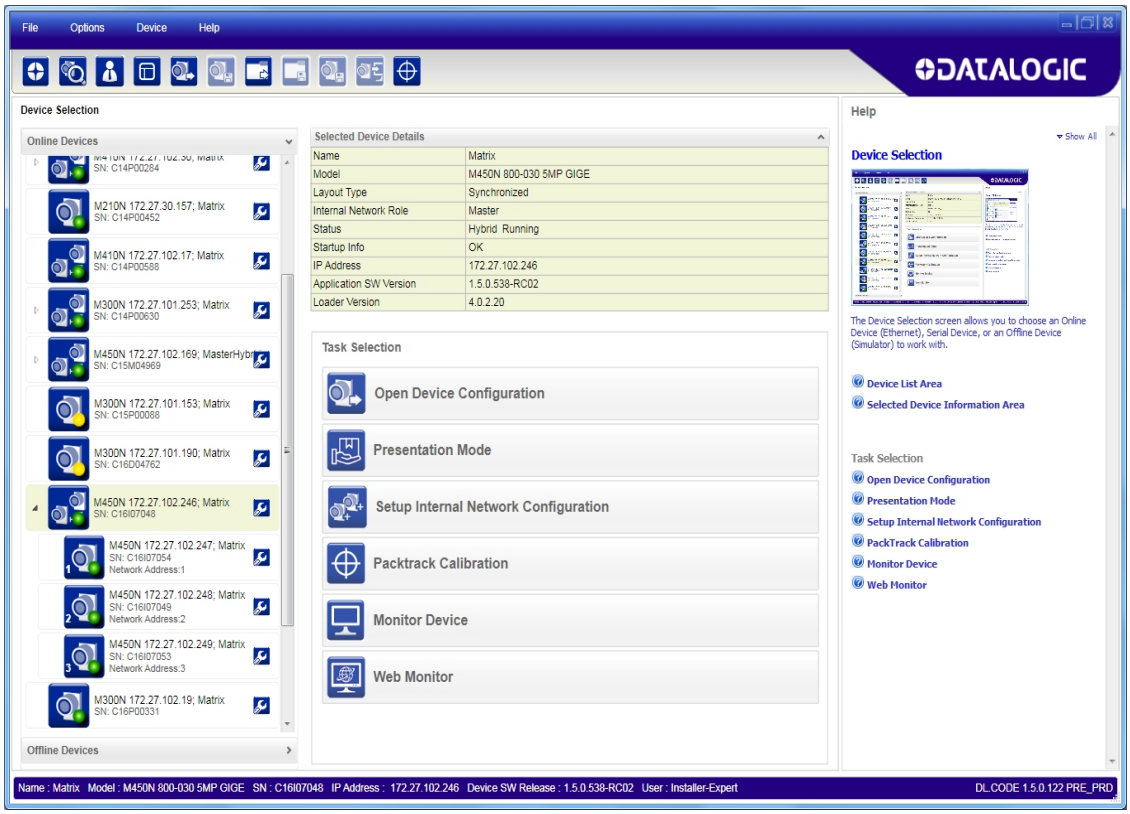
DL.CODE™

USER'S MANUAL



User Interface Client Application

DATALOGIC

Datalogic S.r.l.
Via S. Vitalino, 13
40012 Calderara di Reno (BO)
Italy
Tel. +39 051 3147011
Fax +39 051 3147205

This manual refers to software version 1.11.0 and later.

Disclaimer

Datalogic has taken reasonable measures to provide information in this manual that is complete and accurate, however, 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. Datalogic reserves the right to change any specification at any time without prior notice.

Trademarks

Datalogic and the Datalogic logo are registered trademarks of Datalogic S.p.A. in many countries, including the U.S.A. and the E.U.

DL.CODE, ID-NET, and X-PRESS are trademarks of Datalogic S.p.A. and/or its affiliates. Digimarc® and DWCODE™ are trademarks of Digimarc Corporation. All other trademarks and brands are property of their respective owners.

Patents

See www.patents.datalogic.com for patent list.

DATALOGIC END USER LICENSE AGREEMENT

NOTICE TO END USER: BY DOWNLOADING OR INSTALLING THE SOFTWARE, OR BY USING THE DATALOGIC PRODUCT THAT INCLUDES THIS SOFTWARE, THE END USER CONSENTS TO BE BOUND BY THIS AGREEMENT. IF THE END USER DOES NOT AGREE WITH ALL OF THE TERMS OF THIS AGREEMENT, THEN DATALOGIC IS NOT WILLING TO LICENSE THE SOFTWARE AND THE END USER IS NOT ALLOWED TO DOWNLOAD, INSTALL OR USE THE SOFTWARE OR THE DATALOGIC PRODUCT

This End User License Agreement ("EULA") is between Datalogic IP Tech S.r.l. having its registered office at Via San Vitalino 13, 40012 Calderara di Reno (Bologna), Italy (“Datalogic”), and you, either an individual or a single entity, (“End User or “You””) who has purchased one or more Datalogic automated data capture devices. This EULA applies to the DL.Code software (“DL.Code Software”) and to any other software, to the extent applicable, offered as optional application software (“Application Software”) that together with “DL Code Software” is herein referred to as “Software”). Any Open Source used in a conjunction with the Software is subject to the Open Source licenses available at the following website: http://www.datalogic.com/oss.

1. Grant of License

1.1 Datalogic grants to End User, a personal, non-exclusive, non-transferable, non-sublicensable, revocable, limited license to use the Software for reading device’s configuration, monitoring and diagnostic exclusively in conjunction with Datalogic automated data capture devices for which it is intended for use. Subject to the provisions of this Agreement, the Software could be, to the extent applicable, installed by End User and used at an End User’ single computer which shall be compliant with the system requirements described in the Datalogic Product documentation. The Software can be used only in machine-readable form only, solely for End User's internal business purposes.

This License does not allow the Software to exist on more than one computer at a time, and End User may not make the Software available over a network where it could be used by multiple computers at the same time, except for the monitoring features included in the Software which could be accessed simultaneously by more users within End User’s organization running the Software.

1.2 End Users shall not distribute, sublicense, rent, loan, lease, assign, export, re-export, resell, ship or divert or cause to be exported, re-exported, resold, shipped or diverted, directly or indirectly, the Software under this Agreement. End User shall not, and shall not permit others to: (i) modify, translate, decompile, reverse engineer, disassemble, or extract the inner workings of the Software, (ii) copy the functionality of the Software; (iii) remove any proprietary notices, marks, labels, or logos from the Software; (iv) rent or transfer all or some of the Software to any other party without Datalogic’s prior written consent.

1.3. Title to the licensed Software shall be and remain with Datalogic or the third party from whom Datalogic has obtained a license right. This Agreement does not grant to End User any intellectual property rights. As used in this Agreement the term “purchase” or its equivalent when applied to the Software shall mean “acquire under license”. End User is not entitled to receipt or use of the source code of any licensed Software.

1.4 Portions of the Software are protected by the relevant and applicable patent and copyright laws, international treaty provisions, and other applicable laws. Therefore, End User must treat

the Software like any other copyrighted material (e.g. a book or musical recording) except that End User may make one copy of the Software solely for back-up purposes. Unauthorized duplication of the Software constitutes copyright infringement.

1.5 Any use of the Software outside of the conditions set forth herein and in any ancillary Datalogic product documentation is strictly prohibited and will be deemed a breach of this Agreement resulting in immediate termination of this Agreement. In the event of a breach of this Agreement, Datalogic will be entitled to all available remedies at law or in equity (including but not limiting to immediate termination of the license without notice, immediate injunctive relief and block of the Software).

1.6 Without prejudice of the foregoing, End User grants to Datalogic and its independent accountants or consultants the right to examine End User's books, records and accounts during End User's normal business hours to verify compliance with this Agreement. In the event such audit discloses non-compliance with this Agreement, Datalogic shall be entitled to immediately terminate the Agreement, request End User to promptly pay all (additional) license fees due and any further damages, if any.

2. License Fee

License fees shall be due by End User to Datalogic according to the terms provided for in the relevant contract for the purchase of the Datalogic product.

3. Termination

Without prejudice to any other rights or remedies Datalogic may have, Datalogic may terminate this Agreement if End User fails to comply with the terms and conditions of this Agreement. Datalogic may terminate this Agreement by offering you a superseding Agreement for the Software or any replacement or modified version of or upgrade to the Software and conditioning your continued use of the Software or such replacement, modified or upgraded version on your acceptance of such superseding Agreement. In addition, either party may terminate this Agreement at any time. Subject to the foregoing, termination shall be effective upon notice to the other party. In the event that this Agreement terminates for any reason, End User’s license to use the Software will immediately terminate, and End User must immediately stop using the Software, destroy all copies of the Software and all of its component parts, and, upon request, provide an affidavit certifying your compliance with the foregoing. The provisions of Sub-sections 1.2, 1.3, 1.4, 1.5, 4, 5, 6, 8, and 12 shall survive termination of this Agreement.

4. Limited Warranty

Datalogic warrants that, under normal use and operation, the Software will conform substantially to the applicable Datalogic product documentation for the period specified in the same, Datalogic’s entire liability and End User’s sole and exclusive remedy for any breach of the foregoing limited warranty will be, at Datalogic’s option, the provision of a downloadable patch or replacement software.

Datalogic does not warrant (i) that Software will meet End User's requirements; (ii) that Software will be uninterrupted or defect error free; (iii) any non-conformity derived from unauthorized use, and/or improper installation or repair not in compliance with Datalogic product documentation; (iv) any back up of End User data; (vi) any particular security measures for data processing and exchange between the Software and End User systems. End User agrees that the existence of such non conformities or errors and Datalogic’s inability to remedy such errors shall not constitute a breach of this Agreement.

EXCEPT AS PROVIDED IN THIS AGREEMENT, THE DATALOGIC PRODUCT IS PROVIDED "AS IS" AND DATALOGIC MAKES NO WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, WRITTEN OR ORAL, WITH RESPECT TO THE SOFTWARE, AND SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

5. Limitation of Liability

EXCEPT AS PROVIDED IN THIS AGREEMENT, NEITHER DATALOGIC NOR ITS LICENSORS SHALL BE LIABLE FOR ANY CLAIMS AGAINST END USER BY ANY OTHER PARTY. IN NO EVENT SHALL DATALOGIC'S LIABILITY FOR DAMAGES, IF ANY, WHETHER BASED UPON CONTRACT, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY, STRICT LIABILITY, WARRANTY, OR ANY OTHER BASIS, EXCEED THE PRICE OR FEE PAID BY END USER FOR THE DATALOGIC PRODUCT. UNDER NO CIRCUMSTANCES SHALL DATALOGIC OR ITS LICENSORS BE LIABLE TO END USER OR ANY THIRD PARTY FOR LOST PROFITS, LOST DATA, INTERRUPTION OF BUSINESS OR SERVICE, OR FOR ANY OTHER SPECIAL, CONSEQUENTIAL, CONTINGENT, INDIRECT, INCIDENTAL, PUNITIVE, EXEMPLARY, OR OTHER SIMILAR DAMAGES, EVEN IF DATALOGIC OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. END USER MUST BRING ANY ACTION UNDER THIS AGREEMENT WITHIN 12 (TWELVE) MONTHS AFTER THE CAUSE OF ACTION ARISES.

6. Indemnification

To the maximum extent permitted by law, End User agrees to defend, indemnify and hold harmless Datalogic, its affiliates and their respective directors, officers, employees and agents from and against any and all claims, actions, suits or proceedings, as well as any and all losses, liabilities, damages, costs and expenses (including reasonable attorney's fees) arising out of or accruing from or related to Licensee's failure to comply with the terms of this Agreement, including but not limited to (1) non compliance with any applicable laws or regulations with Datalogic product documentation, (2) unauthorized use or disclosure of Software, and (3) use of Software in combination with software, hardware, systems, or other items not provided by Datalogic.

7. Support

End User may request support for Software from Datalogic at Datalogic's standard support fees and under Datalogic's standard support terms and conditions in effect at the time the support is requested.

8. Government Restricted Rights; International Use

8.1 Use, duplication, or disclosure of the Software by the U.S. Government is subject to the restrictions for computer software developed at private expense as set forth in the U.S. Federal Acquisition Regulations at FAR 52.227-14(g), or 52.227-19 or in the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013(c)(1)(ii), whichever is applicable.

8.2 If End User is using the Software outside of the United States, End User must comply with the applicable local laws of the country in which the Software is used and with U.S. export control laws. Without prejudice of the foregoing, the End User agrees to not export or re-export the Software, any part thereof, or any process or service that is the direct product of the Software to any country, person or entity subject to U.S. export restrictions. End User specifically agrees not to export or re-export any of the Software: (i) to any country to which the U.S. has embargoed or restricted the export of goods or services or to any national of any such country, wherever located, who intends to transmit or transport the Software back to such country; (ii) to any person or entity who you know or have reason to know will utilize the Software in the design, development or production of nuclear, chemical or biological weapons; or (iii) to any person or entity who has been prohibited from participating in U.S. export transactions by any federal agency of the U.S. government. End User warrants and represents that neither the U.S. Commerce Department, Bureau of Export Administration nor any other U.S. federal agency has suspended, revoked or denied its export privileges.

9. Third Party Software

The Software may contain one or more items of third party software which use is governed by separate third party license, unless otherwise stated.

10. Open Source Software

Portions of the Software include or operate with Open Source software ("Open Source Software").

Open Source Software is software covered by a publicly available license governed solely under Copyright law, whereas the complete terms and obligations of such license attach to a licensee

solely through the act of copying, using and/or distributing the licensed software, such obligations often include one or more of attribution obligations, distribution obligations, copyleft obligations, and intellectual property encumbrances. The use of any Open Source Software is subject to the terms and conditions of this Agreement as well as the terms and conditions of the corresponding license of each Open Source Software package. If there is a conflict between the terms and conditions of this Agreement and the terms and conditions of the Open Source Software license, the applicable Open Source Software license will take precedence. Datalogic is required to reproduce the software licenses, acknowledgments and copyright notices as provided by the authors and owners, thus, all such information is provided in its native language form, without modification or translation. Please reference and review the above mentioned information to identify which Open Source Software packages have source code provided or available.

11. Notices

All notices required or authorized under this Agreement shall be given in writing, and shall be effective when received, with evidence of receipt. Notices to Datalogic shall be sent to the attention of Datalogic USA Inc., Legal & IP Department, 959 Terry Street, Eugene, OR 97402, U.S.A., or such other address as may be specified by Datalogic in writing.

12. General Provisions.

12.1. Entire Agreement; Amendment. This document contains the entire agreement between the parties relating to use of the Datalogic products and the licensing of the Software and supersedes all prior or contemporaneous agreements, written or oral, between the parties concerning the use of the Datalogic products and licensing of the Software. This Agreement may not be changed, amended, or modified except by written document signed by Datalogic.

12.2. Waiver. A party's failure to enforce any of the terms and conditions of this Agreement shall not prevent the party's later enforcement of such terms and conditions.

12.3. Governing Law; Venue: End User agrees to the application of the laws of the country in which End User obtained the license to govern, interpret, and enforce all of End User's and Datalogic's respective rights, duties, and obligations arising from, or relating in any manner to, the subject matter of this Agreement, without regard to conflict of law principles. The United Nations Convention on Contracts for the International Sale of Goods does not apply.

All rights, duties, and obligations are subject to the courts of the country in which End User obtained the license. For licenses granted by Licensee who operates in the countries specified below, the following terms applies.

For Americas:

This Agreement is governed by the laws of the State of Oregon. This Agreement and the rights of the parties hereunder shall be governed by and construed in accordance with the laws of the State of Oregon U.S.A, without regard to the rules governing conflicts of law. The state or federal courts of the State of Oregon located in either Multnomah or Lane counties shall have exclusive jurisdiction over all matters regarding this Agreement, except that Datalogic shall have the right, at its absolute discretion, to initiate proceedings in the courts of any other state, country, or territory in which End User resides, or in which any of End User's assets are located. In the event an action is brought to enforce the terms and conditions of this Agreement, the prevailing party shall be entitled to reasonable attorneys' fees, both at trial and on appeal.

For Europe, Middle East and Africa:

This Agreement is governed by the laws of Italy. This Agreement and the rights of the parties hereunder shall be governed by and construed in accordance with the laws of Italy, without regard to the rules governing conflicts of law. Italian Court of Bologna shall have exclusive jurisdiction over all matters regarding this Agreement, except that Datalogic shall have the right, at its absolute discretion, to initiate proceedings in the courts of any other state, country, or territory in which End User resides, or in which any of End User's assets are located. In the event an

action is brought to enforce the terms and conditions of this Agreement, the prevailing party shall be entitled to reasonable attorneys' fees, both at trial and on appeal.

For Asia- Pacific Countries:

The validity, interpretation and construction of the Agreement shall be governed by and construed in accordance with Laws of the Republic of Singapore. Parties expressly disclaim the application of the United Nations Convention for International Sale of Goods.

Any dispute arising out of or in connection with this contract, including any question regarding its existence, validity or termination, shall be referred to and finally resolved by arbitration administered by the Singapore International Arbitration Centre ("SIAC") in accordance with the Arbitration Rules of the Singapore International Arbitration Centre ("SIAC Rules") for the time being in force, which rules are deemed to be incorporated by reference in this clause. The seat of the arbitration shall be Singapore.

The number of arbitrators will be three, with each side to the dispute being entitled to appoint one arbitrator. The two arbitrators appointed by the parties will appoint a third arbitrator who will act as chairman of the proceedings. Vacancies in the post of chairman will be filled by the president of the SIAC. Other vacancies will be filled by the respective nominating party. Proceedings will continue from the stage they were at when the vacancy occurred. If one of the parties refuses or otherwise fails to appoint an arbitrator within 30 days of the date the other party appoints its, the first appointed arbitrator will be the sole arbitrator, provided that the arbitrator was validly and properly appointed. All proceedings will be conducted, including all documents presented in such proceedings, in the English language. The English language version of these terms and conditions prevails over any other language version.

12.4. Attorneys' Fees. In the event an action is brought to enforce the terms and conditions of this Agreement, the prevailing party shall be entitled to reasonable attorneys' fees, both at trial and on appeal.

- END -

TABLE OF CONTENTS

PREFACE

ABOUT THIS MANUAL

This User's Manual is provided for users seeking advanced technical information, including connection, programming, maintenance and specifications. Other publications associated with this product can be downloaded free of charge from the website listed on the back cover of this manual.

Manual Conventions

The following conventions are used in this document:

The symbols listed below are used in this manual to notify the reader of key issues or procedures that must be observed when using the reader:

NOTE

Notes contain information necessary for properly diagnosing, repairing and operating the reader.

CAUTION

The CAUTION symbol advises you of actions that could damage equipment or property.

WARNING

The WARNING symbol advises you of actions that could result in harm or injury to the person performing the task.

TECHNICAL SUPPORT

Support Through the Website

Datalogic provides several services as well as technical support through its website. Log on to (www.datalogic.com).

For quick access, from the home page click on the search icon, and type in the name of the product you're looking for. This allows you access to download Data Sheets, Manuals, Software & Utilities, and Drawings.

Hover over the Support & Service menu for access to Services and Technical Support.

Reseller Technical Support

An excellent source for technical assistance and information is an authorized Datalogic reseller. A reseller is acquainted with specific types of businesses, application software, and computer systems and can provide individualized assistance.

CHAPTER 1
INTRODUCTION

DL.CODE software is a User Interface client application that provides reading device configuration for Stand Alone, and Master/Slave configurations. It is installed in and runs on Windows-based PCs (usually laptops), and connection takes place through an Ethernet TCP/IP interface.

It also provides visual monitoring of images that can be stored in an Image Database either locally on the device or to the local or a remote PC.

DL.CODE provides PackTrack Calibration for omnidirectional reading and tracking stations used in Logistics applications.

DL.CODE offers statistic and diagnostic information at reading station level whether the station is made up of a single reader or several readers connected in a Master Slave configuration.

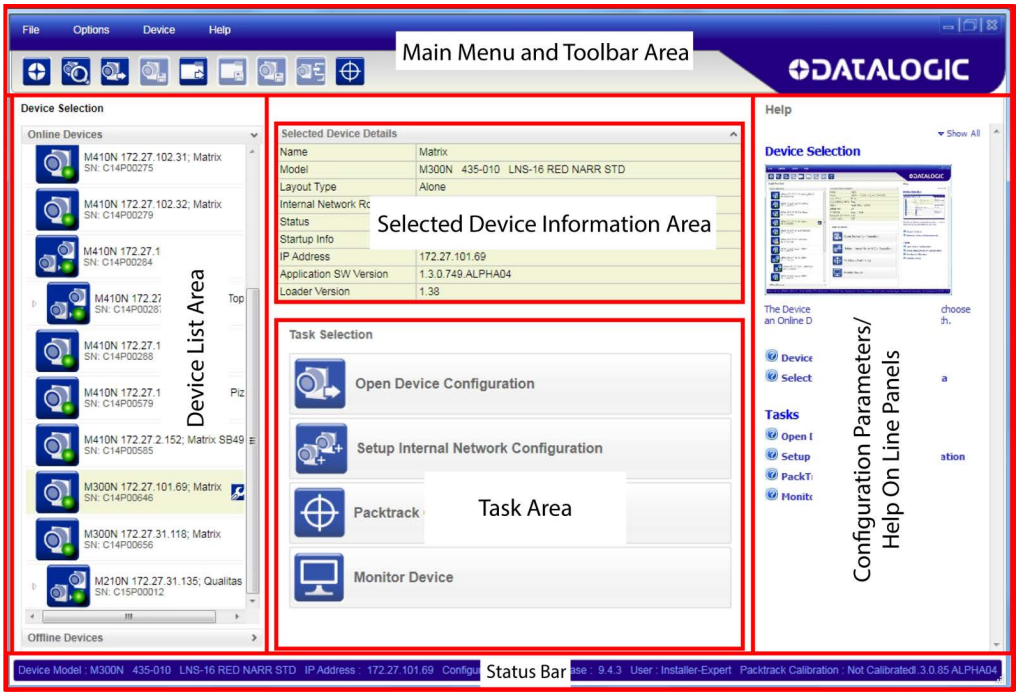


Figure 1 - Main Window Areas

MAIN FEATURES

A summary of the DL.CODE main features is listed below:

- Simultaneous Device Monitoring from different remote PCs
- 3 different User access levels
- User and Session Language configuration in real time
- System configuration
- Dynamic content and automatic page update

CONFIGURATION AND MONITORING SESSIONS

Device configuration can be performed using DL.CODE running on a remote PC through a single session. Multiple instances of DL.CODE cannot be run on a PC and once a device is connected for configuration it cannot be accessed by another PC running DL.CODE.

