Type | Suggested Configuration |
|---|---|
| Applications where there is prior knowledge over the position of the ROI containing the string to be recognized through OCR.<br>Small shifts and rotations of the ROI for less than 2° are generally tolerated. | OCR with absolute ROI |
| Applications where the position of the ROI can change in the horizontal and/or vertical directions but in a limited amount.<br>A single large ROI should not be used in this case as it is faster to define a ROI that covers only the first character and from this determine the position for the ROI containing the entire string.<br>Rotations of less than 2° are generally tolerated. | OCR with localization through a character or a logo glyph |
| Applications where a 1D code is present.<br>It is assumed that the string to be recognized and the code have no relative rotation.<br>Shiftings or rotation will be tolerated as long as both the string and the code remain inside the acquired image<br>Note: This application requires that distances (DeltaX and DeltaY) between the centre of the code and the centre of the ROI where the OCR is performed are invariant . | OCR with localization through a 1D code |
| Applications where a 2D code is present.<br>It is assumed that the string to be recognized and the code have no relative rotation.<br>Shiftings or rotation will be tolerated as long as both the string and codes remain inside the acquired image<br>Note: This application requires that distances (DeltaX and DeltaY) between the centre of the code and the centre of the ROI where the OCR is performed are invariant . | OCR with localization through a 2D code |
| Typical OCV Applications such as performing OCR and matching the expected string. | OCR with absolute ROI<br>(Any supported configuration) |
| Applications where the position of the ROI can change in the horizontal or vertical directions but over a limited range and where vertical or horizontal bar/line/stripe is present (they will serve used for edge detection).<br>Less than 2° rotations are tolerated. | OCR located through one edge |
| change in the horizontal or vertical directions but over a limited range and where vertical or horizontal bars/lines/stripes are present (they will serve used for edge detection).<br>Less than 2° rotations are tolerated. | OCR located through two edges |

## 1.2 CHOOSING A PROPER MATRIX MODEL FOR OCR

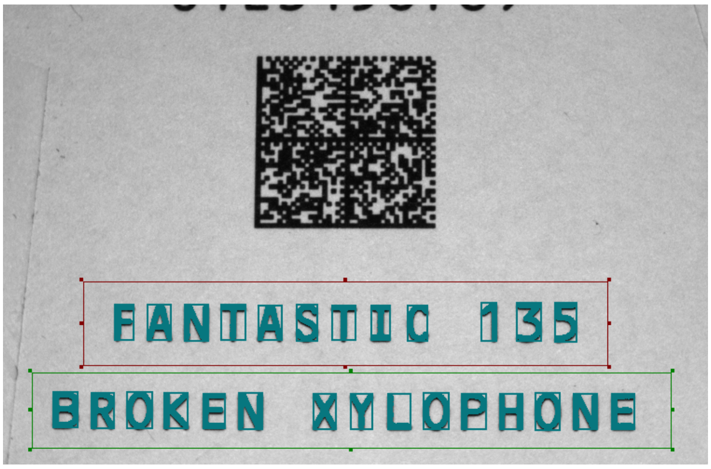
Choosing the correct Matrix reader model to be used in OCR applications largely depends on the size of the printed character and the desired FOV for the application.

The OCR algorithm is optimized for an average character height in the range [33..35] pixels with a minimum height of 28 pixels.

These requirements should be taken into consideration when choosing the Matrix reader model especially when Matrix 210™ is involved.

| **Minimum Height 28 pixels** | **Suggested Height 35 pixels** | **Image Size** | **Focus Distance** | **Matrix 210™ Model** | **PPI** | **Field of View (mm)** | **Pixels per mm = PPI/25.4** |
|---|---|---|---|---|---|---|---|
| 0.62 mm | 0,77 mm | 752x480 | 30 mm | 214 UHD | 1150 | 16.5 x 10.5 | 1150/25.4=45.2 |
| 1.3 mm | 1,63 mm | 752x480 | 40 mm | 211 NEAR | 545 | 35 x 22 | 545/25.4=21.45 |
| 1.87 mm | 2,34 mm | 752x480 | 65 mm | 212 MEDIUM | 380 | 50 x 32 | 380/25.4=14.96 |
| 2.98 mm | 3.73 mm | 752x480 | 105 mm | 213 FAR | 238 | 80 x 50 | 238/25.4=9.37 |

Note that the OCR string requires a quite zone of at least 10 pixels in all sides.