However the Monitoring feature can be accessed simultaneously by several PCs running DL.CODE.

SIMULATORS

DL.CODE has different device prototypes which can be loaded as Offline devices. This allows offline configurations to be prepared and loaded to a device at a later time.

NOTE

Creating and saving offline configuration (.dlcfg) files and then loading them onto physical devices can be done but the following precautions should be followed:

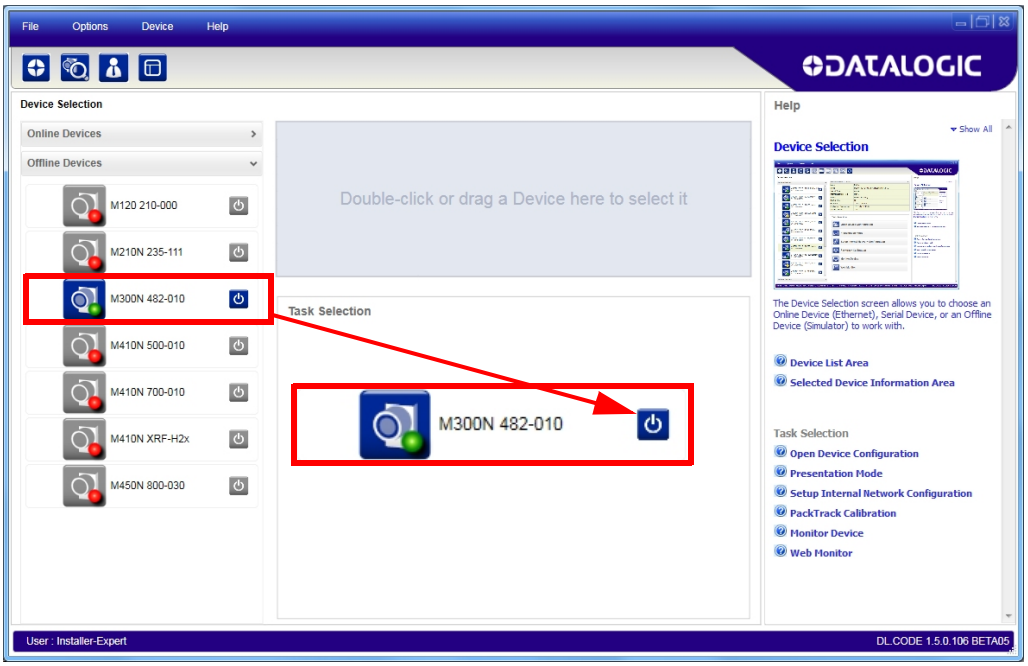
- Due to differences in image sensors, internal memory, etc., it is strongly suggested to use the same family product as the device to be configured.

- The following parameter groups may need to be adjusted: Advanced Setup Image Settings, Reading Phase, Output Setup.

- Advanced Setup Code Settings, Good Read, and Data Formatting will be maintained.

To load a Simulator, Click on the *Offline Devices* tab at the bottom of the Device List Area to open the list of available simulators.

All of the simulators are offline by default. To select a reader click its Simulator Power button icon.



Now you can double-click or drag the simulator into the Selected Device Information Area and begin a new configuration. See Chapter 3, .

DEVICE CONFIGURATION

DL.CODE is designed to simplify standard configuration by grouping the basic functions into three major parameter groups: Decoding, Operating Mode, and Output.

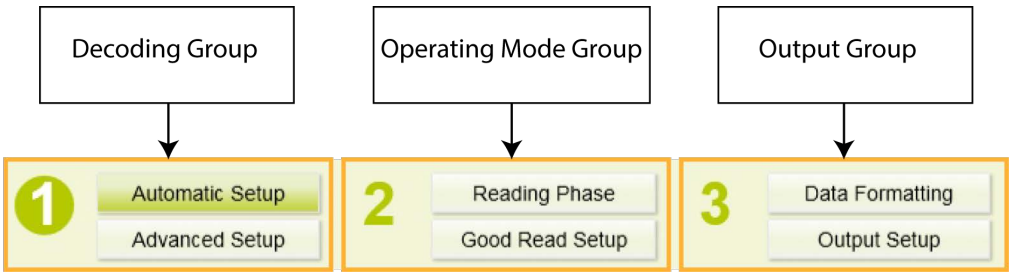


Figure 2 - DL.CODE Configuration Groups

Each major group is sub-divided into two parts as follows:

1. Decoding:
   - Automatic Setup: this is a new feature that provides a one-step process for automatically setting the photometry for image acquisition (Image Setting) and the code symbology selection. This is particularly useful for DPM applications and requires no calibration. See "Automatic Setup" starting on page 15.
   - Advanced Setup: this group completely manages:

- Image Setting parameters, including **Image Auto Setup** for photometry, multiple Image Settings, **Focus Autolearn** for liquid lens models, a **Focus Calibration** tool with oscilloscope for manual focusing and a **PPI** acquisition tool for calibration.
- General Settings including Processing Time and Code Grading parameter management.
- Code Symbology selection and configuration including Code Filtering parameters and a Code Localization tool. It also contains the **Code Autolearn** routine to find one or more unknown code symbologies on an acquired image.
- The image Cropping Region Area tool is available to this group to help speed up decoding.

2. Operating Mode:
   - Reading Phase: this group manages the operating mode for image acquisition.
   - Good Read Setup: this group manages data collection: Code Collection, Code Combination, Code Presentation, or Match Code.

3. Output
   - Data Formatting: this group manages the output message to the Host.
   - Output Setup: this group manages the digital outputs as well as the Green/Red Spots.

CHAPTER 2
INSTALLATION

DL.CODE DISTRIBUTION CONTENTS

The DL.CODE program distribution contains the following:

- Complete Installation of DL.CODE
- .NET Framework (if not already present)
- This manual

HARDWARE REQUIREMENTS

Typical hardware requirements for a DL.CODE Client PC are:

- 2.00 GHz or faster microprocessor
- 1 GB RAM
- 2 GB hard disk for 64-bit machines; 1 GB hard disk for 32-bit machines
- 100 Base-T Ethernet
- One 19” or larger monitor (optimized for 1280x1024 resolution)

SOFTWARE REQUIREMENTS

- One of the following Windows Operating Systems (32 or 64-bit):

  - Windows 7

  - Windows 8

  - Windows 10

NOTE

The Windows XP operating system is no longer supported by DL.CODE.

- Web Browser: Google Chrome, Mozilla Firefox, Microsoft Internet Explorer, Opera, etc.

NOTE

The Google Chrome Web Browser is recommended for it superior performance characteristics.

CAUTION

DL.CODEdoes not currently support Windows Embedded (often used in industrial PCs and/or PLCs).

INSTALLING DL.CODE

NOTE

Starting from version 1.5.0, multiple versions of DL.CODE can be installed on the same PC. Each version will be installed in its own folder, however only one version can be run at a time.

Standard Installation

1. On the PC that will be used for configuration (running Windows 7, 8 or 10), download the DL.CODE mini-DVD .zip file. Extract the files maintaining the folder structure and run the **start.hta** file to access the installation pop-up. Click on the **Install DL.CODE** link to run the installation program and follow the installation procedure.

NOTE

If you need to configure Serial models, check the RS232 Serial Port Driver installation box in the Welcome window of the DL.CODE Installer.

NOTE

If you need to configure USB models, check the Matrix USB Driver installation box in the Welcome window of the DL.CODE Installer and follow the procedure given in the specific reader Reference Manual.

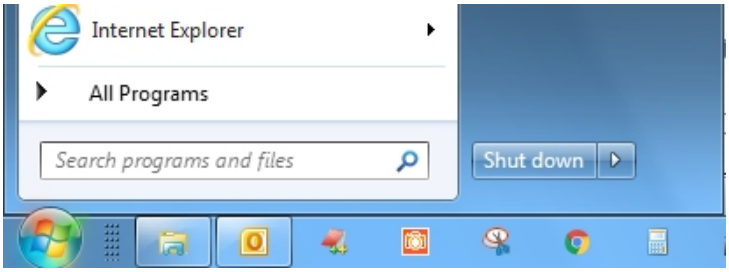
2. When the installation is complete, the DL.CODE entry is created in the Start>Programs bar under “Datalogic” as well as a desktop icon. Double-click the desktop icon to run it.

Silent Installation

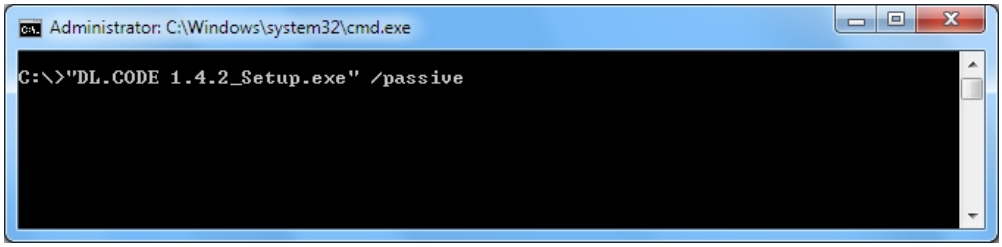
A silent installation which requires no input from the user can be run from a command line prompt. You must have Administrator rights on the PC for this installation. There are two options:

- Quiet: this does not display any user interface.
- Passive: this displays the user interface but proceeds without requiring any user input.

1. Open a command line prompt from the Windows Start Menu by typing “cmd” in the search box:



2. Set the directory to where the extracted DL.CODE setup package is located.
3. Invoke the package with double quotes “ ” and either **/passive** or **/quiet** switches.



UNINSTALLING DL.CODE

To completely uninstall DL.CODE including software drivers from your PC, you must use the following sequence:

1. Uninstall all versions of DL.CODEfrom your PC from the Control Panel.
2. Uninstall the com0com driver from the com0com folder in Windows Start Menu – All Programs.
3. Uninstall the USBCOMInstaller from the Control Panel.
4. Uninstall the cdcecmInstaller from the Control Panel.

NOTE

Trying to uninstall software drivers before all versions of DL.CODE are uninstalled will cause an error.

CHAPTER 3
QUICK START

To help you get started, here is an example configuration demonstrating the basic steps of DL.CODE configuration.

To configure your device for your application using DL.CODE, the following preliminary steps are assumed:

- The reading device(s) are installed and running.
- DL.CODE is installed and running (Chapter 2,).

ETHERNET DEVICE DISCOVERY

NOTE

To discover Serial models see "Serial Device Discovery" on page 11**. To discover USB models, see the specific reader Reference Manual for details.**

The User Interface opens and displays a list of all the devices belonging to the Local Area Network. DL.CODE has a discovery feature to accomplish this task.

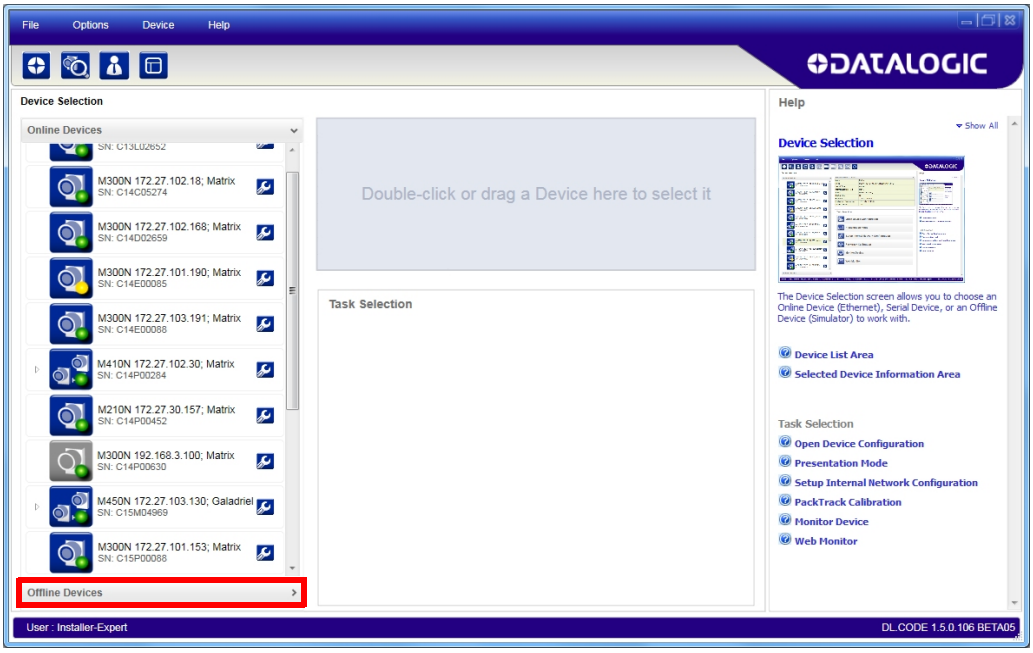


Figure 3 - Device Discovery

The discovery feature will also show devices not belonging to the LAN and display them in gray (see Figure 3).

The following procedure will demonstrate an example configuration.

1. First, the device must be added to the LAN by aligning its IP Address to the network. The network administrator should provide valid LAN address(es).

2. Click on the device wrench icon to open the Device Environment Configuration window.

3. Change the Ethernet Settings (IP Address, Subnet Mask, Gateway Address etc.) according to the network requirements. See also Figure 4 below.

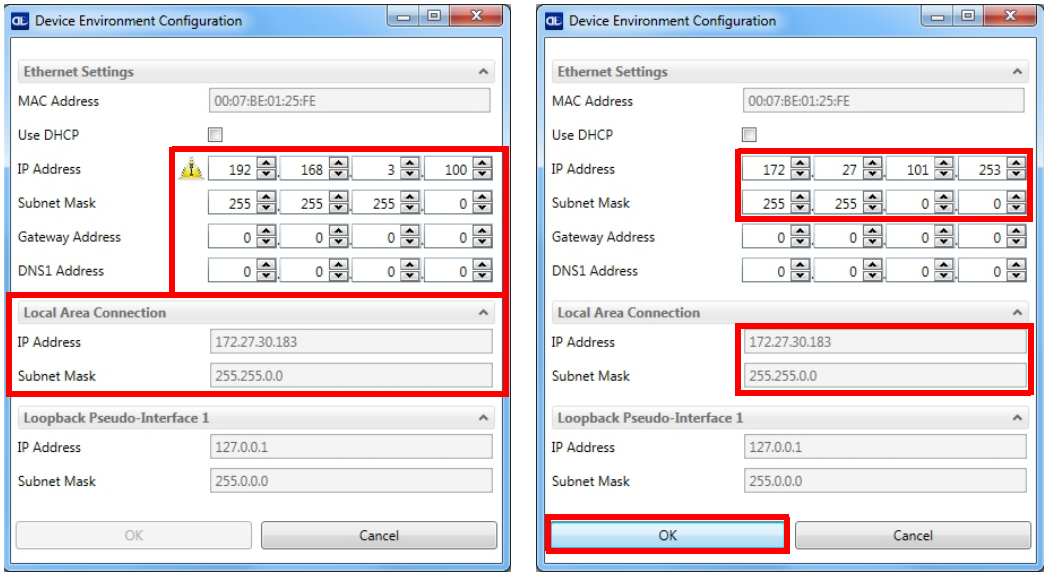
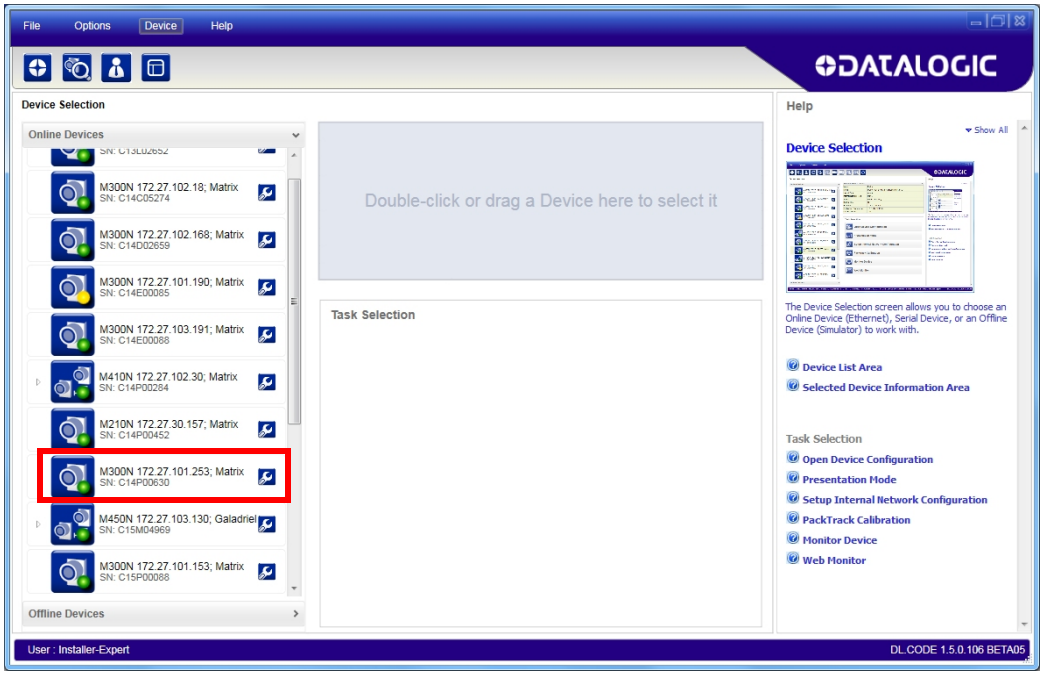
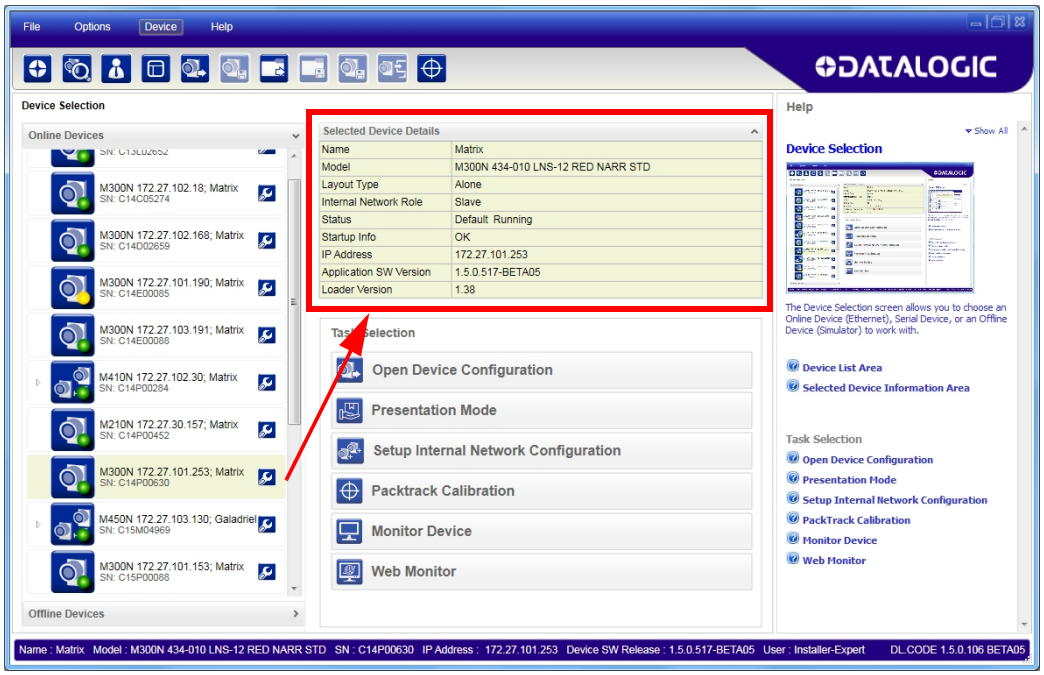


Figure 4 - Device Environment Configuration Window

4. Click OK; the device will reappear in the list of Online Devices (in color) meaning it is now part of the LAN and can be configured. The new IP address will also be displayed.



5. Double-click on or drag the device icon into the Selected Device Information Area. Details about the device will be displayed in this area.



NOTE

After device discovery, configure your device through DL.CODE as described in Decoding Configuration Parameters, Operating Mode Configuration Parameters, and Output Configuration Parameters.

SERIAL DEVICE DISCOVERY

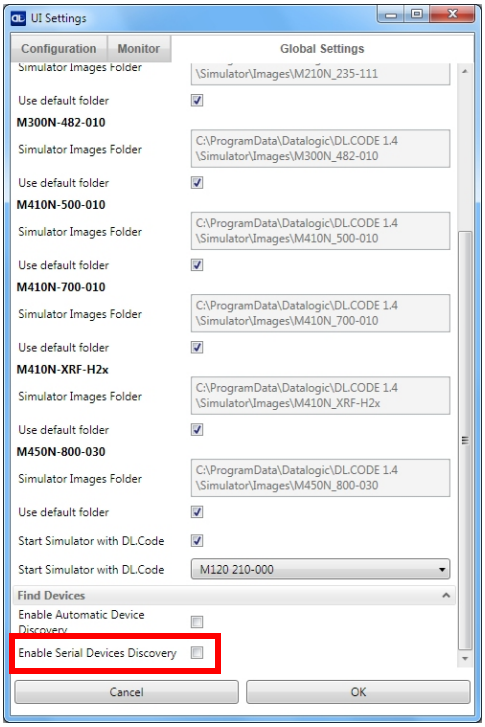
Starting from DL.CODE 1.4.0, serial port communication is supported for device discovery and configuration. This allows dedicated serial communication models to be configured through DL.CODE.

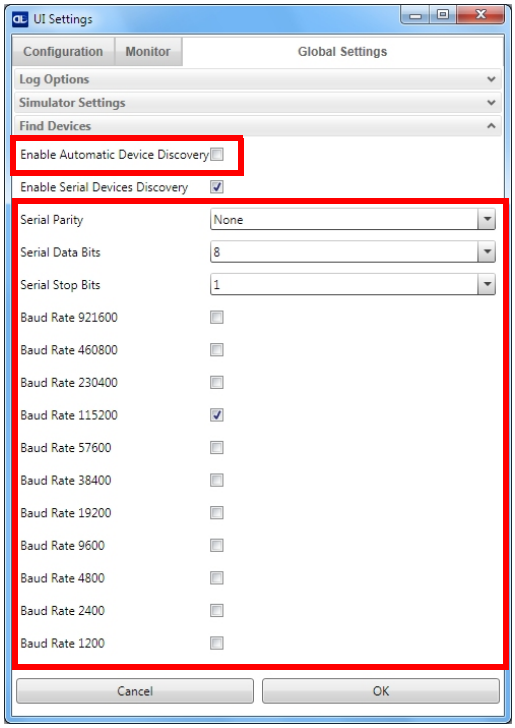
NOTE

Although this feature allows all devices to be configured through their Serial Interface, be aware that transmission speeds and some DL.CODE features are limited when using this interface. It is always advised to use the Ethernet interface whenever possible.

This feature is not enabled by default, so the first thing to do is to enable it through the UI Settings window.

1. From the main menu open the Options > UI Settings Window.
2. Click on the Global Settings menu and scroll down to the Find Devices tab.
3. Check the Enable Serial Device Discovery box. Scroll down to see the following selections.
4. Select the Serial communication parameters according to your application.





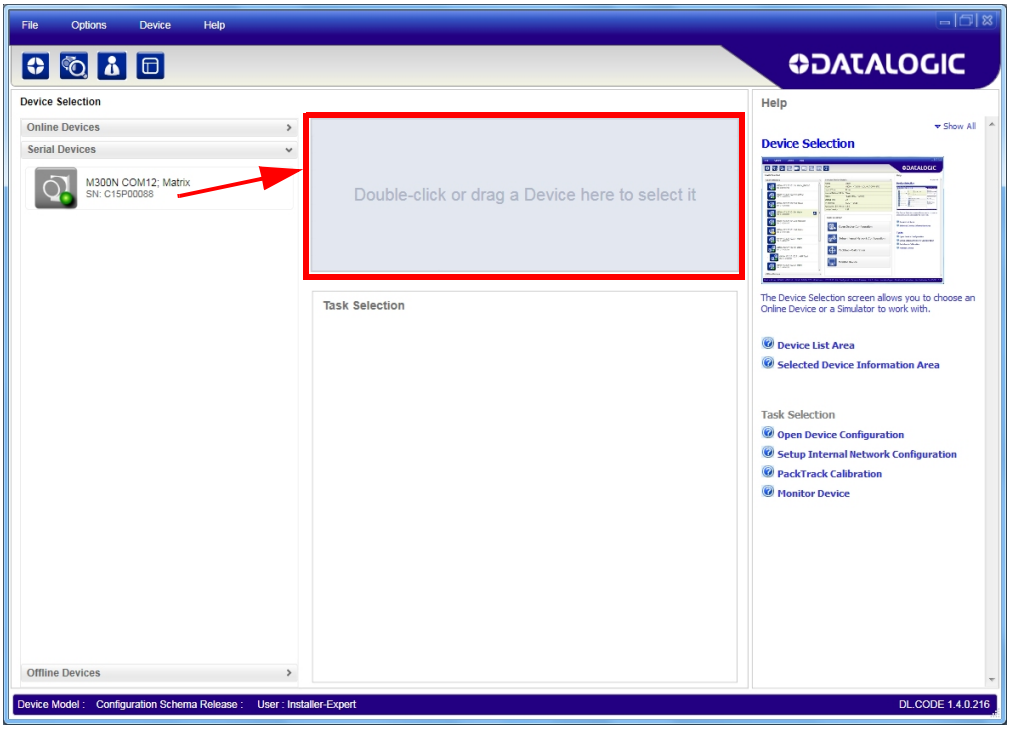
NOTE

If you're not sure of the Serial baud rate you can also check the Enable Automatic Device Discovery box which for serial devices will try communication at all baud rates, but only at No parity, 8 data bits;1 stop bit.

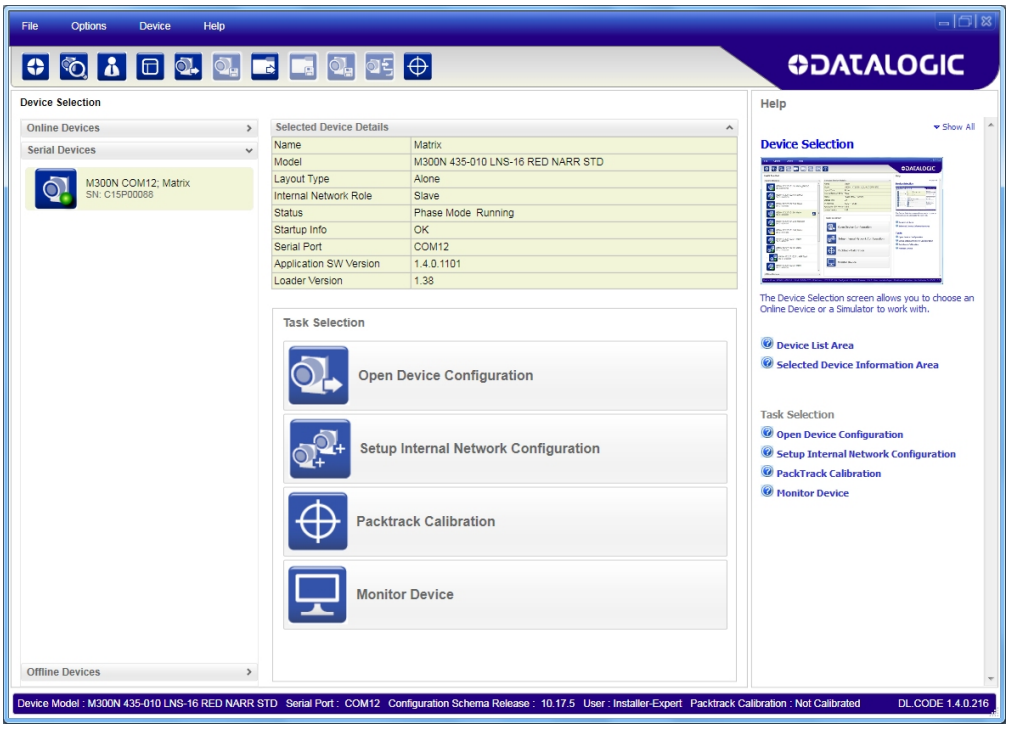
Enabling this parameter can notably lengthen Discovery time, therefore in general it is better to disable it to increase Discovery efficiency.

5. Click OK to return to DL.CODE and click on the Getting Started icon.

6. Open the Serial devices tab and double-click on or drag the device icon into the Selected Device Information Area.



The device is now connected to the DL.CODE Configuration environment. Configure your device through DL.CODE as described in Decoding Configuration Parameters, Operating Mode Configuration Parameters, and Output Configuration Parameters.



DECODING CONFIGURATION PARAMETERS

The Decoding Configuration parameters are divided into two groups: **Automatic Setup** and **Advanced Setup**.

- **Automatic Setup** provides an automatic procedure for setting optical/illumination and code definition parameters to obtain the most stable decoding conditions for one or more code symbologies based on the images presented to the reader. It can be set to include Image Filters if necessary. See the table below for codes and filters managed by Automatic Setup. Automatic Setup is especially useful for DPM applications.

Enabled 1D Codes	Enabled 2D Codes	Enabled Postal Codes
CODE 128 CODE 25 CODE 93 CODE 39 CODABAR INTERLEAVED 2 OF 5 UPC/EAN MSI PDF GS1 DATABAR	DATAMATRIX ECC 200 QR AZTEC MAXICODE DOTCODE	POSTNET PLANET

- Advanced Setup provides access to the complete array of optical/illumination and code definition parameters that can be fine-tuned semi-automatically and manually to obtain the best results for applications of any complexity.

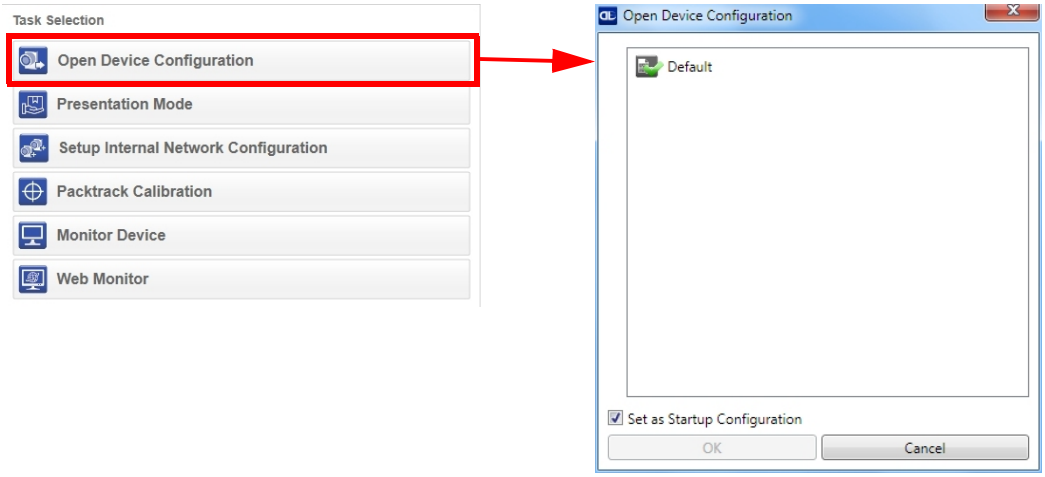
NOTE

If your application requires multiple code symbologies, multiple Image settings, Code Grading or other parameter settings for decoding, then use the Advanced Setup (see Advanced Setup, starting on page 17**).**

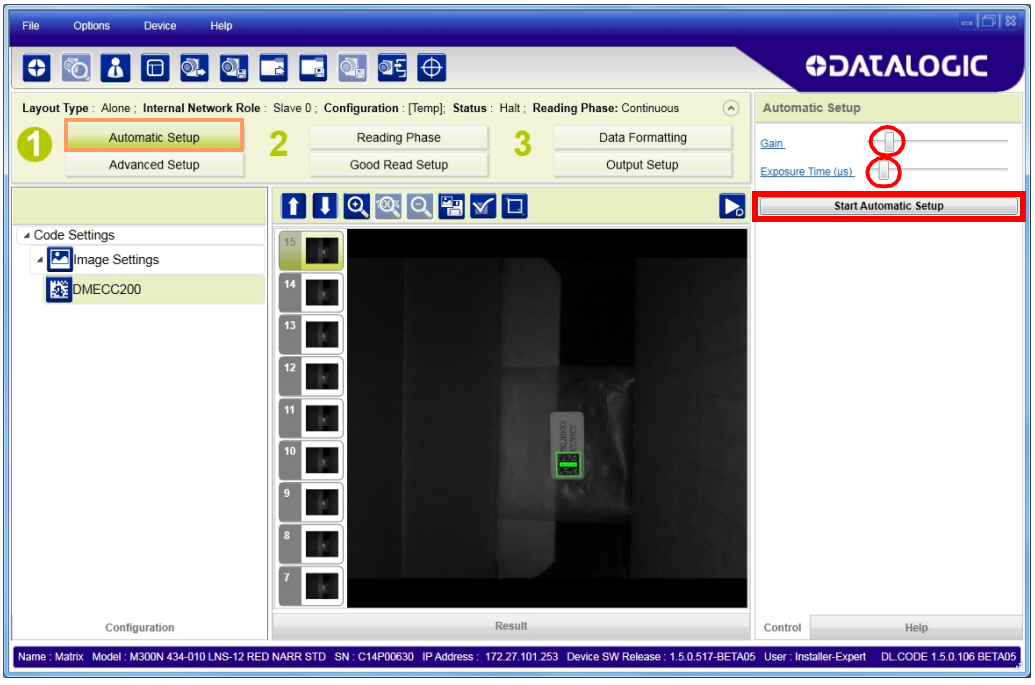
Automatic Setup

To begin configuration, the reader must be correctly mounted so that its Field of View covers the application reading area.

1. From the Task Area select Open Device Configuration.
2. The Open Device Configuration window opens showing the list of currently saved configurations (jobs) saved on the device. For new devices, the only saved job is the Default configuration. Click OK. The device enters run mode and begins acquiring images.



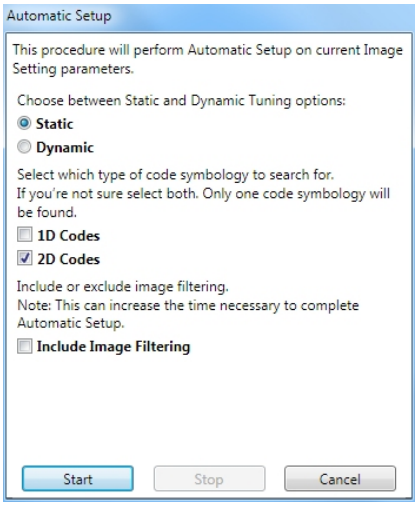
3. Place the application code in front of the reader at the correct application reading distance.
4. Click on the Pause button  to stop image acquisition.



NOTE

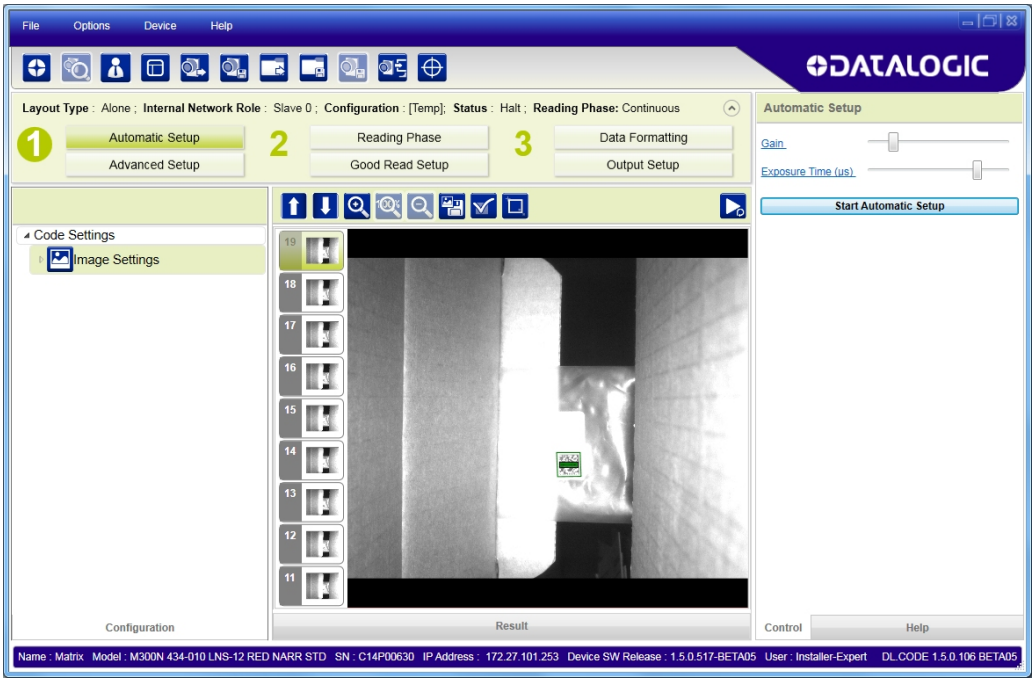
If the image display area is too dark to see the images being captured, you can drag the Gain and Exposure time sliders (circled in red in the figure above) to the right to increase visibility. This will not affect Automatic Setup.

5. Click on the **Start Automatic Setup** button. The following window is displayed.



6. Select the correct reading conditions: Static or Dynamic Tuning, 1D or 2D code, Include Image Filtering (to find the best decoding condition).

7. Click Start to begin the procedure. The reader begins acquiring images. At the end of the procedure the **Status: Completed** message appears. You can Close the window.

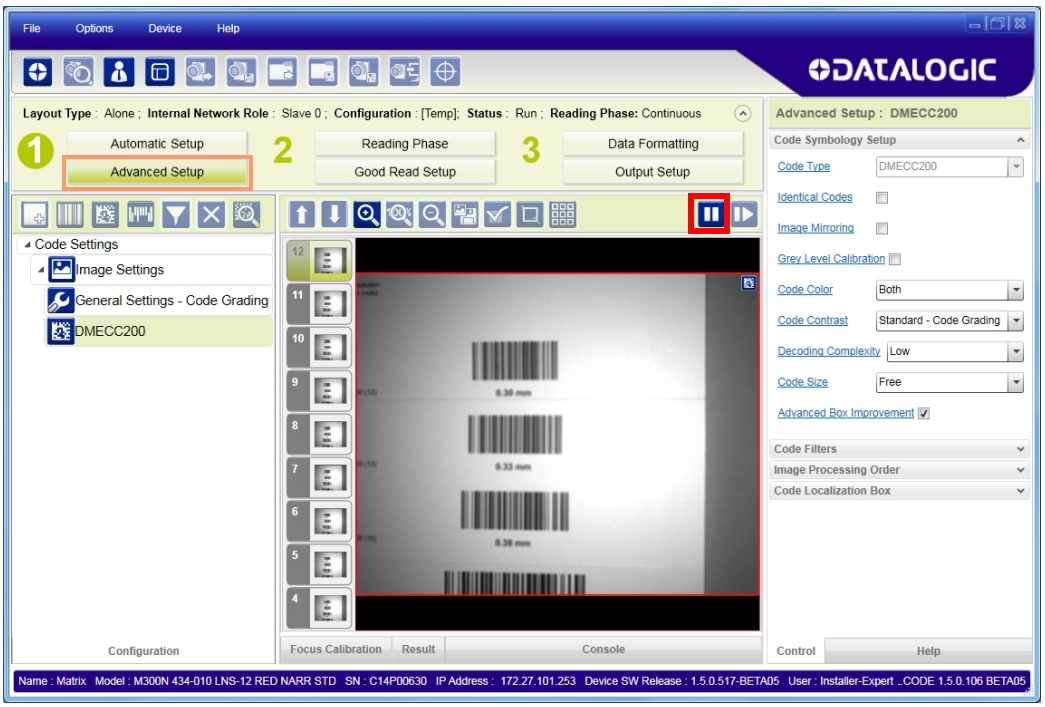


Your reader is now optimized for decoding. Continue with the Operating Mode configuration described in Operating Mode Configuration Parameters.

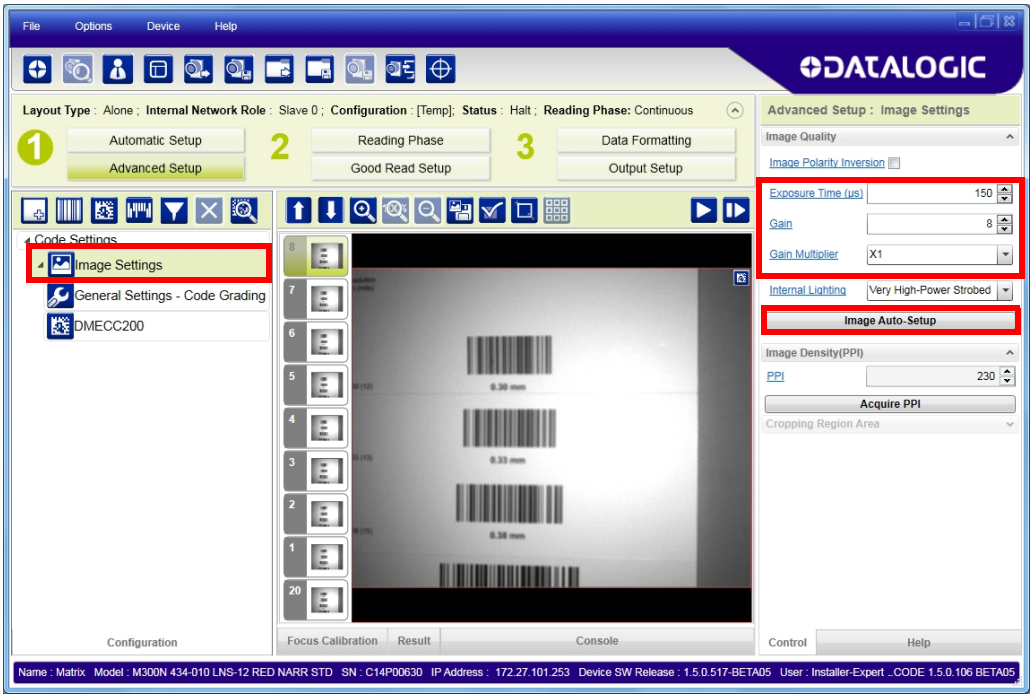
Advanced Setup

If your application requires multiple code symbologies, multiple image settings or other parameter settings for decoding, the use the Advance Setup.

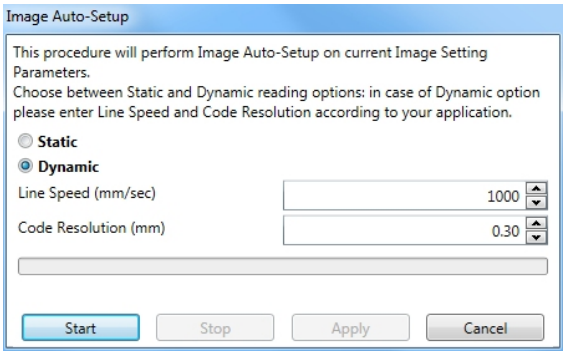
1. Click on the Advanced Setup button and press the Play icon.
2. Place the **Grade A Barcode Test Chart** in the reading area. Once positioned, stop image acquisition by clicking on the Pause button.