# 2 OCR FEATURE

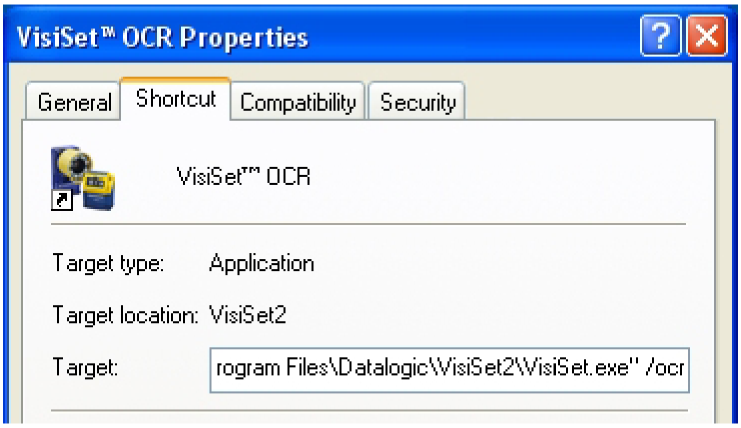
The configuration of the OCR feature on the Matrix reader involves two software tools:

- DLReader: for configuration of the inspection, font training and saving of the OCR-data files (the entire OCR parameter set is available and therefore the complete potential of the library).
- VisiSet™: allows for normal configuration of the Matrix reader but also introduces the OCR menu: this menu is where you can start the OCR configuration tool (DLReader) and then download the configuration file for the inspection to the Matrix reader, thus activating the OCR function.

The OCR function inside Matrix products is subject to a licensing mechanism. The OCR function is therefore disabled by default.

To show the OCR menu and the OCR command menu in VisiSet™, the application must be run with the command-line switch " /ocr". All the OCR parameters and commands are described in the OCR Help On Line and the relative Software Parameter Configuration Guide.

To achieve this, a new shortcut can be created on the PC desktop to launch Visiset.exe, then it can be modified as shown in the image. To create a new shortcut, right-click on the Visiset shortcut icon in the Windows Start Menu, then select 'Create Shortcut'. A new shortcut will be created in the same location as the current Visiset shortcut icon. To customize the new shortcut as in the image, right-click the newly created icon and select 'Properties'.



VisiSet™ OCR

Target type: Application

Target location: VisiSet2

Target: rogram Files\Datalogic\VisiSet2\VisiSet.exe" /ocr

**NOTE**

*In the VisiSet™ application the OCR/OCV function is treated as a symbology and can be used during reader configuration just like other symbologies for example as an option in Data Collection>Code Filter Settings.*

## 2.1 DLREADER TOOL

DLReader can be run from VisiSet™ menu toolbar under “OCR”-> “Run DLReader.
The only way to configure an OCR application is by using this tool and follow the instructions contained in the DLReader manual [1].
DLReader can load any bitmap image format/resolution but the resulting DLR file relates to a specific format.
The same DLR file can also be loaded on multiple readers that support the same image format/resolution (eg., two Matrix 210™s) whereas it is NOT possible to use the same DLR file on readers that have different image formats (such as a Matrix 210™ and a Matrix 410™).

| Type | Img Dim | Notes |
|---|---|---|
| Matrix 210 | 752x480 | Check also the image dimension (Parameter SetUp -> Calibration) |
| Matrix 410 | 1280x1024 | Check also the image dimension (Parameter SetUp -> Calibration) |
| Matrix 410-600 | 1600x1200 | Check also the image dimension (Parameter SetUp -> Calibration) |

It also must be checked the compatibility of the DLR file with the set-up of the Matrix reader where it is loaded. Indeed a file created for a DLR Matrix 210™ FAR (model 213) may have a ROI which turns out to be too small for a Matrix 210™ MEDIUM (Model 212).

### 2.1.1 Installing DLReader

DLReader comes inside a ZIP archive file that also contains some library files necessary to make it work
The full content of the zip archive must be extracted to the VisiSet™ installation directory path (typically **C:\Program Files\Datalogic\VisiSet2\)**

If the extraction PATH is not **C:\Program Files\Datalogic\VisiSet2**\, DLREADER fails to launch from VisiSet™.
Expert Windows’ Users can open the registry key [HKCU]\Software\Datalogic\Visiset and look at the value of the ‘FullPath’ key. There is where the content of the DLReader package has to be extracted in order to make it work seamless with VisiSet™.