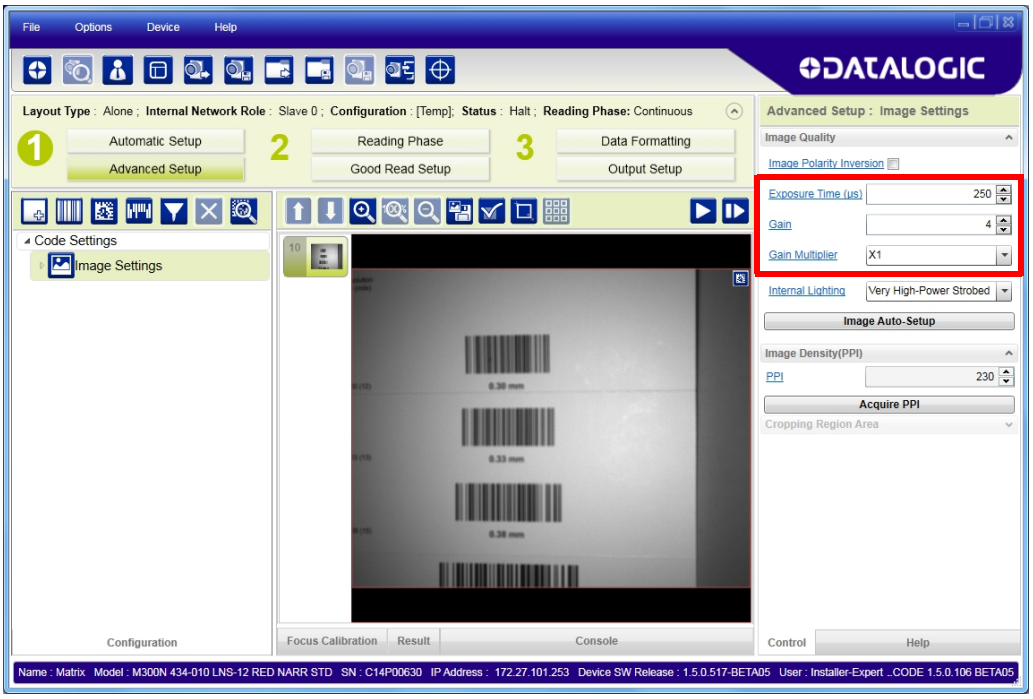


3. Click the Image Settings branch and then click the Image Auto-Setup button to automatically acquire the best exposure time and gain values.



4. Select the Static or Dynamic Self-Tuning option; Start the Image Auto Setup and Apply to the Image Settings.



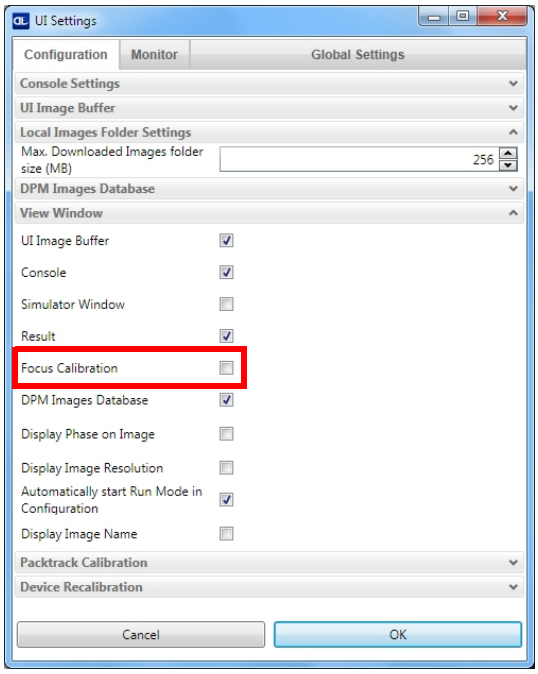


NOTE

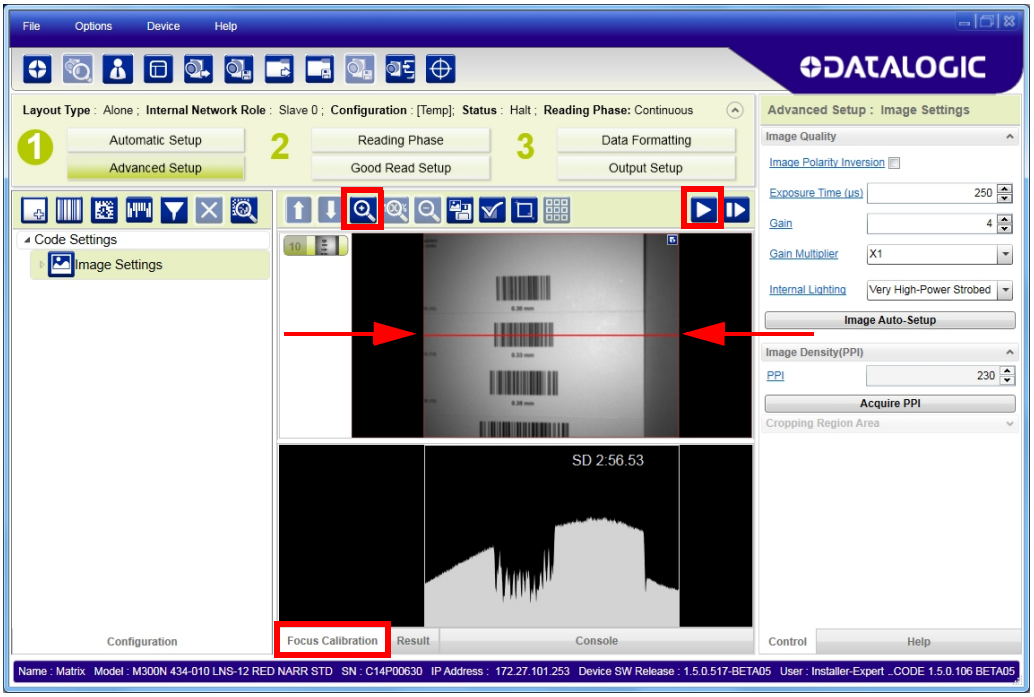
For applications having multiple lighting or code reading conditions, up to 10 different

Image Settings can be configured by adding them with the icon.

For the next step you need to enable the Focus Calibration Tool from the Options>UI Settings Configuration tab if not already enabled.



5. Now click on the Focus Calibration tab at the bottom of the window. The oscilloscope view is shown in the bottom panel and can be used for manual focus adjustment.



The red line in the image panel above the oscilloscope must pass through the code. You can click and drag the red line vertically to reposition it over the code.

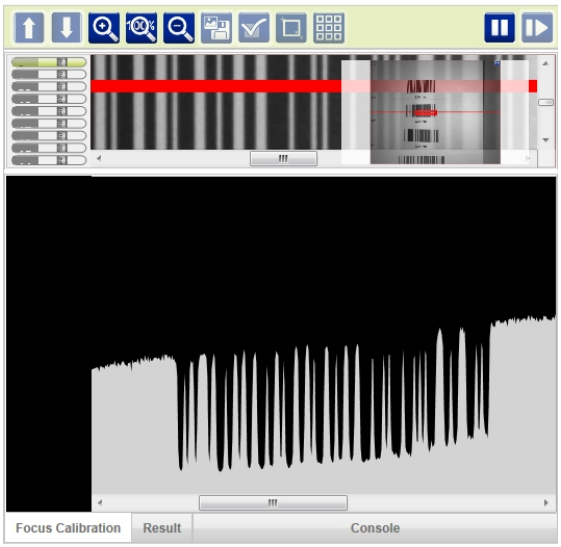
NOTE

You can enlarge the visual image of the code and the oscilloscope views, you can drag the Focus Calibration window up and click on the zoom image icon repositioning it on the code.

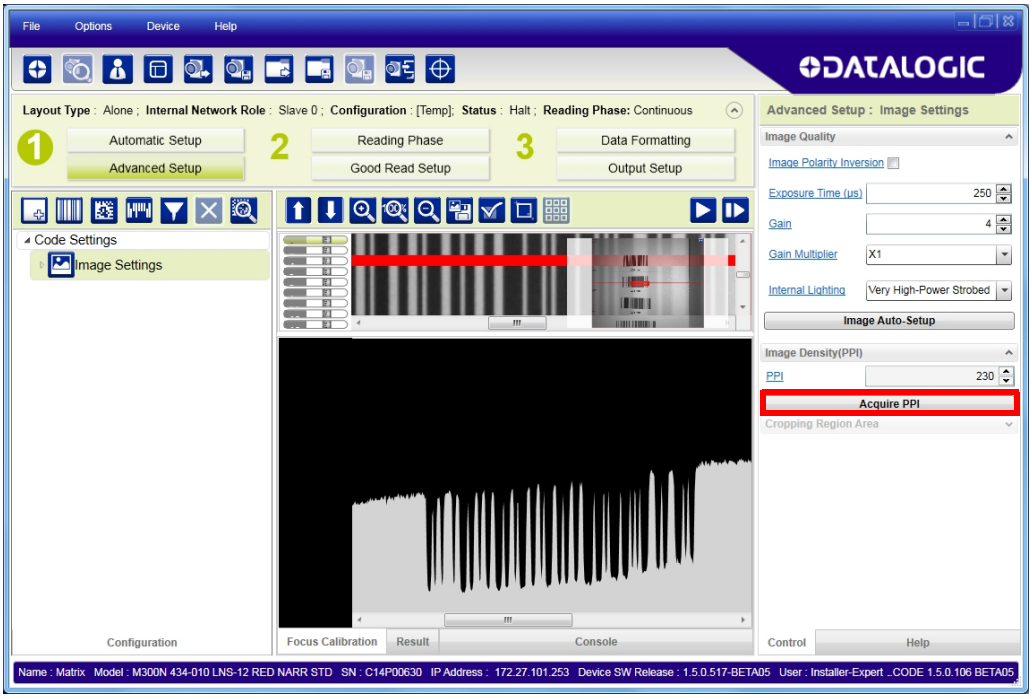
While in run mode, manually adjust the focus until the bars relative to the code in the oscilloscope demonstrate their maximum length (focus).

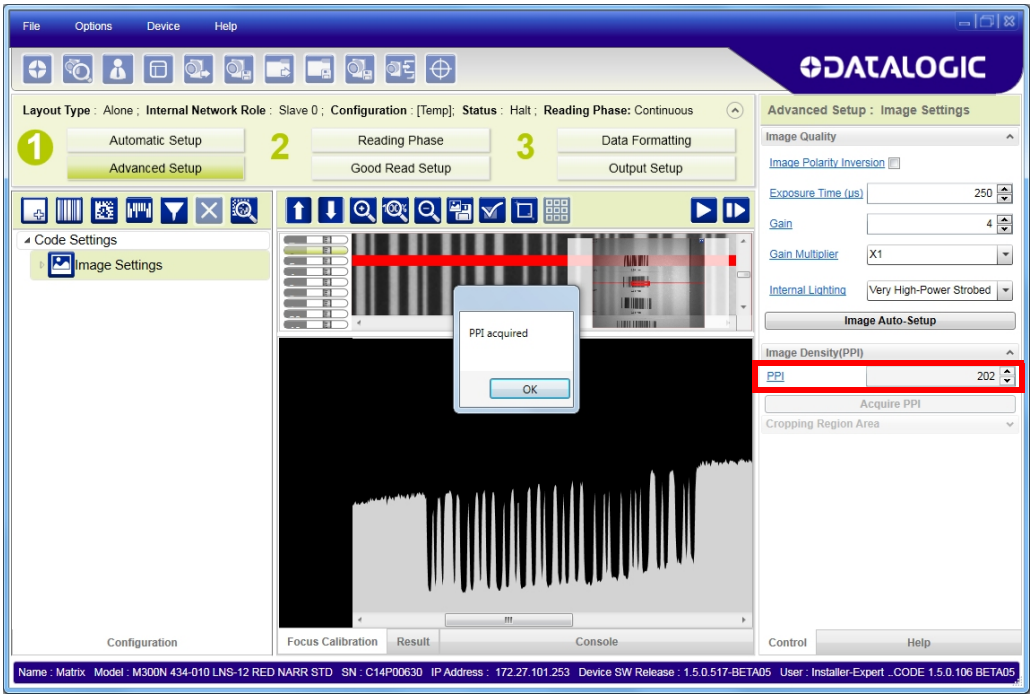
You can also see the visual focus on the code view.

When focused, click Pause to stop image acquisition.



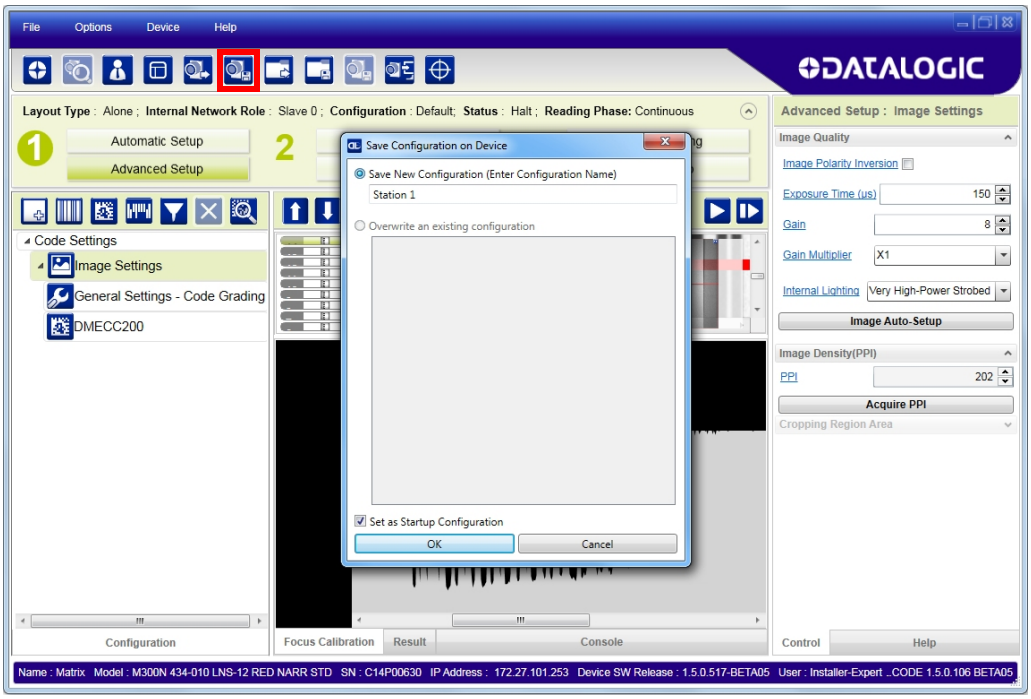
6. Click the **Acquire PPI** button to automatically set Image Density so that reader will function correctly and to the fullest extent of its capabilities. This procedure is necessary for first time installations, or if the focal distance is changed.





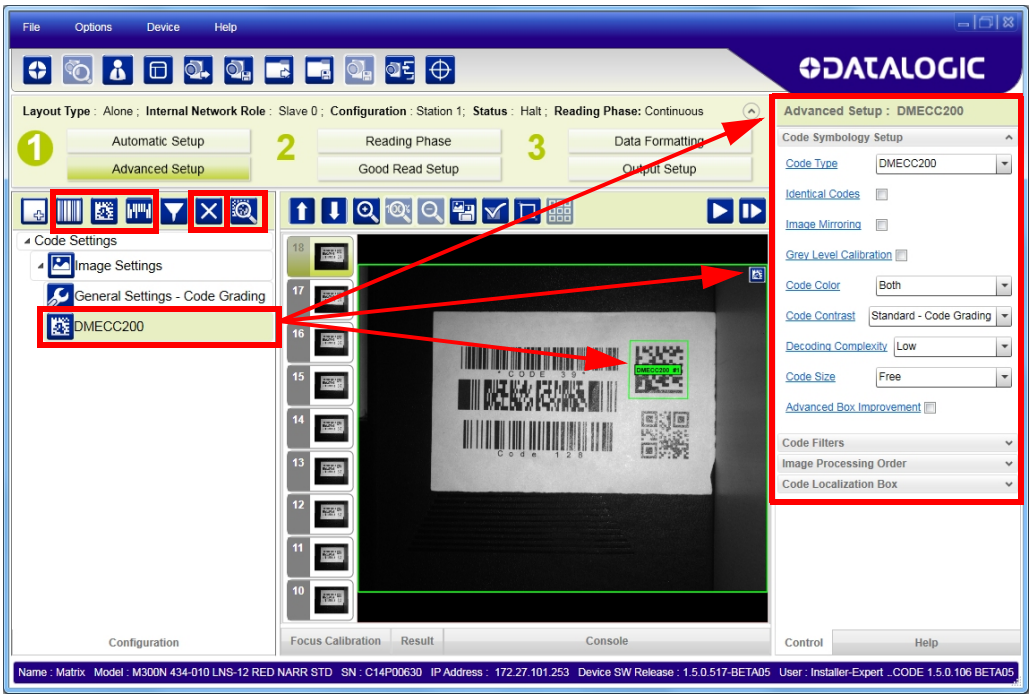
NOTE

At this point it is probably a good idea to save the configuration from temporary memory to permanent memory giving it a specific name.



7. Now place **an application specific code** in front of the reader and **only** click the **Image Auto-Setup** button to register any changes in lighting or code surface contrast. **Do not repeat Focus Calibration or PPI.**

8. Click on the Data Matrix ECC 200 symbology under the Image Settings branch (enabled by default). If this symbology is among those in your application it will be shown in the image display with its code symbology name and a green box around it indicating it is decoded.



NOTE

The large green box for each symbol indicates the code localization area which by default is equal to the maximum FoV. It can be resized and moved by dragging its borders with the mouse. The code must be found within this area in order to be decoded.

9. Add your application specific codes to the Code Settings by selecting them from the icons over the Configuration Parameters tree area. If the Data Matrix symbology is not used, then delete it from the Code Settings with the Delete icon.

If you don't know the code type, you can use the Code Autolearn feature by clicking on the Autolearn icon.

See "Code Autolearn Feature" on page 46 for details.

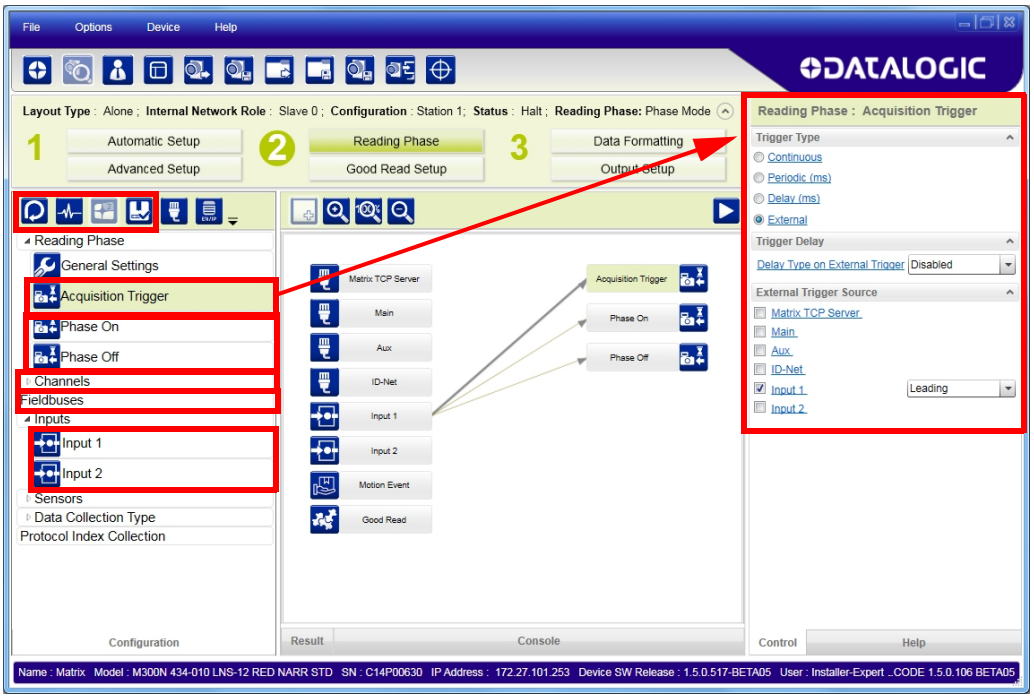
10. For each symbology set the relative parameters according to your application.

OPERATING MODE CONFIGURATION PARAMETERS

The Operating Mode Configuration parameters are divided into two groups: **Reading Phase** parameters and **Good Read Setup** parameters.

Reading Phase

1. Select your application specific Operating Mode from the icons over the Configuration Parameters tree area: Continuous, One Shot, Phase Mode or PackTrack.

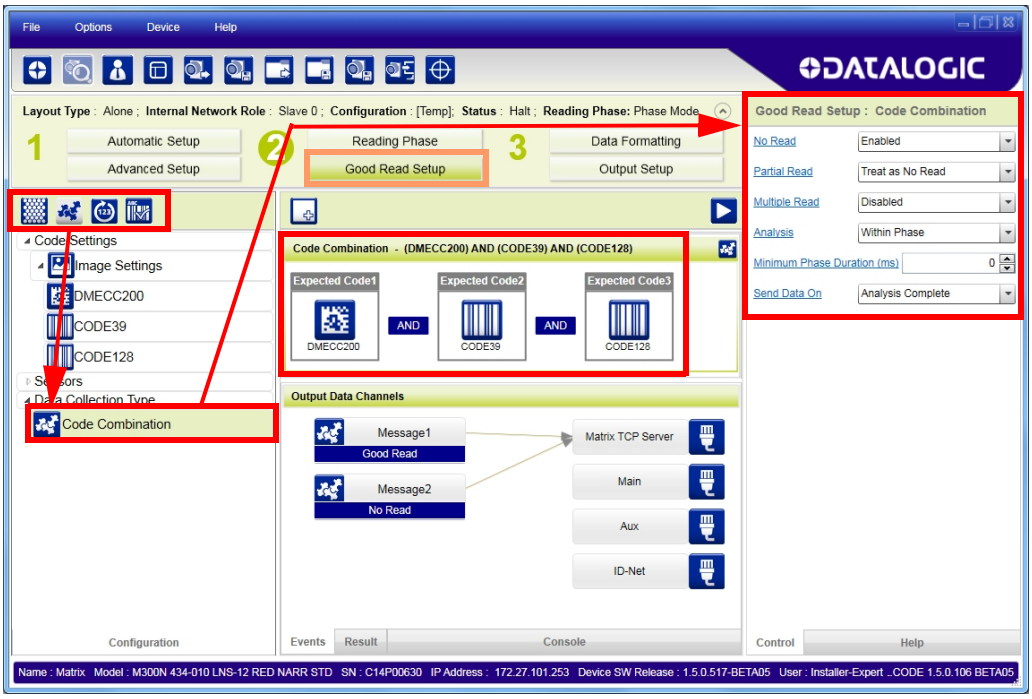


2. Configure the relative Operating Mode parameters from the Reading Phase parameters panel. Different groups will appear in the panel depending on the selected icons over the Configuration Parameters tree area.

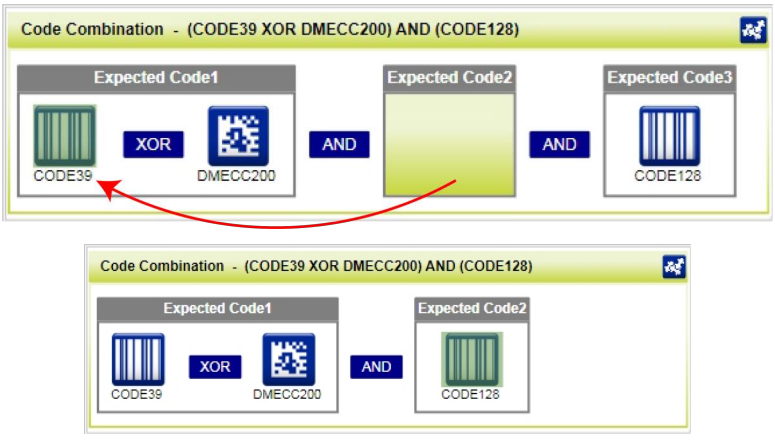
Good Read Setup

1. Select your specific data collection type from the icons over the Configuration Parameters tree area: Code Collection, Code Combination, Code Presentation or Match Code. Not all data collection types are available for all Operating Modes; for example PackTrack Operating Mode only supports Code Combination. Incompatible data collection types will be shown in gray and cannot be selected.

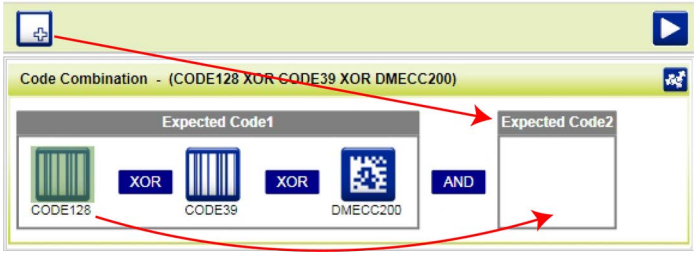
The following example shows Code Combination. By default, the Expected Codes (when more than one code type is selected), are in logical AND, which means that all codes are required to be decoded to produce a Good Read condition.



2. If a Good Read condition should be produced when any single code is decoded, independent from the others, then they need to be combined in logical XOR. To do this, drag the code icon(s) from their relative Expected Code box into the Expected Code box of the XOR combination you wish to create. Then delete the empty box by selecting it with the mouse (highlighted) and pressing the delete key on your keyboard.



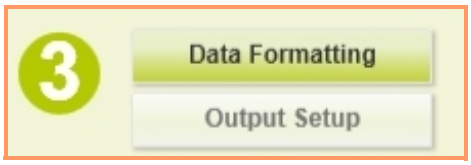
To create a logical AND condition from a logical XOR, create a new Expected Code box using the Add icon. Then drag the desired code from one box to the other.



OUTPUT CONFIGURATION PARAMETERS

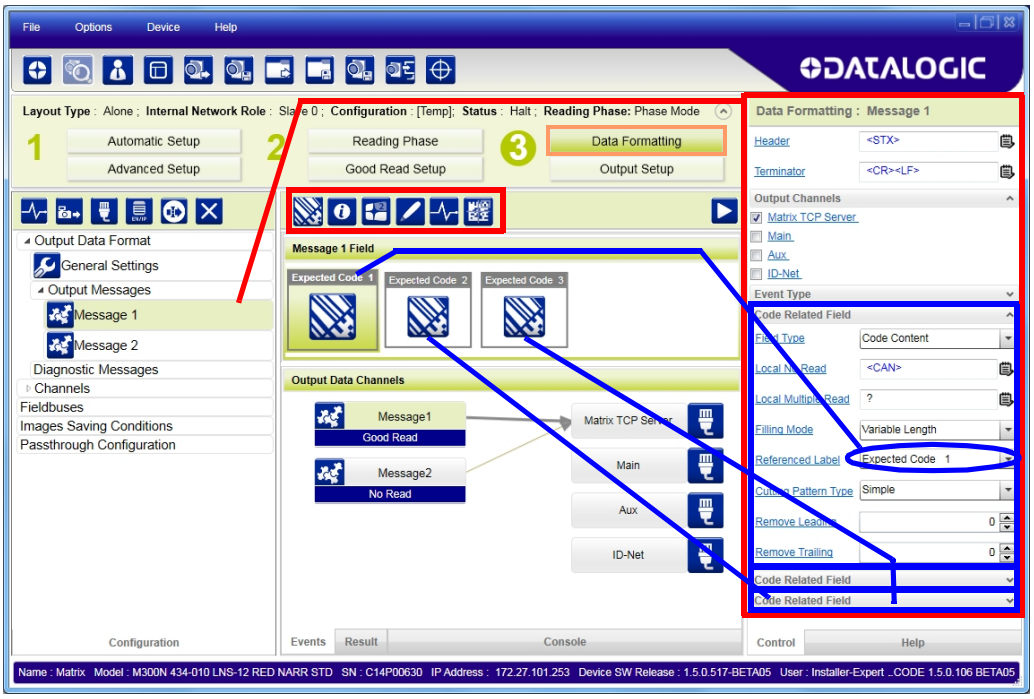
The Output Configuration parameters are divided into two groups: **Data Formatting** parameters and **Output Setup** parameters.

Output Group



Data Formatting

1. Configure your application specific Data Formatting Message(s) from the Configuration Parameters tree area: Message 1, Message 2, etc.

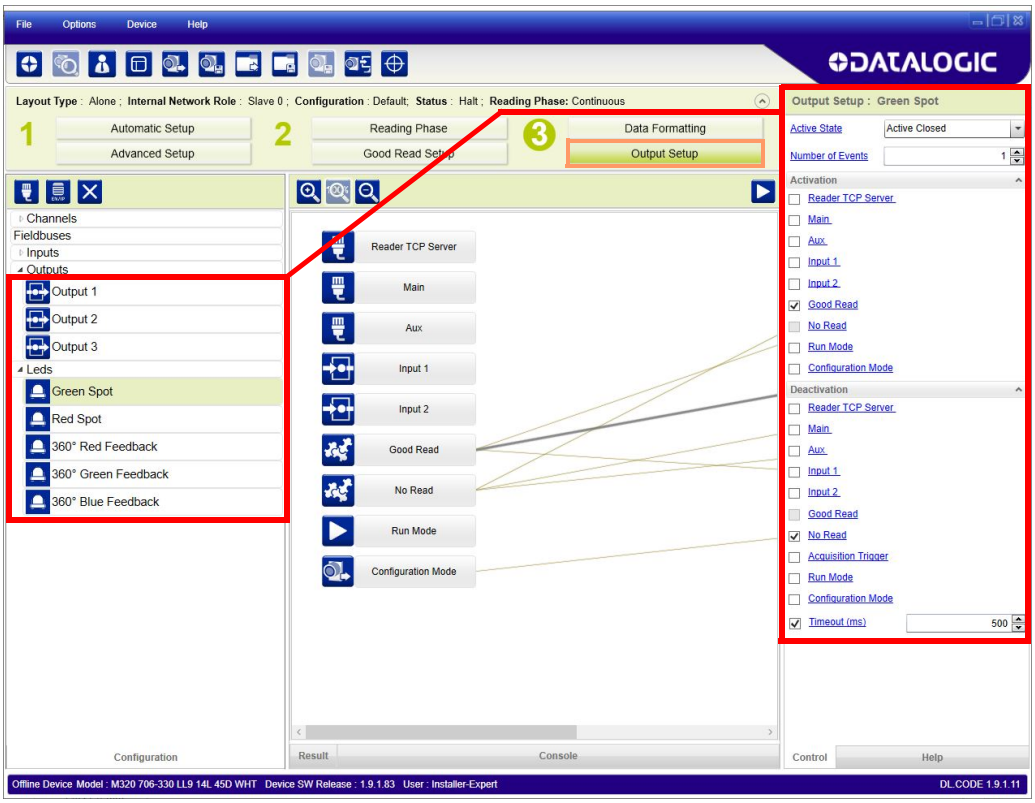


You can add fields to the output message by clicking on the icons above the Message Field area. They will be appended to the message. You can drag them to position them between other fields in the message so that the output message is ordered according to your application requirements.

Each field has its own relative configuration parameters in the parameters panel.

Output Setup

1. Configure your application specific Digital Output(s), Green/Red Spots and 360° Feedback (depending on Matrix model) from the Configuration Parameters tree area: Output 1, Output 2, etc.



NOTE

Save the configuration from temporary memory to permanent memory, overwriting the previously saved configuration.

CHAPTER 4
DL.CODE USER INTERFACE

DL.CODE DEVICE DISCOVERY WINDOW

After loading your specific device from the discovery window (drag & drop from the Device Selection Area), the DL.CODE Device Discovery window presents the following principal areas:

- **Main Menu and Toolbar Area** – allows access to the major program functions and commands. See "DL.CODE Main Menu and Toolbar" on page 34.
- **Device List Area** – shows all the discovered devices both on and off the LAN. The colored icons next to the device labels indicate network status, see "Device List Area" on page 30. Double-clicking or dragging an available device icon onto the Device Information Area connects it to DL.CODE for configuration.
- **Device Information Area** – this area shows all device specific information: Name, Model, Role, running software and version details.
- **Task Area** – Presents a graphical list of the main features to be performed upon device connection: create a new configuration (Stand Alone or Master/Slave), Open an existing configuration, perform PackTrack calibration, or switch to Monitor mode. These selections are also available in the File and Device Menus.
- **Control/Help Panel** – In the DL.CODE opening window this panel provides a Help description for Device Selection. Once a new or existing configuration is loaded, this is the key area which allows all the individual device configuration parameters to be set. Context sensitive Help is also available from this panel. See "Control/Help Panel" on page 32.
- **Status Bar** – a reserved area that keeps specific information about the connected device: Name, Model, SN, IP Address, Device SW version, user level, and DL.CODE program version always visible.

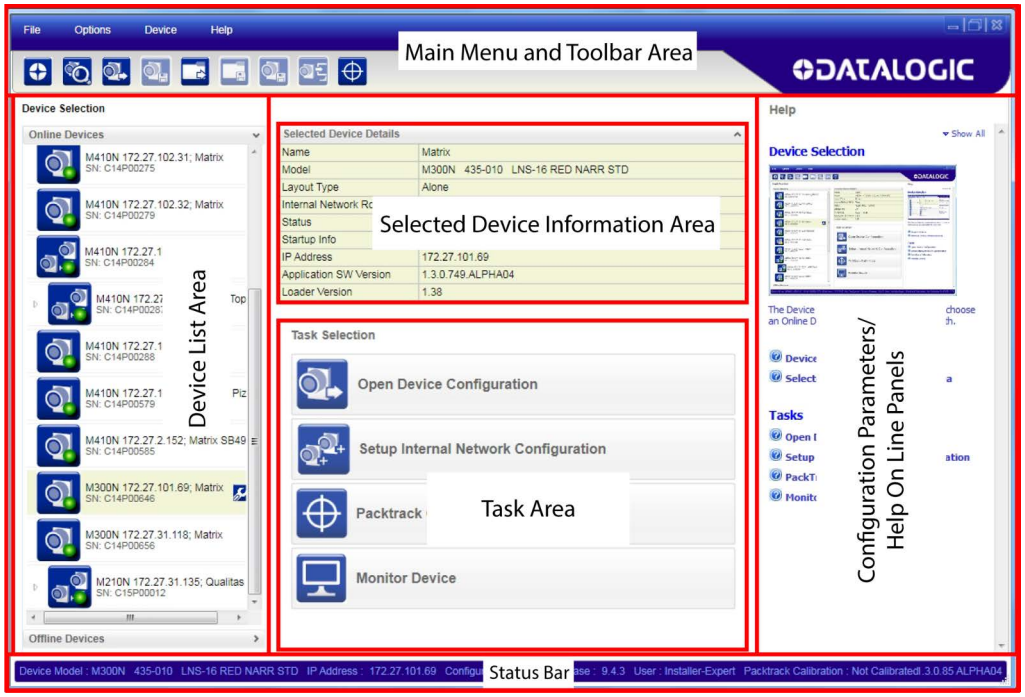


Figure 5 - Device Discovery Window Areas

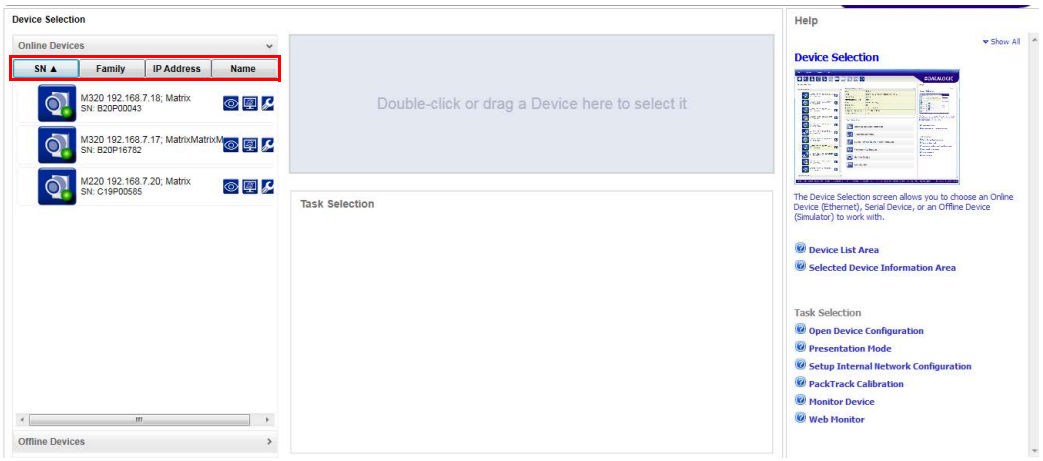
Device List Area

The Device List area shows all the discovered devices both on and off the LAN. The icons indicate the Device status as shown in the table below:

	Stand Alone device connected to the network with a valid LAN IP address. Available for configuration through DL.CODE. Stand Alone device connected to DL.CODE via Serial COM Port.
	Stand Alone device connected to the network with a valid LAN IP address but currently connected to DL.CODE by another User. Not available for configuration through DL.CODE (double-clicking or dragging the icon has no effect).
	Device connected to the network but without a valid LAN IP address (i.e. default IP address). The IP addressing parameters must be modified to connect to the LAN. See "Ethernet Device Discovery" on page 8. Device connected to a Serial COM port but not to DL.CODE.
	Offline device deactivated. Only one offline device at a time can be activated.
	Master device connected to the network with a valid LAN IP address. Available for configuration through DL.CODE.
	Slave device # connected to the Master via ID-NET and to the network with a valid LAN IP address. Available for configuration directly through DL.CODE.
	Slave device # connected to the Master via ID-NET network but not to the LAN. Available for configuration through the Master connected to DL.CODE.
	Slave device # not connected to the Master via ID-NET. Not available for configuration.

Online Device Sorting

From software version 1.9.2, the list of online devices can be sorted by Serial Number (SN), Family (device model), IP Address, and Device Name.

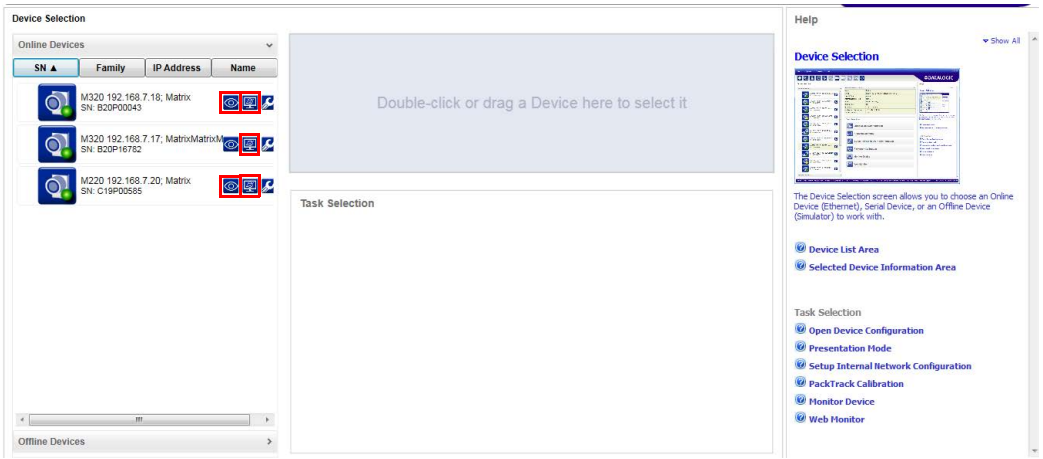


Sorting is only applied to first level devices, i.e. Standalone and Master devices, while Slave devices are always sorted by Slave address.

At each application launch, the last sorting used will be applied.

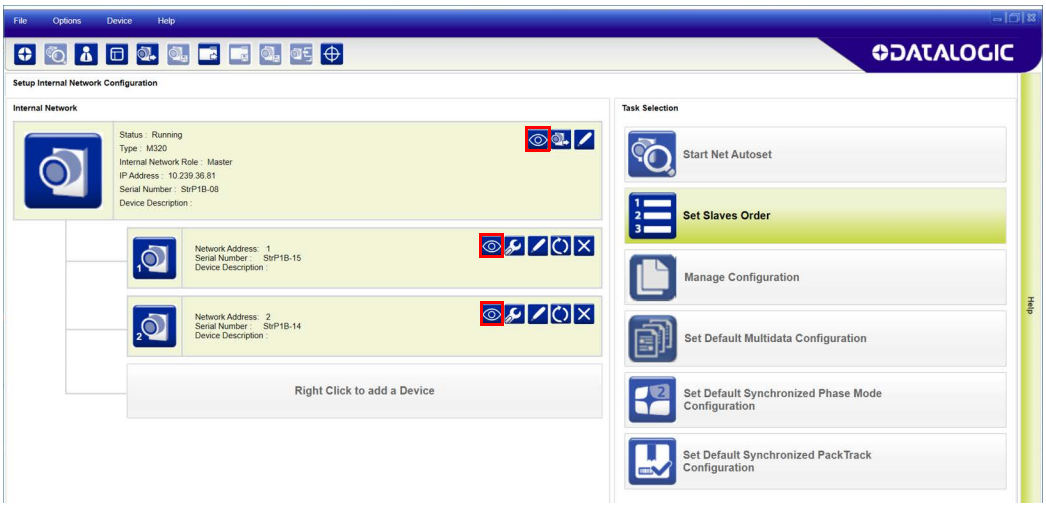
Wink and Web Monitor icons

The device tree area shows the Wink and Web Monitor icons next to all devices supporting these functions, as shown in the figure below:



Clicking on the Wink icon  makes the device screen or on-board LEDs flash for 3 seconds. Where available, the 360° Visual Feedback will blink too.

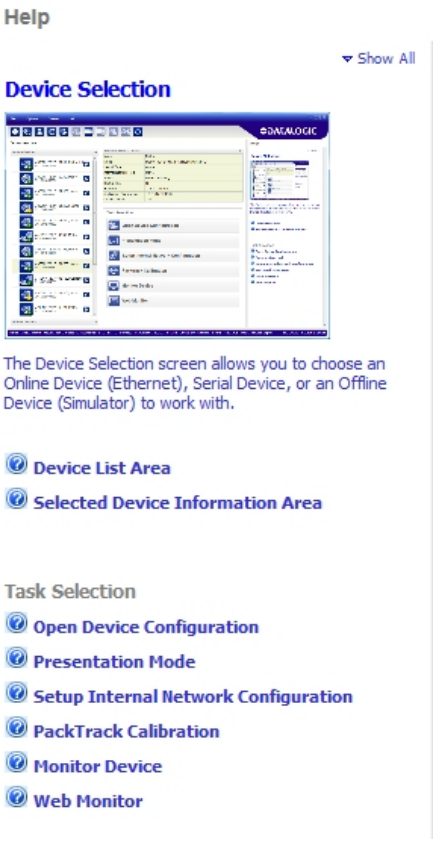
The Wink function is also available in the Internal Network Configurations view:



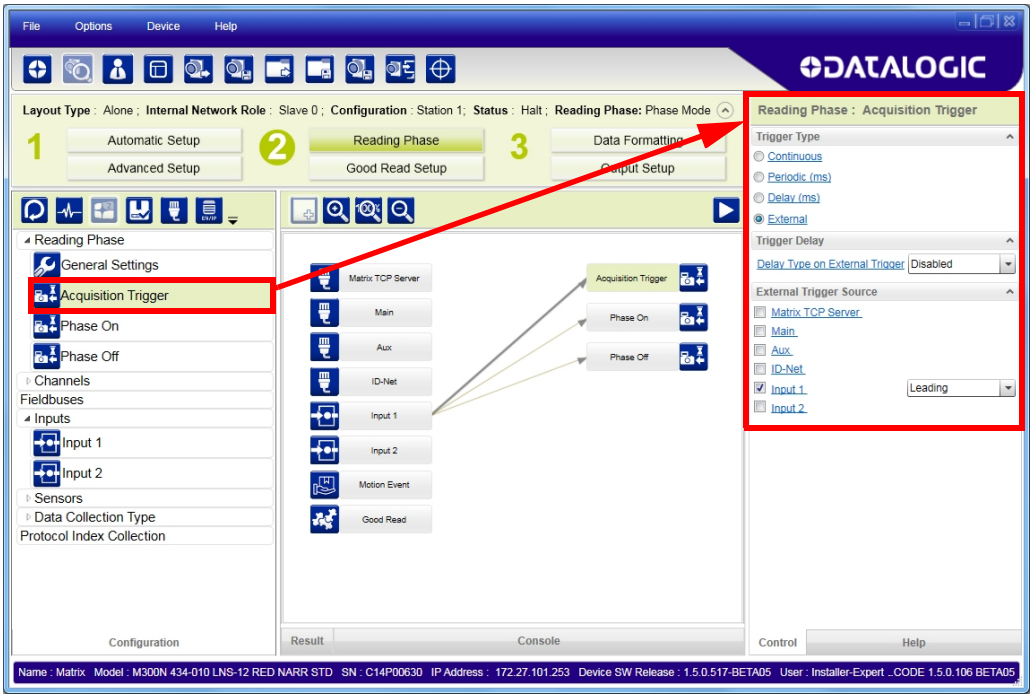
Control/Help Panel

Upon opening DL.CODE, the Device Selection Help Page is displayed with dropdown descriptions briefly explaining how to connect to your device and the various configuration selections. Click on the icon to open the dropdown description.