**CAUTION**

*Editing the windows registry could damage your Windows installation, please proceed with editing the registry only if you know what you are doing.*

## 2.2 LICENSING THE OCR FEATURE

As mentioned before, the OCR feature of the Matrix product is a licensed feature, disabled by default: this means that before the user is be able to use it, a valid OCR unlocking license key has to be supplied to the reader.

The licensing mechanism is performed in two steps and requires VisiSet™ and the reader to be running:

1. first a ‘Reader Registration Code’ has to be produced from the Matrix and saved to the local PC as a text file. This can be done using VisiSet™ under the “OCR\Download Reader Registration Code” menu
2. The saved file has to be shipped to a Datalogic subsidiary where it will be modified and returned to the user as a text file containing the ‘Unlocking License Key’. (NOTE: Datalogic may require the payment of a licensing fee or the enrolment into a OCR licensing program, please contact your local sales representative for additional information).
3. Upon receiving the unlocking license key file, it has to be uploaded to the Matrix reader via the “OCR\Upload License Key” menu
4. On successful feature unlocking, the reader will reset and the feature made available

## 2.3 OCR APPLICATIONS CONFIGURATION STEPS

In order to design and run your first OCR application with a Matrix reader, you will want to follow a sequence of steps involving the OCR engine configuration application (DLReader), the Matrix reader configuration application (VisiSet™) and some 'playing' with sample images. Following is a list of activities to guide the user in the process of designing and running an OCR application.

1. Start off with some samples and with the Matrix reader

   First make sure you have some good quality samples for your OCR application, make sure you place them under the Matrix reader and you set up the Matrix using VisiSet™ (Calibration, Setup Wizard) to make it acquire images whose brightness and contrast match your application's true operating conditions. Once you are satisfied with the result of the acquisition, save this image (or the images set[1] needed for the application) to the local file system and keep it/them for future reference: we will be referring to this image as **The Reference Image(s)**. Of course it is possible to change the reference image for the application at any time, but that will to go through the sequence of operations being described in this section.

2. Configure the OCR engine using DLReader

   When you are satisfied with the quality of the acquired image, you can run DLReader application (via the Ocr->Run DLReader menu entry in VisiSet™). The configuration procedure begins with the opening of The Reference Image and continues further with the setup of the OCR engine parameters. This process is described in its entirety in the DLReader User's Manual and completes with the output from DLReader of a '.DLR' configuration file.

3. Upload the configuration to the Matrix reader

   Now let us go back to the VisiSet™ configuration application: the produced '.DLR' configuration file is uploaded to the Matrix reader via the 'Ocr->Upload Configuration File' menu entry. This file is saved into a dedicated flash area of the Matrix reader and the upload and storage process follows a similar sequence as with matrix binary files (similar sequence of messages are shown inside VisiSet™'s terminal screen) and completes with the message 'Configuration Saved in Permanent Memory!'.

4. Configure Matrix to work with the 'OCR' symbology

   Once the configuration for the OCR has been successfully uploaded (stored) on the Matrix reader, the configuration of OCR-related parameters via VisiSet™ begin first by enabling the OCR functionality (OCR tab \ OCR OCV \ Status => enabled).

5. Run the OCR application

   When all the steps above have been executed, the Matrix reader is ready to run the OCR OCV application in the usual way (triggering, output, etc.). If you wish to refine the settings, you can either make some modifications to the Matrix configuration under the "OCR tab \ OCR OCV" section or if you need to start off a new Reference Image you will have to repeat the sequence of steps starting with 0, making sure that in 2 you load the NEW Reference Image and you modify the existing configuration by first loading it into DLReader (Ocr -> Restore Reader Data)

[1] NB: it might be necessary to acquire more than one sample image to exploit the complete Font used along the application. In general with OCR it is true the assumption: the more sample images, the more training, the better the result outcome.

## 2.4 OCR AND EDGE SYMBOLOGIES

The output of the Optical Char Recognition feature is associated with two symbologies inside the Matrix:

- “OCR OCV” Symbology for all the output related to the configured ‘readers’;
- “EDGE” Symbology for all the output related to the configured EDGE ROI’s.

## 2.5 OCR REFERENCE IMAGE

The OCR feature uses one (or more) reference images for setup which is transferred within the DLR file and stored in the Matrix FLASH memory; it is also used to initialize all the VisiSet™ OCR parameters using the .DLR file configuration.

All the following OCR aspects configuration is entirely performed using DLReader and do not require modifications to the current VisiSet™ configuration:

- “Absolute ROI”,
- “Localization by Blob”,
- “Localization by Edge”.

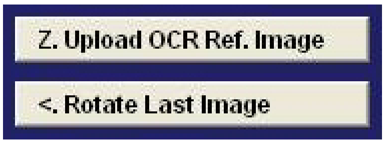
In the above cases, it is not necessary to upload an image to the reader and decode it in order to initialize the OCR functionality.

The only exception to this is the "Localization by Code" configuration that requires the User to:

- configure one or more VisiSet™ OCR parameters and
- perform an upload of an image and
- perform a decoding over the image in order to properly initialize all the OCR features.

There are two command buttons specific to the OCR that are shown only when the reader has a valid OCR License and the OCR Feature is set to Enabled:

- **Upload OCR Ref. Image** (this is shown only if a valid DLR file is present inside the Matrix’s FLASH).
- **Rotate Last Image.**



Instead of using the “Upload the Bitmap Image” button, the user can use the new button "Z. Upload OCR Reference Image" to automatically load the reference image into the reader’s system memory. Once this has been done, OCR identification can be run.

The Rotate Last Image button is meaningful only when a code is present: it uses decoded code rotation info in order to rotate the image to obtain a code orientation of 0 degrees.
This can be very useful in two cases:

- To create a reference image where code rotation = 0 degrees. This guarantees the success of the 'Localization by Code' Action.
- To train the OCR functionality on characters which were not recognized during the Training Phase because they were rotated.

# 3 CONFIGURING AN OCR APPLICATION

## 3.1 GENERAL OCR CONFIGURATION STEPS

### 3.1.1 Creating an OCR Configuration

An OCR application with a Matrix reader can be configured using the following procedure:

**VisiSet™**

1. Capture an image and save it to a PC (test.bmp).
2. Launch **DLReader** (the OCR Configuration GUI).

**DLReader**

3. OCR Configuration
   a) Load the previously saved image (test.bmp) (Open command).
   b) Configure one or more reader ROIs boxing the strings to be read.
   c) Tune the OCR engine configuration if necessary.
   d) Train for the Fonts.
   e) Save the results to file (test.dlr) containing both the configuration and the trained Font.

**VisiSet™**

4. Open the Parameter Setup dialog (using the Get Configuration From Temporary Memory command of the Device Menù) and enable the OCR function (Status parameter of the OCR tab).
5. Upload the configuration file and Font (test.dlr) to the Matrix reader using the Upload the configuration file command of the OCR menu. The loaded DLR file also contains the OCR Reference Image.
6. When needed, open the Parameter Setup dialog and modify the OCR configuration (for example to enable the Localization by Code).
7. Run an initial decoding to initialize the OCR parameters, if required.
8. Save the configuration to permanent memory (using the Send button of the Parameter Setup dialog) if you want the configuration to persist upon resets of the Matrix reader.

In cases where font training is needed over multiple images, you will want to repeat steps 1 through 3.

***Tip***: Save all the images where the font has been trained under a different directory for each application, together with the respective DLR generated file and the VisiSet™ configuration, after the first capture.

### 3.1.2 Deleting an OCR Configuration

An OCR application can be deleted from a Matrix reader using the following procedure:

**VisiSet™**

1. Delete the configuration and Font file (test.dlr) on Matrix reader. To delete the OCR configuration, select "Erase Configuration File" from the ***VisiSet™ OCR menu***.
2. Open the Parameter Setup dialog and disable the OCR function (Status parameter).
3. Save the configuration to permanent memory.

### 3.1.3 Check & Modify the OCR Configuration

The Quality Level, the OCR Match String and the ROI position (already configured by DLReader) are also modifiable using the ***VisiSet™*** Parameter Setup dialog. They are visible enabling the OCR Advanced View parameter. Note that the Quality Level initially reflects the value of the Image Fail parameter set in the DLReader configuration.