You can also click on the Show All link at the top of the Help page to open all collapsed text descriptions.



When a new configuration is created or an existing one is opened, the Control page showing all of the configuration parameters for the selected configuration step or item is displayed in the Help/Control area.



By clicking on any parameter name with a hyperlink, the relative contextual help page will open and present the specific parameter description.

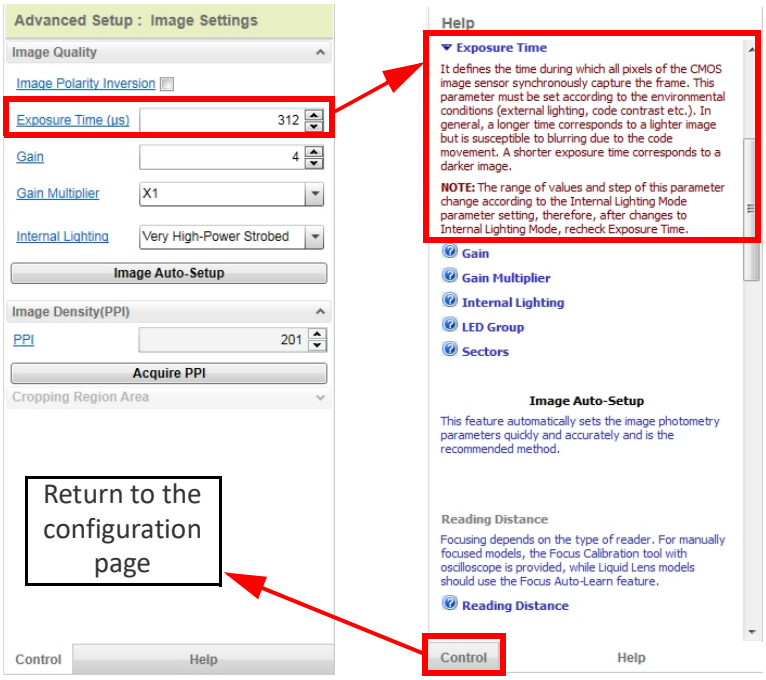


Figure 6 - Control Panel and Relative Contextual Help Page

To return to the parameter page, click on the **Control** bar.

DL.CODE MAIN MENU AND TOOLBAR

The Main Menu and Toolbar icons are located at the top of the DL.CODE window as shown below.

The Main Menu presents the following items:

File:

- **Getting Started**: returns to the initial Device Discovery window to load a different device. You will be prompted to Save or Discard the current configuration.
- **Open**: load a previously saved configuration from the device memory or from a .dlcfg file selected from a disk/directory of your choice.
- **Save**: save the current configuration to the device memory or to a .dlcfg file selected from a disk/directory of your choice.
- **Setup Internal Network**: sets the current device's internal network Role to Master and performs the Net Auto-set feature to automatically recognize its connected ID-NET Slaves. The Internal Network Configuration window allows ID-NET configuration management. See "Multi Device Configuration Options" starting on page 131 for details.
- **Monitor**: sets the device to run mode for testing configuration results. An image window is displayed along with Statistics, Diagnostics and a Console for output message verification. See "Monitor" starting on page 112 for details on using and configuring the Monitor.
- **Exit**: exits the DL.CODE User Interface.

Option:

- **Change Language**: allows you to change the display language used for DL.CODE in real time. The selected language will also be used for successive sessions.
- **UI Settings**: opens a window where various settings can be made regarding DL.CODE presentation on the PC. The following DL.CODE features can be configured: console presentation, Image Buffer positioning and behavior, Simulator Settings, and selecting which features to enable for viewing. The Monitor can also be configured from this window by selecting the Monitor tab.
- **Change Log Level**: allows the configuration log level to be changed between Verbose, Information and Error.
- **Change User**: allows the configuration access level to be changed between Basic-User (no parameters modification allowed), Basic-Installer (medium level of parameter modification allowed) and Expert-Installer (maximum level of parameter modification allowed).
- **Restore UI to Default**: restores all UI panels to their default positions.

Device:

- **Connect to Device**: if not already connected to a device, this allows you to connect to a device on the LAN by inputting its IP Address into the dialog box and clicking Connect.
- **Find Devices**: searches for new devices connected to the LAN without disconnecting from the current device.
- **Settings**: opens device configuration windows for configuring Environment parameters (see "Device Configuration Settings" starting on page 186), Advanced Configu-

ration Settings, resetting Statistics and/or Diagnostic counters and viewing HMP shortcuts.

- **Update Package**: allows updating the device firmware (application program, schema, etc.) either upgrading or downgrading to a different version, see also Caution below. For the complete procedure, see "Update Package" on page 199
- **Multi-Device Package Update**: allows updating multiple devices simultaneously. This command is only available when no device is currently connected to DL.CODE. Update firmware (application program, schema, etc.) either upgrading or downgrading to a different version, see also Caution below.
  After selecting the package to update, select the devices from the list in the Mult-Device Package Update window. When updating is completed the window reappears indicating success or failure for each device.
- **Change Current Configuration**:
- **Restart Device**: performs a software reset on the loaded device.
- **Backup/Restore**: manages all the backup and restore options to and from the device as well as restoring the default settings.
- **RAM Image Buffer Settings**: opens the RAM Image Buffer Management window to either save or discard images in the device's RAM Image Buffer.
- **PackTrack Calibration**: launches the Packtrack Calibration procedure. See "Packtrack Calibration" starting on page 155 for details.

Help:

- **About**: opens the information window containing the DL.CODE program release version number.
- **DL.CODE User's Manual**: opens this manual.

Toolbar buttons:

	Getting Started: Disconnects the current device and returns to the Device Discovery window. If the current configuration hasn't been saved you will be prompted to do so before disconnecting.
	Find Devices: Executes a device discovery to find new or modified devices on the LAN without disconnecting from the current device.
	Change User: Select a different User configuration access level.
	Restore UI to Default Layout: resets all the various DL.CODE resizable graphical areas (window panels) to their default layout positions.
	Open Device Configuration: Open a previously saved configuration from the device memory. You can create a new configuration by opening a configuration in the list and then saving it with a new name.
	Save on Device: Save the current configuration to the device.
	Load from PC: Open a previously saved .dlcfg configuration file from the local PC or from a remote network location.
	Save on PC: Save the current configuration to a .dlcfg file on the local PC or to a remote network location.
	Save Configuration in Temporary Memory.
	Configuration/Monitor Switch: toggles between the Configuration environment and the Monitor environment. See "Monitor" starting on page 112 for details.
	PackTrack Calibration: starts the PackTrack Calibration feature. See "Packtrack Calibration" starting on page 155.

User Levels

DL.CODE has a 3-level user interface (**Basic User** level, **Basic Installer** level, **Expert Installer** level).

Each level can be accessed by selecting it from the Options Menu.

The User levels have the following access rights.

Basic User: only access to the Monitor feature is given. No device configuration.

Basic Installer: a subset of program features and configuration parameters is allowed.

No Master/Slave configuration, no multiple Image Settings configurations, no Postal Codes configuration, no Fieldbus configuration.

Expert Installer: access is given to all program features and configuration parameters.

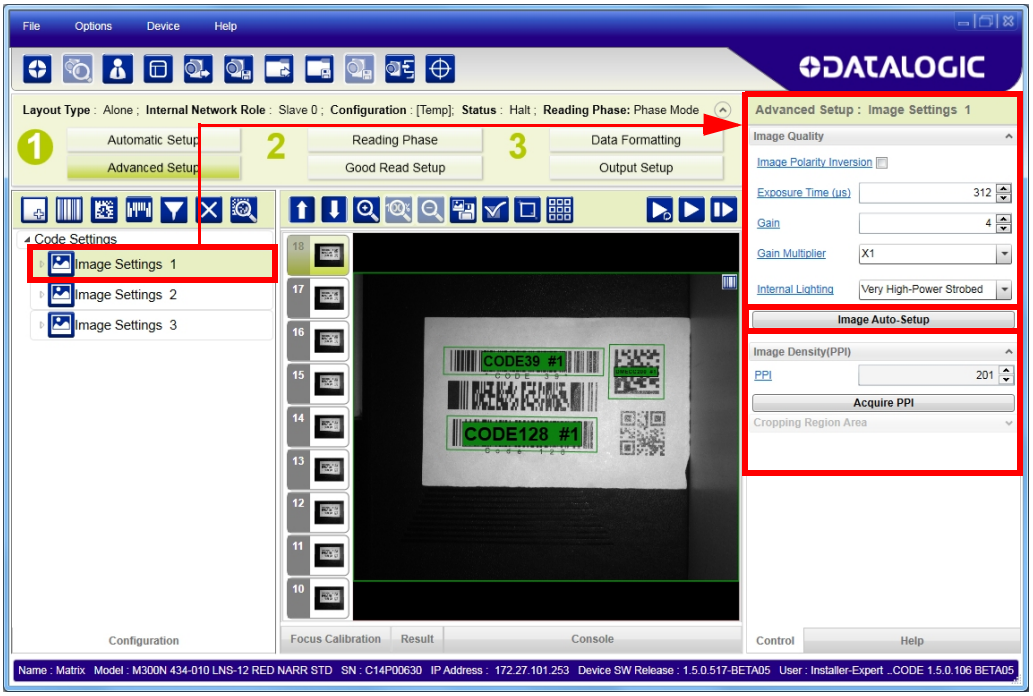
MULTI IMAGE ACQUISITION SETTINGS

When controlled variable conditions occur in the application, Multiple ***Image Acquisition Settings*** can be defined to create a database of parameter groups that handle each specific application condition. This database of pre-defined settings improves system flexibility and readiness by being applied either automatically or selectively by an activation event.

For example, an application may have several stable but different lighting conditions which require different lighting options. One Image Acquisition Setting could enable and use an internal illuminator and another setting could enable and use an external lighting system.

This feature is available for all Operating Modes.

Image Settings are found in the DL.CODE **Advanced Setup** step. Up to 10 different Image Settings can be configured by adding them with the Add icon.



For each Image Setting condition start Image Auto Setup, select the Static or Dynamic Self-Tuning option and Apply it.

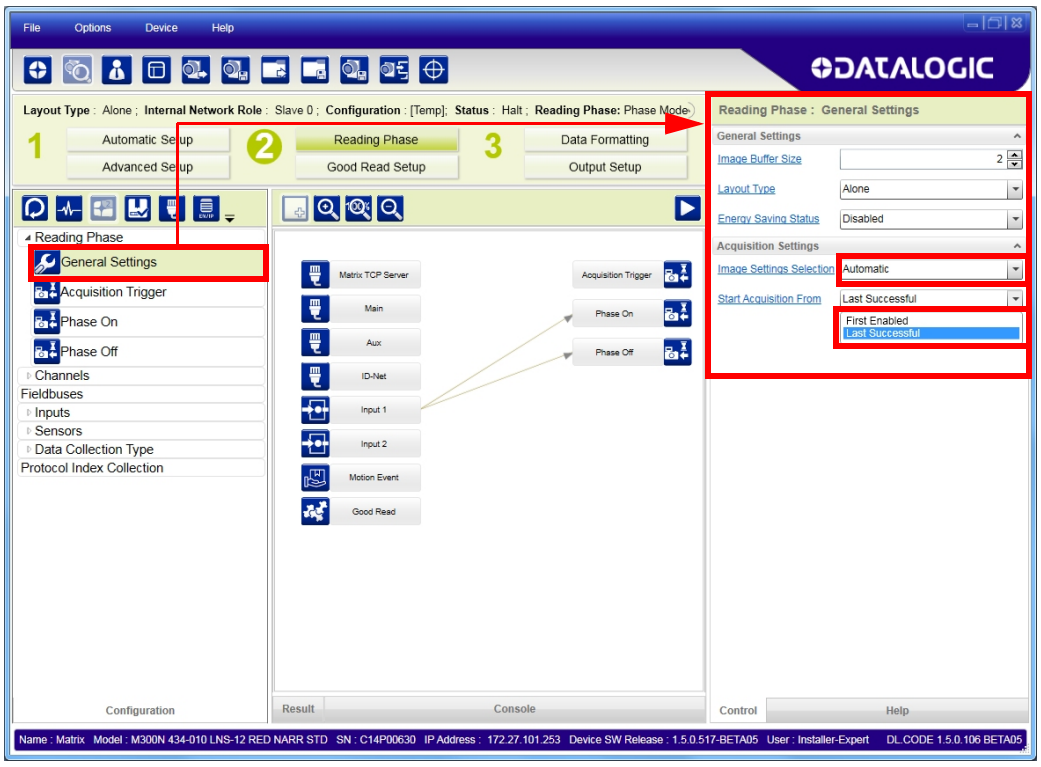
NOTE

The Automatic Setup procedure can be used for each Image Setting that is added through the Advanced Setup group, however only one code will be associated with each Image Setting.

Automatic Image Settings Selection

If we don't know from one item to the next which reading condition will be presented, we will cycle through the pre-defined database of Image Settings (one per acquisition) in order to automatically capture the correctly lighted image over the course of several acquisitions. When the correct condition is matched, the result should be able to produce a Good Read.

When the **Image Settings Selection** is **Automatic** (default), then these Image Settings groups will be used cyclically.



NOTE

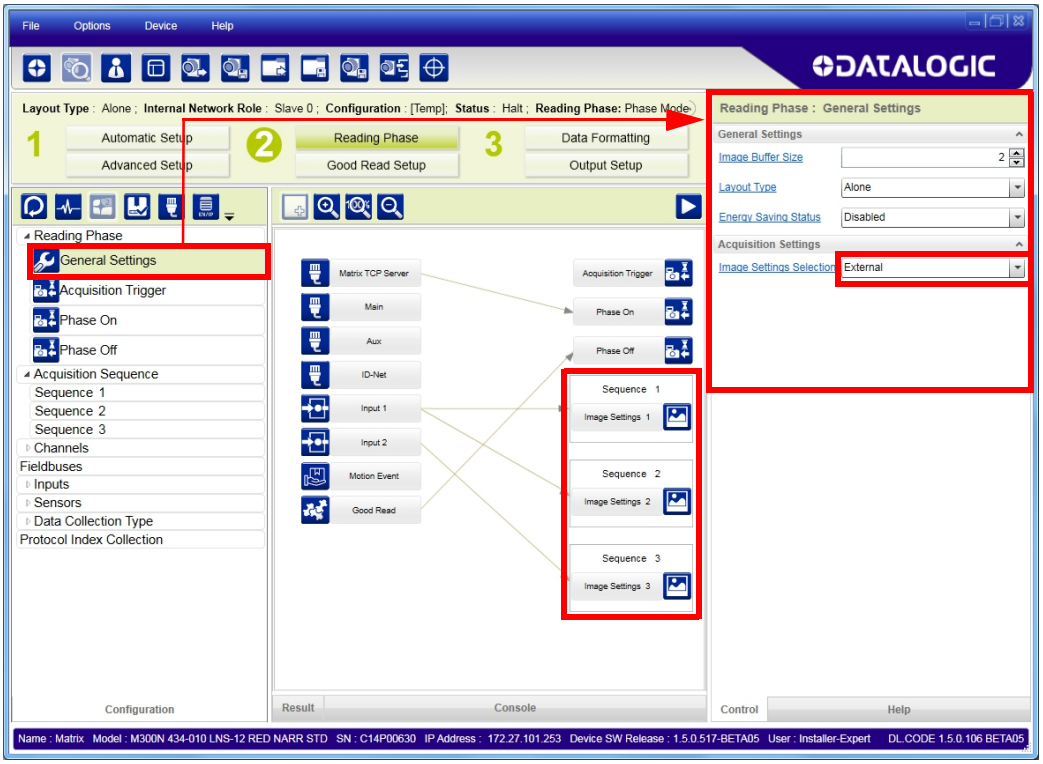
Applications typically require more than one acquisition to obtain sufficient Good Read percentages. This means that for *N* acquisitions we will surely have captured the correct lighting condition but we will also have captured some acquisitions with the wrong lighting condition. A consequence then is that we cannot run these types of applications at the maximum speed because not all the acquisitions are useful to us. We must wait for a sufficient number of acquisitions that will guarantee a Good Read result.

If items to be read having the same lighting conditions are grouped together, then we can improve the read rate through the **Start Acquisition From** parameter. By choosing the **Last Successful** value, we will start with the Image Setting that last produced a Good Read. For this group of items the last Image Setting used will be correct for the next item and so we start each cycle with the acquisition that will potentially produce a Good Read.

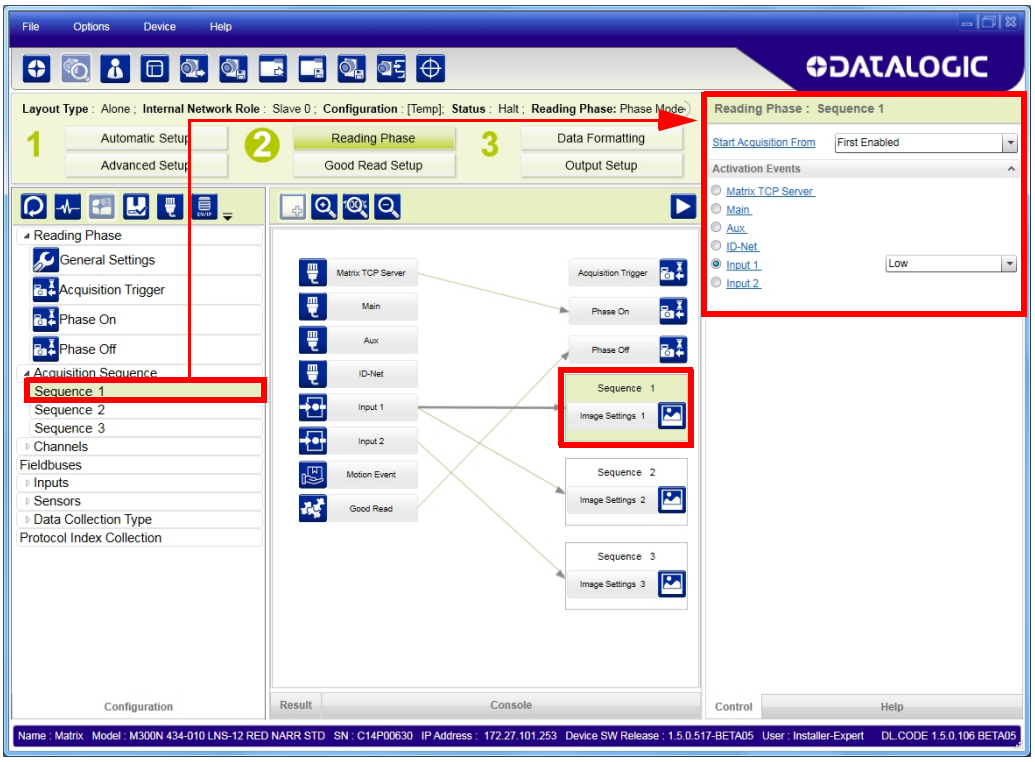
External Image Settings Selection

There are some applications where the lighting conditions are known before each item is read and therefore we can pre-select the correct Image Setting from an external source.

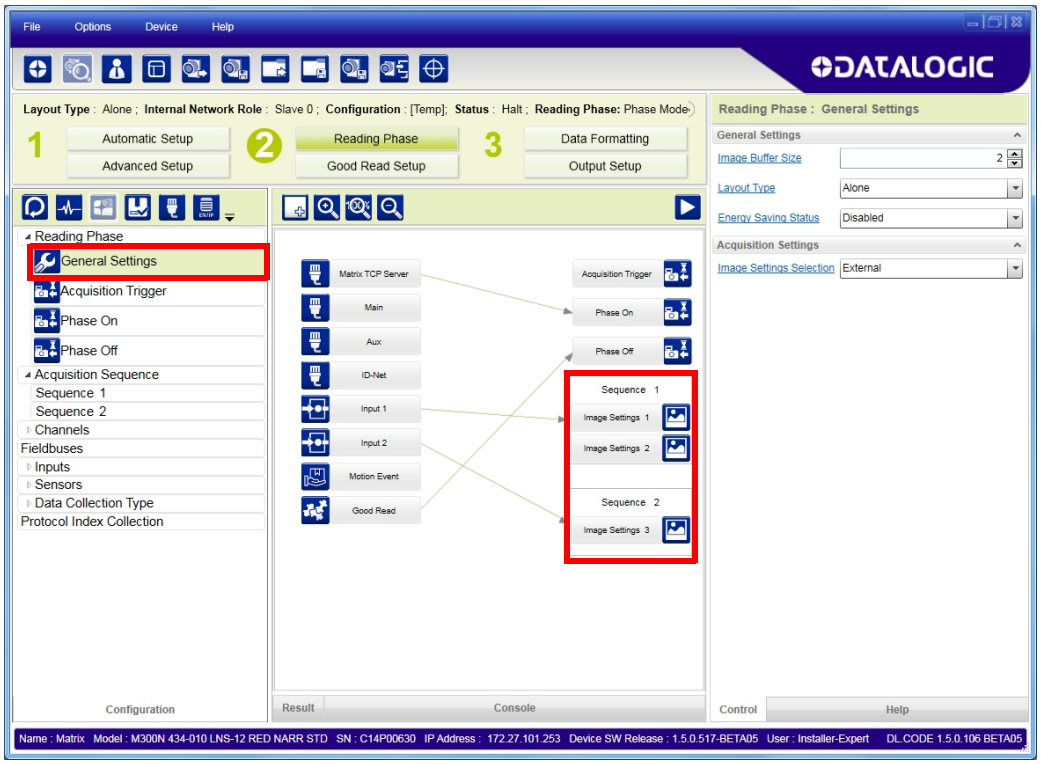
When the **Image Settings Selection** is **External**, Acquisition Sequences are created and by default each Image Setting has its own Acquisition Sequence.



Each **Acquisition Sequence** can be activated exclusively by a single event, either through a string from an available communication channel or by a digital input.