The enabling of the Show All OCR ROIs parameter causes the ROI to be drawn on the captured image so you will also be able to see them. This parameter is very useful for troubleshooting especially to verify the ROI position in case of rotated ROIs (OCR located by 1D/2D barcode).

Any changes to the ***VisiSet™*** Parameter doesn't affect the .dlr file.

## 3.2 OCR WITH AN ABSOLUTE ROI

### DLReader

1. Create as many reader ROI objects as there are LABELS to be read.
2. No level of acceptance should be set 0.
3. Run Font training.
4. Save the DLR file to the file system.

### VisiSet™

After loading the DLR file on the Matrix reader (configuration, Font and reference image), the OCR application is ready and initialized.

5. Verify that the value for the OCR Type Configuration parameter becomes “Absolute ROI.”
6. If the Matrix is reset, the OCR configuration will persist (if previously saved to permanent memory).
7. In the case of a Matrix reader SW upgrade, without explicitly Sending the Default Configuration parameters, the OCR settings persist.

## 3.3 OCR LOCALIZED BY A CHARACTER/LOGO

### DLReader

1. Create a Reader ROI object to locate the Character/Logo.
   - assign a name to the created reader object (for example, “blob_loc”)
   - Set its level of acceptance to 0.
   - NOTE: for the localization, ONLY the leftmost character in the defined reader ROI is considered.

2. Create as many reader ROIs as there are LABELS to be read.
   - Their level of acceptance shouldn't be 0.
   - In the ‘link’ field select the value “blob_loc” for everyone of them.
   - Run Font training on every added reader.

*After this, any change in the position of the first Character/Logo on the left side the reader ROI of “blob_loc”, will result in a change in the position of the reader ROIs used for the recognition of the other strings.*

3. Save the DLR file to the file system.

### VisiSet™

After loading the DLR file (configuration, trained Font and reference image) in Matrix reader, the application is now ready and initialized.

4. Verify that the value for the Configuration parameter becomes “Located by Blob.”

   **At this point, any position differences in samples, will be accepted within the limits of the application as it has been configured.**
5. Save to permanent memory.
6. If the Matrix reader is reset, the OCR configuration will persist (if previously saved).
7. In the case of a Matrix reader SW upgrade, without explicitly Sending the Default Configuration parameters, the OCR settings persist.

If the information has not been saved in permanent memory and a restart occurs, it is sufficient to upload the OCR Reference Image and run a first reading on it to properly initialize all the data structures related to the OCR and localization functions.

## 3.4 OCR LOCALIZED BY A 1D/2D CODE

### DLReader

1. Create as many reader ROI objects as there are strings to be read.
2. Run Font training.
3. Save the file DLR to the file system.

### VisiSet™

After loading the DLR file (configuration, Font and reference image) on the Matrix reader, the application is ready but still in Absolute ROI. To configure the code location some other action should be performed:

4. Open the parameters window using the Device -> Get configuration from Temporary Memory command from the main menu. Do not enter the Calibration Tool wizard.
5. Check the 1D/2D/Postal Codes tabs and make sure that ONLY the code type present on the reference image is enabled in VisiSet™ configuration.
6. Set the OCR parameter ‘Localization by Code’ to the value “Enabled”.
7. Save the parameters in Temporary memory and close the dialog.
8. Upload the reference image using the new button “z.Upload OCR Ref. Image”: it will retrieve image information from the Matrix FLASH memory where it has been stored with the DLR file information.
9. Run an initial reading (VisiSet™ button c.Decode Last Image) to initialize any data structures related to the OCR.
10. In VisiSet™ reopen the configuration using the Device -> Get configuration from Temporary Memory command from the main menu and verify that the value for the OCR Type Configuration parameter has changed to “Localized by Code”.
11. **At this point, any position differences in samples, will be accepted within the limits of the application as it has been configured.**
12. Save parameters in permanent memory and close the window.
13. If the Matrix reader is reset, the OCR configuration will persist (if previously saved).
14. In the case of a Matrix reader SW upgrade, without explicitly Sending the Default Configuration parameters, the OCR settings persist.

If the information has not been saved to permanent memory it is sufficient to upload again the reference image run the sequence of operations over it as described above.

### 3.4.1 Font Updates

The OCR training phase can require the addition to the Font of characters not yet recognized.
The DLReader tool accepts only horizontal or vertical strings so if the unrecognized chars belong to a case where the string is rotated, an unrotate action must be performed before loading the image in the DLReader tool.

In this case it is possible to unrotate the Image using the ‘Rotate Last Image’ command; it produces a new image where the string is orientated like the reference image and it is possible to load it inside DLReader. Using DLReader the Font can be upgraded as needed, then the usual application configuration steps take place.

### 3.4.2 Disabling OCR Localization by Code

#### VisiSet™

To remove the reader ROI objects **Localization by Code**:

1. Open the parameter window
2. Set the OCR parameter ‘Localization by Code’ to the value “Disabled”.
3. Save parameters to Temporary memory and close the window.
4. Trigger a read (VisiSet™ button c.Decode Last Image) so that all data structures for OCR can get correctly initialized.
5. Make sure that the value of the Configuration parameter returns to “Absolute ROI”.

## 3.5 OCR LOCALIZED BY EDGE

### 3.5.1 One Edge Present

1. Create as many EDGE ROI objects as there are edges to be recognized.
2. Create as many READER objects as there are LABELS to be read.

The Edge ROI object must be created before the Reader object(s) that it has to locate.

#### DLReader

3. Create 1 EDGE ROI object and assign it a name, for example “edge_loc ‘
4. Create 1 READER object to read the LABEL. It must be the smallest rectangle that contains the string.
5. The level of acceptance must be in the range 60-80.
6. In the link field select the “edge_loc” option.
7. Run Font training.
8. Save the DLR file to the local file system.

#### VisiSet™

After loading the DLR file (configuration, Font and reference image) in Matrix reader, the application is now ready and initialized.

9. Make sure that the value of the Configuration parameter turns to “Located by Edge”.
10. Save the parameters in permanent memory, close the window.
11. If the Matrix reader is reset, the OCR configuration will persist (if it has been previously saved).
12. In the case of a Matrix reader SW upgrade, without explicitly Sending the Default Configuration parameters, the OCR settings persist.

If the information has not been saved in permanent memory, it is sufficient to re-upload the reference image and run a sequence of operations on it as described above.

## 3.5.2 Two Edges Present

1. Create as many EDGE ROI objects as there are edges to be recognized.
2. Create as many READER ROI objects as there are LABELS to be read.

The Edge ROIs should always be created before the Reader ROI objects they have to locate.

### DLReader

3. Create 1 EDGE ROI object and name it, for example, “edge_loc1‘.
4. Create 1 EDGE ROI object; and assign it a name, for example, “edge_loc2‘. In the link field select “edge_loc1”.
5. Create 1 READER ROI object to read the LABEL. It must be the smallest rectangle that contains the string.
6. The level of acceptance must be in the range 60-80.
7. In the link field select “edge_loc2”.
8. Run Font training.
9. Save the DLR file in a system file.

### VisiSet™

After loading the DLR file (configuration, Font and reference image) in Matrix reader, the application is now ready and initialized.

10. Make sure that the value of the Configuration parameter turns to “Located by Edge”.
11. Save the parameters in permanent memory, close the window.
12. If the Matrix reader is reset, the OCR configuration will persist (if previously saved).
13. In the case of a Matrix reader SW upgrade, without explicitly Sending the Default Configuration parameters, the OCR settings persist.