Alternatively, a hybrid configuration can be made where more than one Image Setting can be grouped into an Acquisition Sequence by dragging it into the desired Sequence box. Select the empty Sequence box and delete it with the delete key.



Each **Acquisition Sequence** can be activated exclusively by a single event, either through a string from an available communication channel or by a digital input.

In this case the **Start Acquisition From** parameter can improve the read rate for that Sequence. It has no meaning for a Sequence containing only one Image Setting.

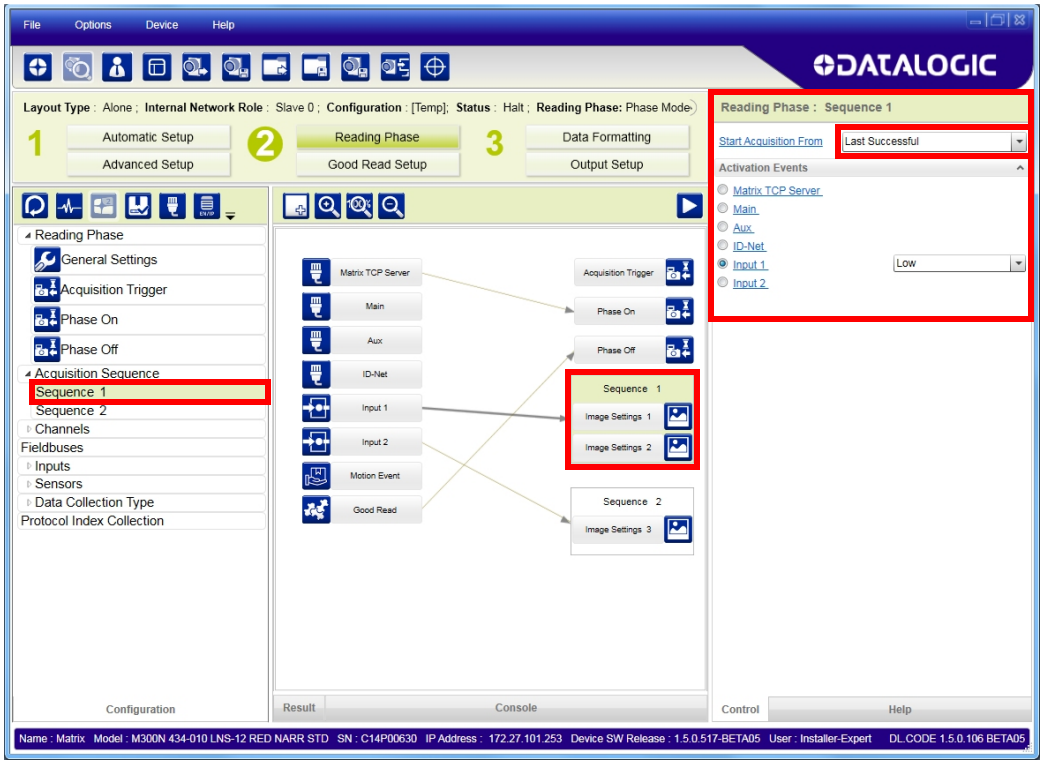
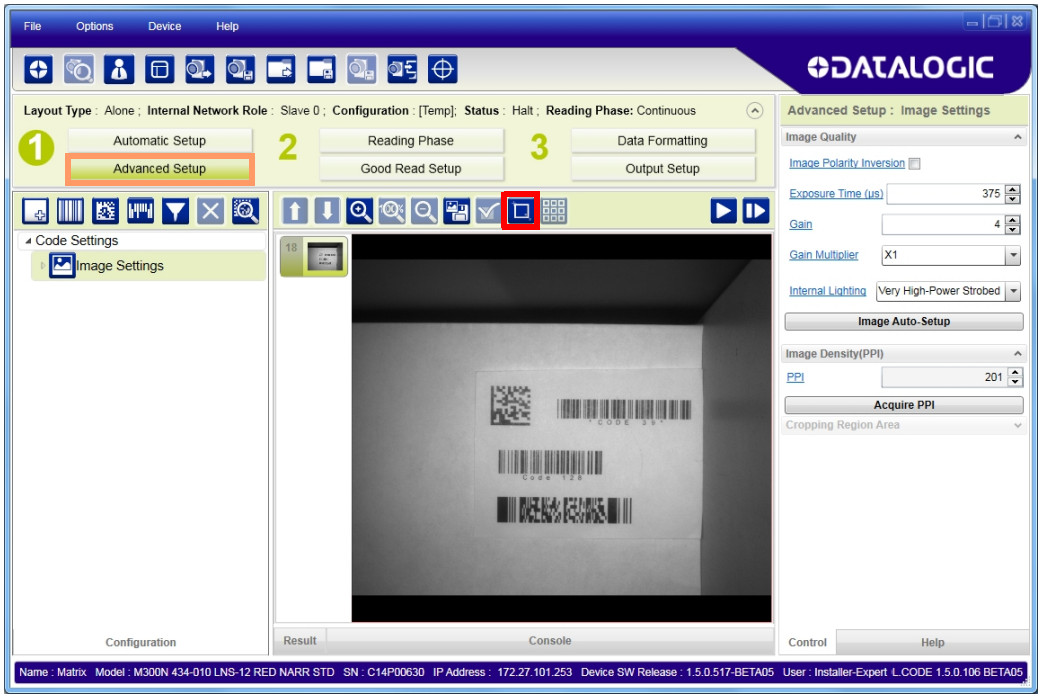


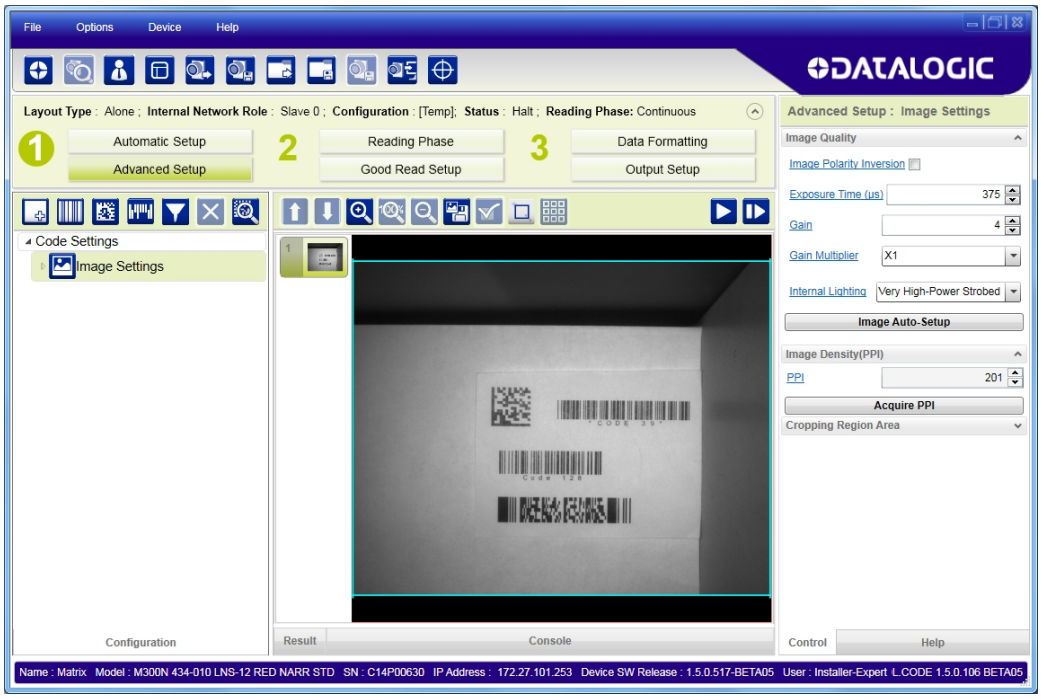
IMAGE CROPPING

In some applications, the Image Cropping feature in DL.CODE can help to increase decoding and result performance. Image cropping is performed from the Advanced Setup tab by clicking on the Add Cropping Region icon as shown below.

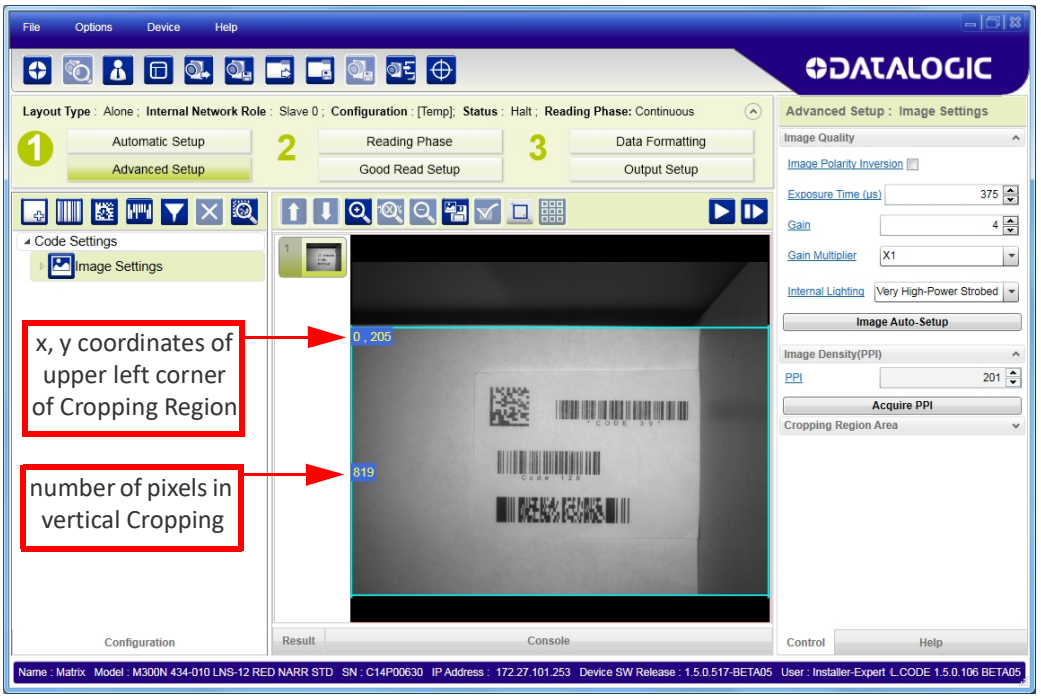
Image cropping allows reducing the Image processing area from the full FoV to a smaller area where codes are present. By excluding portions of the FoV, processing time is reduced.



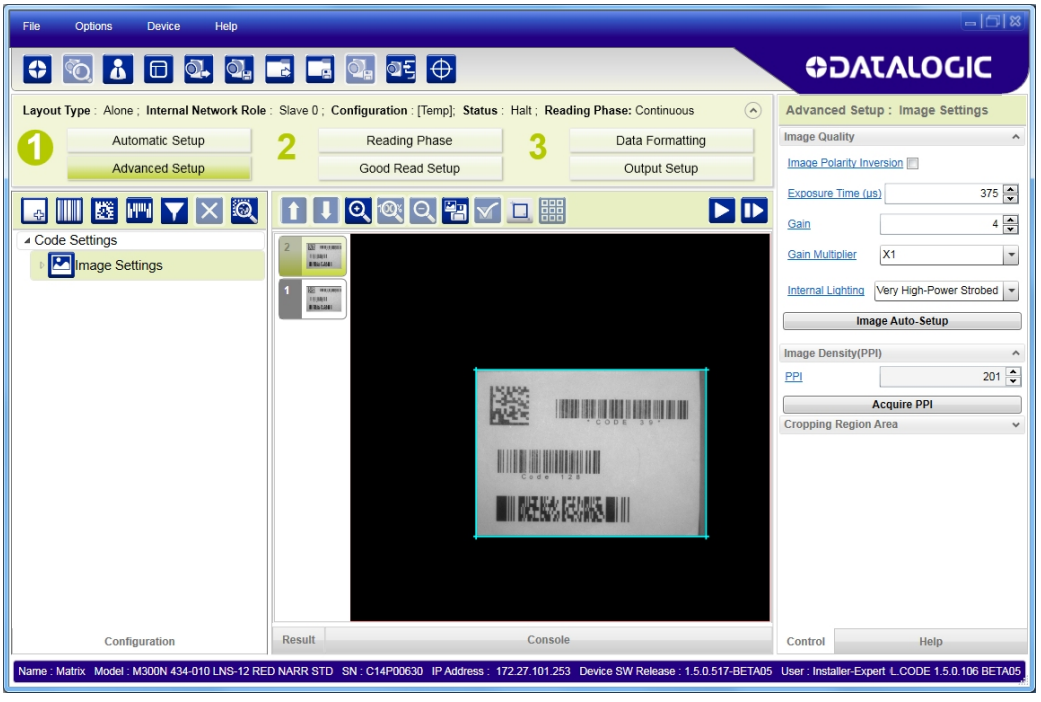
After clicking the Add Cropping Region icon, a blue border appears which by default is equal to the FoV.



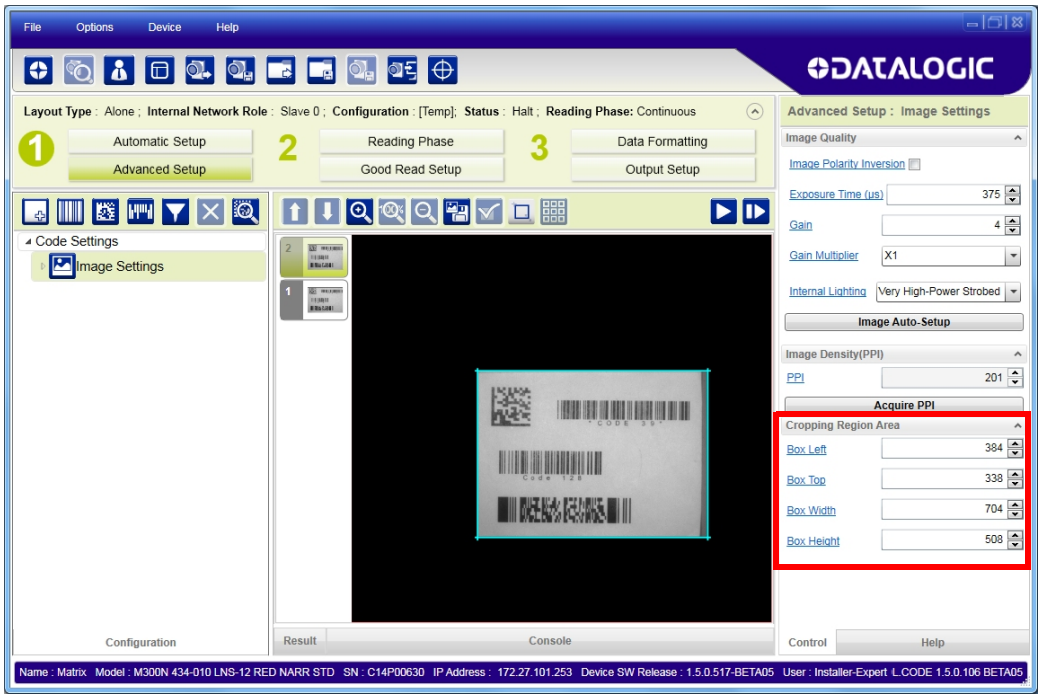
By dragging the edges with the mouse (resizing) you can crop the image to a specific location where codes are present. The numbers in the blue boxes refer to pixel references.



The cropped area can be moved by dragging its center.



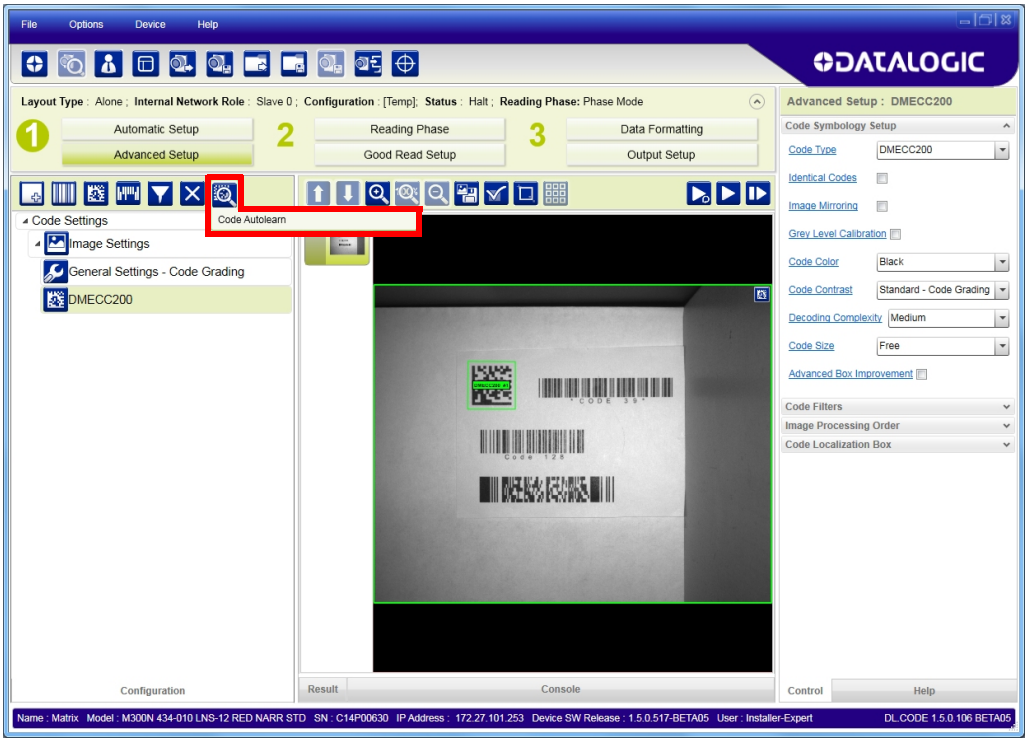
You can also set the cropped image size and position through the Cropping Region Area group of parameters; size = **Width** and **Height**, position = **Left, Top** (x,y) coordinates.



CODE AUTOLEARN FEATURE

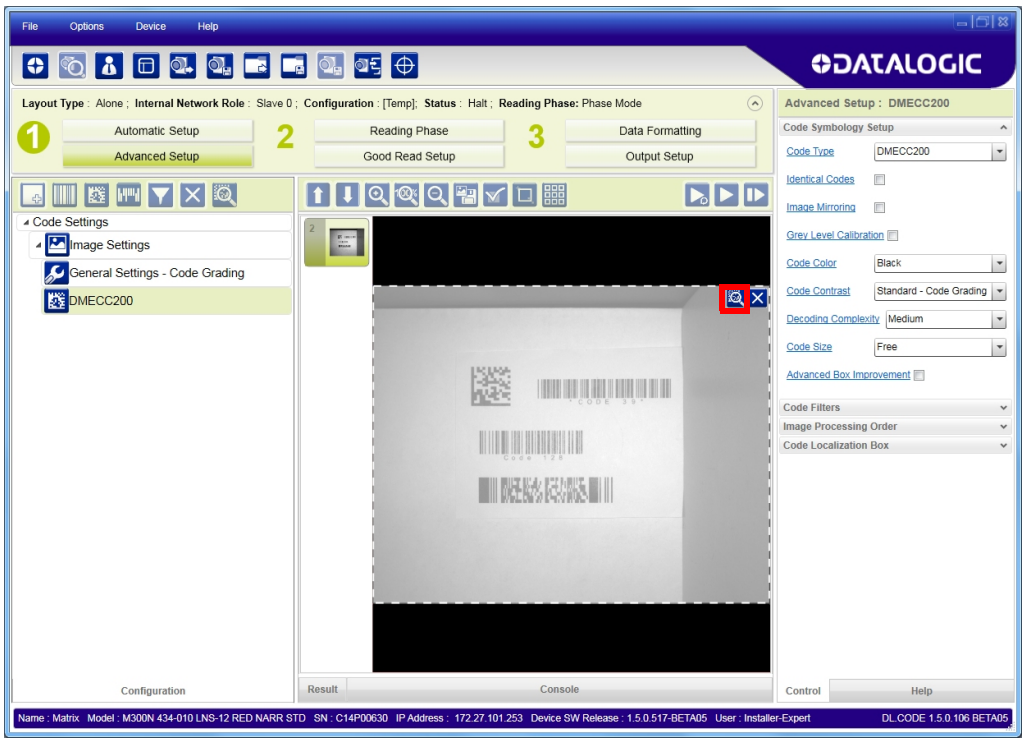
From the Advanced setup page, you can run the Code Autolearn feature which will recognize all the codes present in the captured image.

1. From the Advanced Setup page, click on the Autolearn icon and select Code Autolearn.



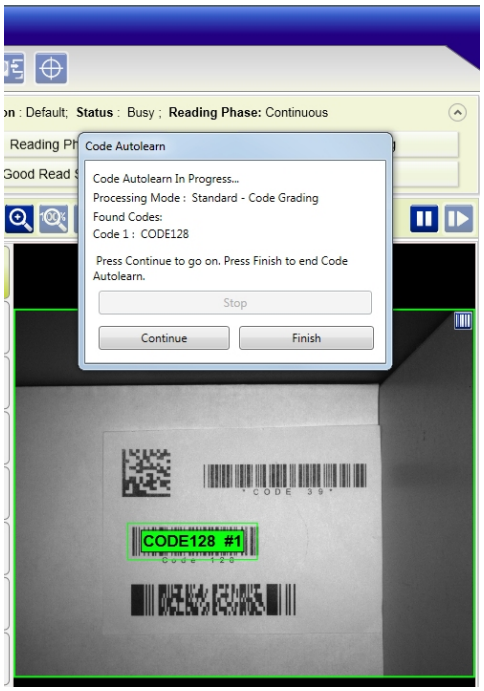
The Autolearn region (equal to the FoV) is shown in grey. You can reduce and/or move the search area by dragging the borders or the center of the area with the mouse.