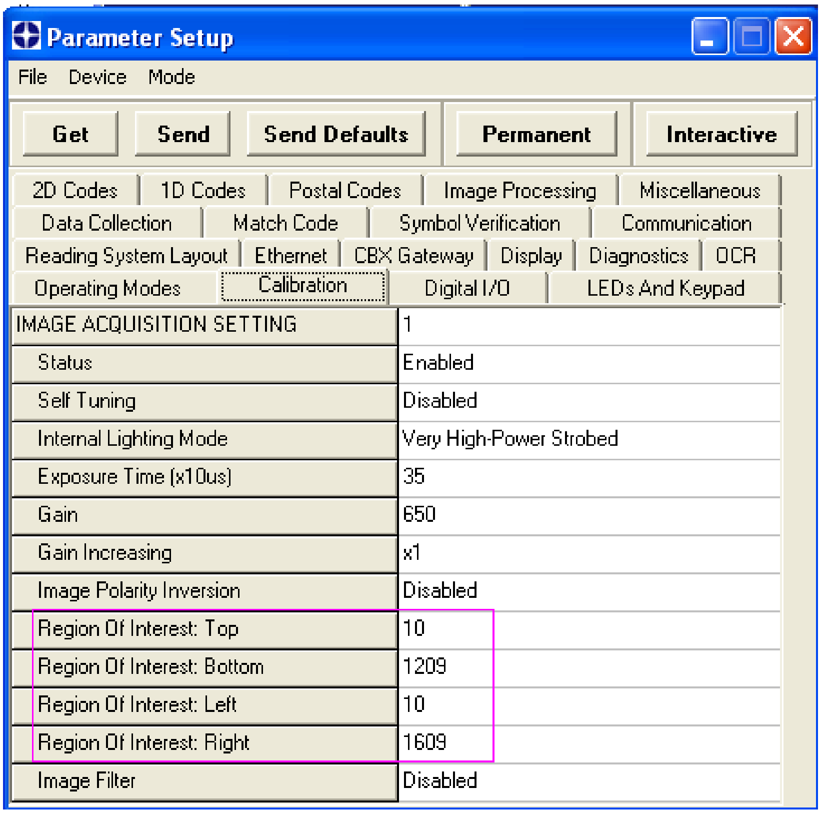
If the information has not been saved in permanent memory it is sufficient to re-upload the reference image and run a sequence of operations on it as described above.

# 4 TROUBLESHOOTING

## 4.1 CHANGE IN IMAGE DIMENSIONS (IMAGE CROP)

The DLR file created with DLReader internally stores the ROI position in absolute coordinates, according to the image that has been used during the configuration phase.

If AFTER having setup the OCR application you need to crop the processed image using values resulting from the Calibration tab of the Parameter Setup, you will have to re-set the information for OCR ROI objects.



| | |
|---|---|
| IMAGE ACQUISITION SETTING | 1 |
| Status | Enabled |
| Self Tuning | Disabled |
| Internal Lighting Mode | Very High-Power Strobed |
| Exposure Time (x10us) | 35 |
| Gain | 650 |
| Gain Increasing | x1 |
| Image Polarity Inversion | Disabled |
| Region Of Interest: Top | 10 |
| Region Of Interest: Bottom | 1209 |
| Region Of Interest: Left | 10 |
| Region Of Interest: Right | 1609 |
| Image Filter | Disabled |

This can easily be done as follows:

1. Load the cropped image inside DLReader
2. load in DLReader the configuration previously created (file DLR, menu entry ‘restore reader data’)
3. move the reader ROI object doing the OCR to the new location
4. save a **new** DLR file for the cropped image.

Follow the usual steps for the OCR configuration.

# INDEX

**Blob**
a group of dark pixels forming a ‘connected path’ over a light background.

**Crop**
obtaining a smaller image frame from a larger one.

**Edge ROI object**
an object performing Edge detection on its assigned region of interest. Inside a single image there can be one or more Edge ROI objects, each one of them will locate an “edge” in the image (if capable or successful).

**Font(s)**
the set of data associated with the training of the OCR engine over a set of known strings. The training process produces Font data which is used during recognition for the decision making process.

**FOV (Field Of Vision)**
the area covered by the reader's visual span.

**Localization/Locator**
identification of the location of a code/text/glyph inside an acquired image frame.

**Match String**
an expression giving priori knowledge about the type of output the OCR engine can expect from a reader ROI object (e.g. only numerical digits).

**Misread**
a situation where the OCR result is different from what a human reads but the OCR did not know it and believed its output to be correct.

**OCR (Optical Character Recognition)**
the process of automatically recognizing a sequence of characters/numbers.

**OCV (Optical Character Verification)**
similar to OCR but in this case there is a priori knowledge on the expected result. Focus is more on the verification process rather than on the identification process.

**Reader**
the device used to capture images (the Matrix).

**Reader ROI object**
an object performing OCR on its assigned region of interest. Inside a single image there can be one or more Reader ROI objects, each one of them will output the result of its recognition (if capable or successful).

**ROI (Region Of Interest)**
a bounded region to which an OCR operation refers.

**Score (characters)**
recognized character sequences are assigned a ‘score’ as an indication of the ‘confidence level’ the OCR engine has over what it has yelded.

**Training**
the process where the OCR engine is "taught" to recognize strings by feeding it with sample data and corresponding human-readable strings.

www.automation.datalogic.com