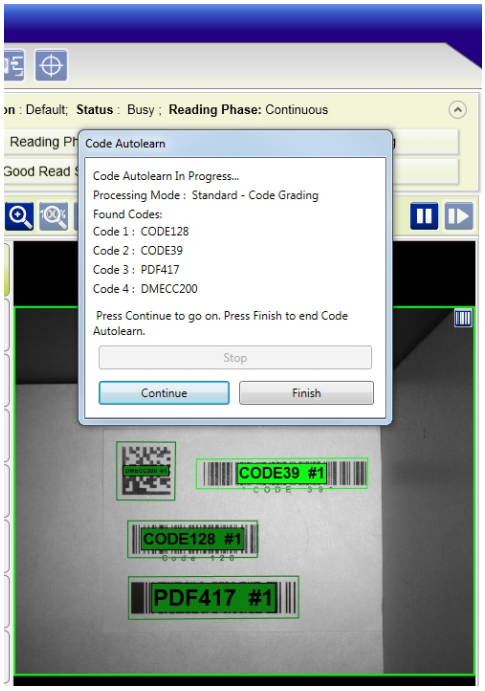
- Reducing the area can reduce the time necessary to find the code(s).
- Moving the search area allows finding specific code(s) in the image.



2. Whether the area is reduced or not, you can start the Autolearn feature by clicking on the Autolearn icon in the display area.

Each Autolearn iteration locates a single code symbology and you will be prompted to Continue (if you need to find other codes) or to Finish.





3. When you have located all the code symbologies, click on Finish. You will be prompted to choose a saving selection.

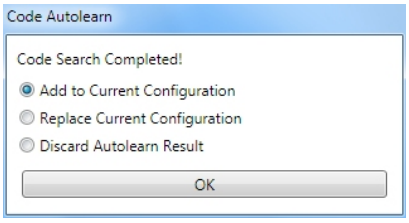


IMAGE FILTERING

Image Filtering is typically done in DPM applications where the marking technique produces module shapes or textures that can make decoding difficult. Special DPM algorithms are provided to improve decoding as well as pre-processing Image Filters which modify the image to compensate for defects.

The following paragraphs detail the DPM parameters used to enhance decoding capabilities.

DPM Algorithms

For **Data Matrix** family codes the ***Decoding Complexity*** parameter is available when Processing Mode is set to Standard and selects the decoding algorithm according to the printing/marking technique used to create the symbol and on the overall printing/marking quality.

The possible selections progress from Low to Very High where Low can improve decoding time for good print/mark quality and/or relatively normal size codes. This is the default setting. Very High can improve the decode rate for low print/mark quality and/or small size codes. This algorithm is much more aggressive but in general it may have longer decoding times than the lower complexity algorithms. **To minimize decoding time it is better to select the lowest value that still guarantees good decoding.**

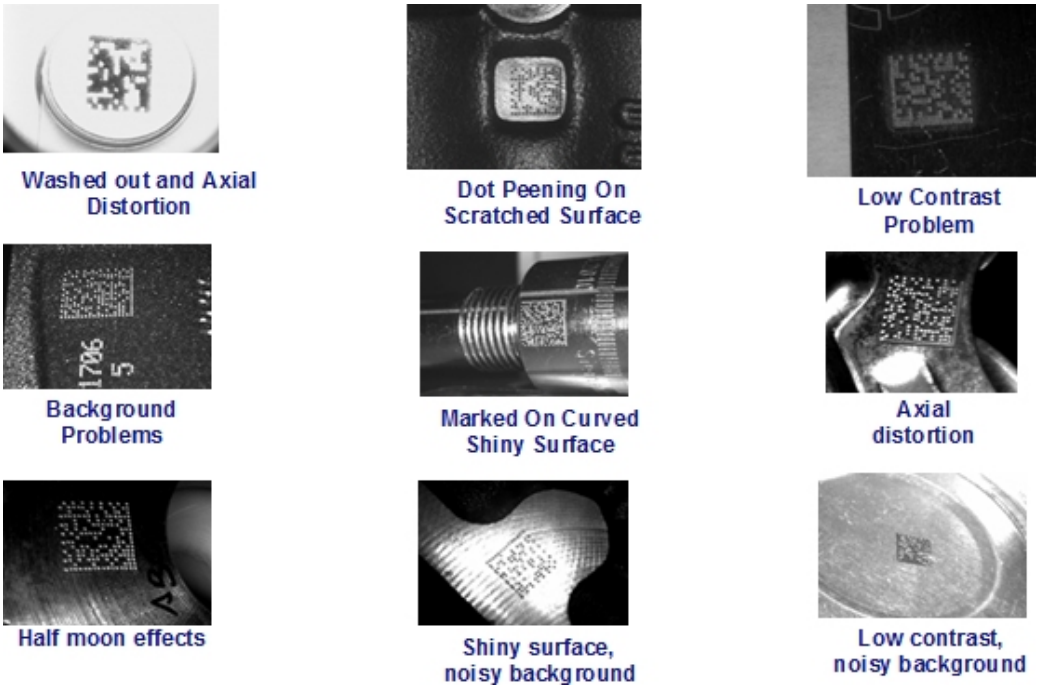


Figure 7 - Problematic Direct Part Marking Examples

For **QR** code the ***Decoding Method*** parameter allows the Dot Peen Decoding algorithm to be selected which improves the decode rate for low quality Direct Part Mark codes and in general for Direct Part Mark codes with dot peening type module shapes.

Image Filters

In DL.CODE, when Image Filters are used, they are always applied relative to a specific code symbology or group of symbologies depending on the nesting logic applied to the tree structure. See “Image Filter Setting Examples” on page 53.

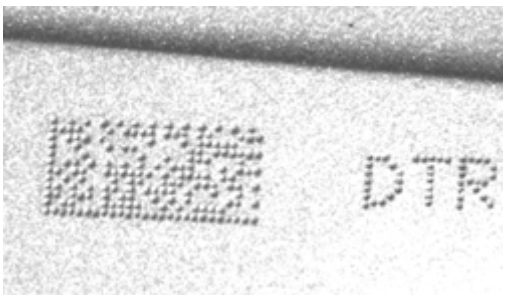
The following Image Filters can be applied to DPM codes to improve decoding.

Image Filter

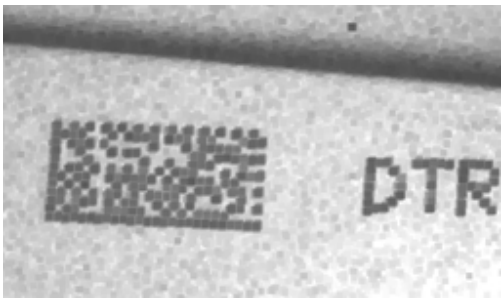
Sets the filter to be applied to the image before being processed. This parameter can be used to successfully decode particular ink-spread printed codes (ex. direct part mark codes).

Different filters can be applied to a single code or group of codes in one or more *Image Settings*. See “Image Filter Setting Examples” on page 53.

The ***Erode*** filter enlarges the image dark zones to increase readability.



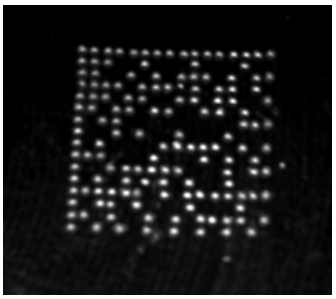
Before – No Read



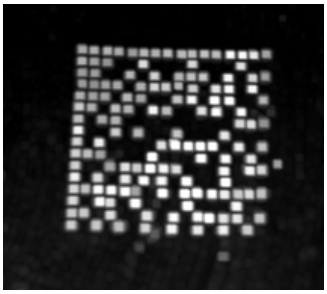
After – Readable

Erode

The ***Dilate*** filter enlarges the image white zones to increase readability.



Before – No Read

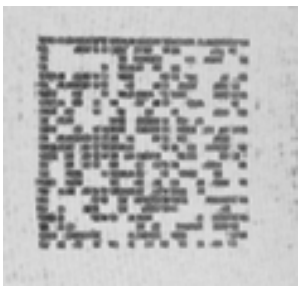


After – Readable

Dilate

The ***Close*** filter eliminates dark areas (defects) in the white zones of the image.

Before – No Read

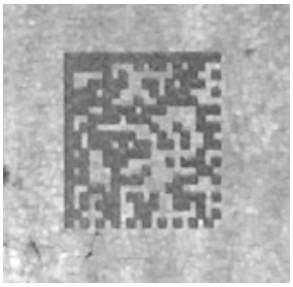


After – Readable

Close

The ***Open*** filter eliminates white areas (defects) in the dark zones of the image.

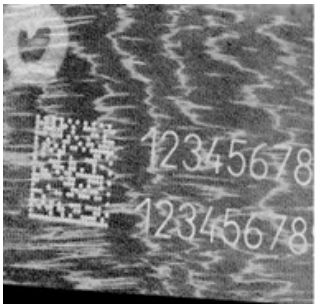
Before – No Read

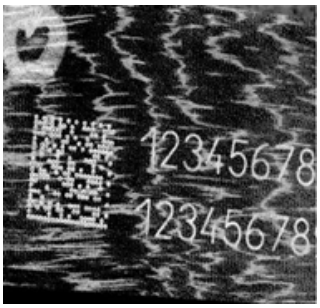


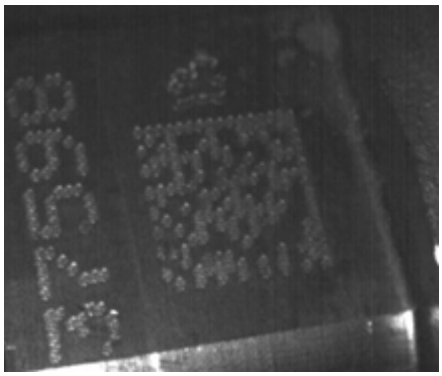
After – Readable

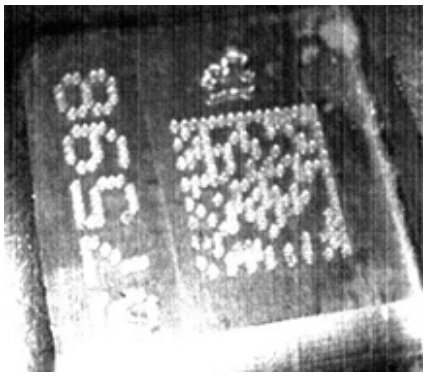
Open

The Contrast Stretching filter maximizes image contrast.









Before – No Read

After – Readable

Contrast Stretching

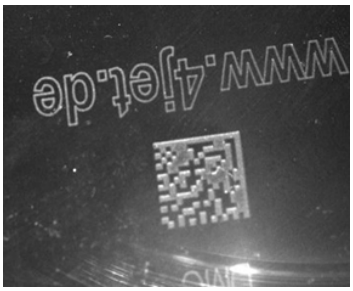
The ***Histogram Equalization*** filter makes the gray level distribution uniform.

Before – No Read

After – Readable

Histogram Equalization

The **Smoothing** filter deletes small (insignificant) details in the center of the image.

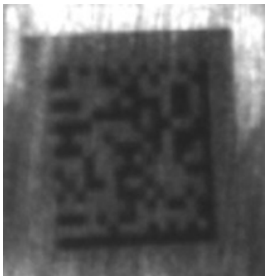


Before – No Read

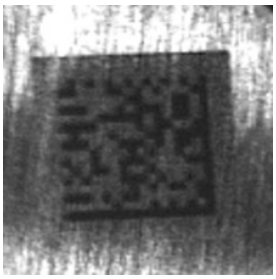
After – Readable

Smoothing

The ***Sharpening*** filter improves out of focus images.



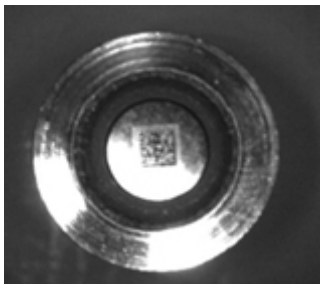
Before – No Read

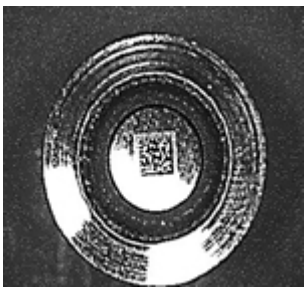


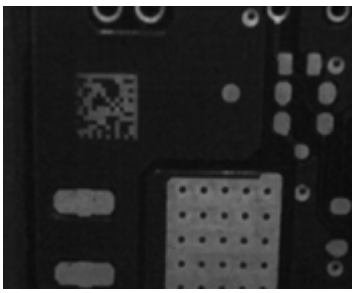
After – Readable

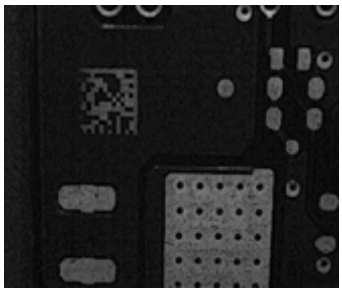
Sharpening

The ***Deblurring*** filter improves blurred images.









Before – No Read

After – Readable

Deblurring

The ***Black Enhancement*** filter produces a nonlinear increase in the black level for light images.

Before – No Read

After – Readable

Black Enhancement

The ***White Enhancement*** filter produces a nonlinear increase in the white level for dark images.

Before – No Read

After – Readable

White Enhancement

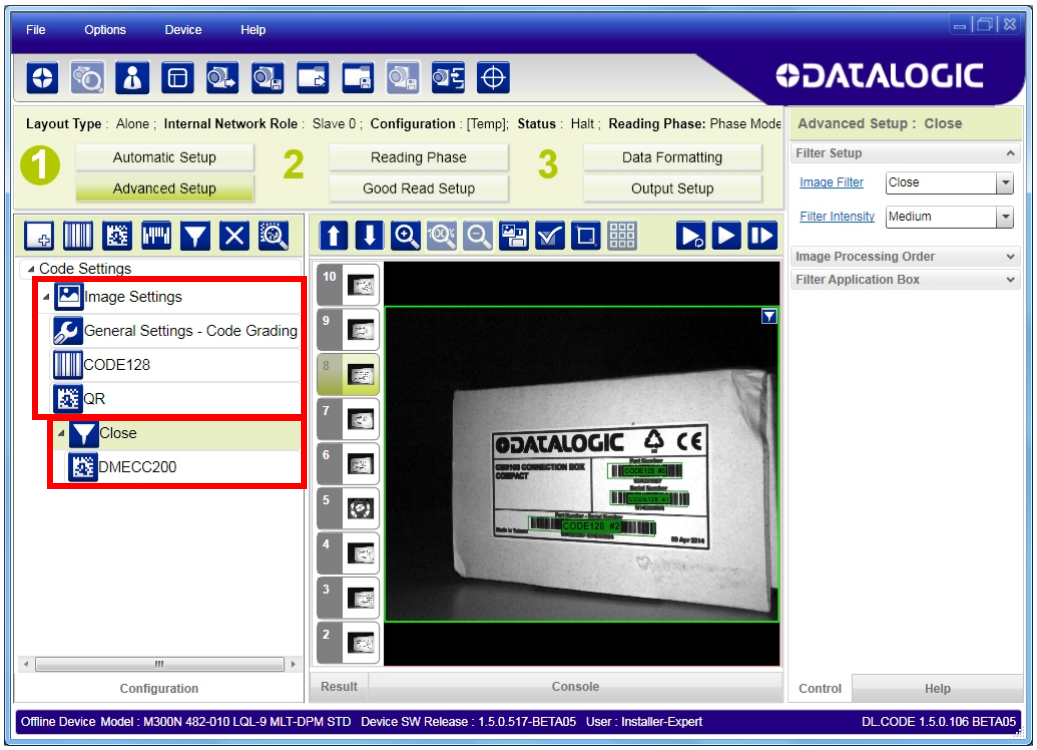
Image Filter Setting Examples

Example 1

To demonstrate how to apply an Image Filter through DL.CODE, the following example shows a hypothetical application in which a Data Matrix ECC 200 DPM code and printed label codes (QR Code and Code 128) must be decoded in the same image. In this example the codes can be found in any area within the image.

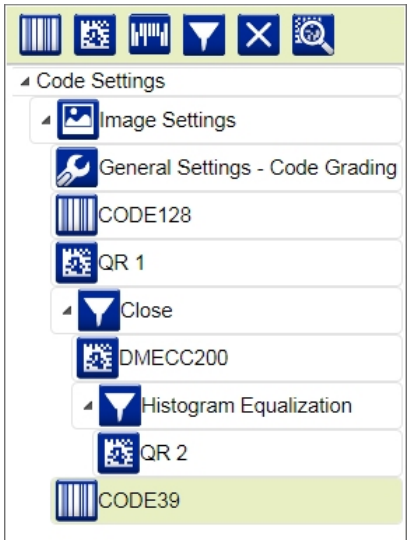
To correctly decode the Data Matrix ECC 200 DPM code, an Image Filter needs to be applied.

Since all codes can be found in any area of the image, both Code Localization and Filter Application boxes will be left at their default values, all covering the entire image area. The resulting image is shown below.



NOTE

By leaving the Filter Application box at its default value, covering the entire image area, you will see the visual filter effects on the entire image, seemingly affecting the other codes, however this pre-processing filter is only applied to the Data Matrix ECC 200 code as shown in the configuration parameter tree. The filter is not applied to Code 128 and QR Codes.

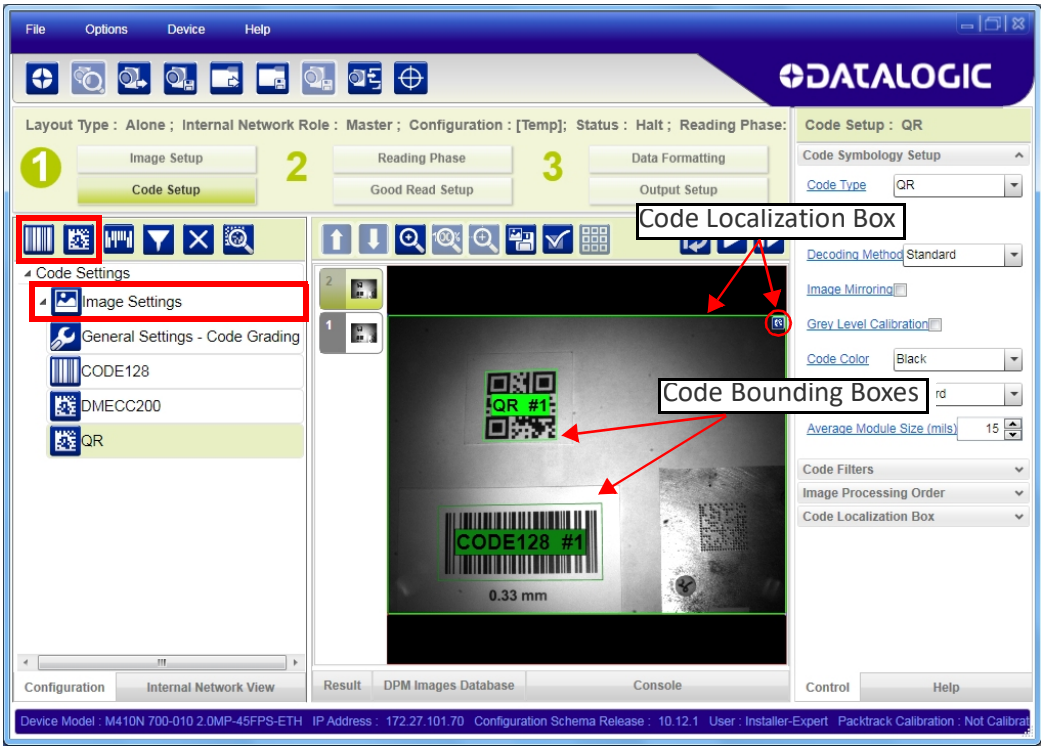


Since the visual effects in the image area are overlapped, you should not use this as a reference to determine how and where Image Filters are applied. You must read Image Filter application from the Tree structure in the configuration parameter tree area. The example in the figure to the left shows:

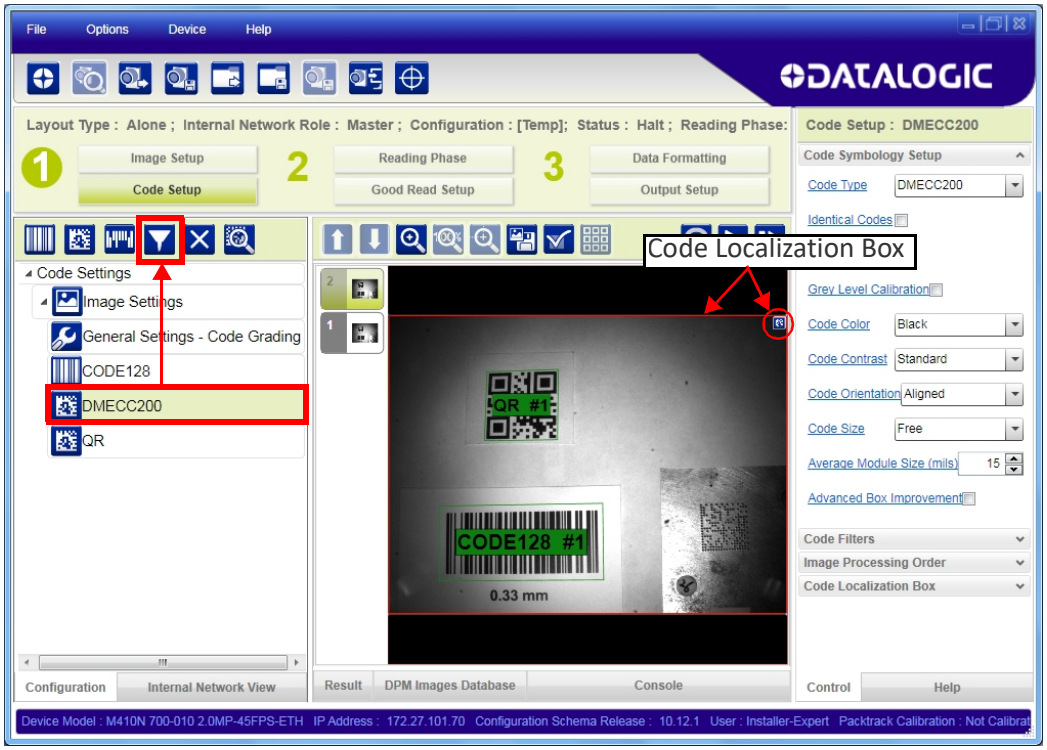
- Code 128, QR1 and Code 39 at the Image Settings level = No Image Filters applied.
- DMECC200 has the Close Filter applied to it.
- QR2 has the Histogram Equalization Filter applied to it and it is also nested under the Close Filter, so both filters apply to QR2.

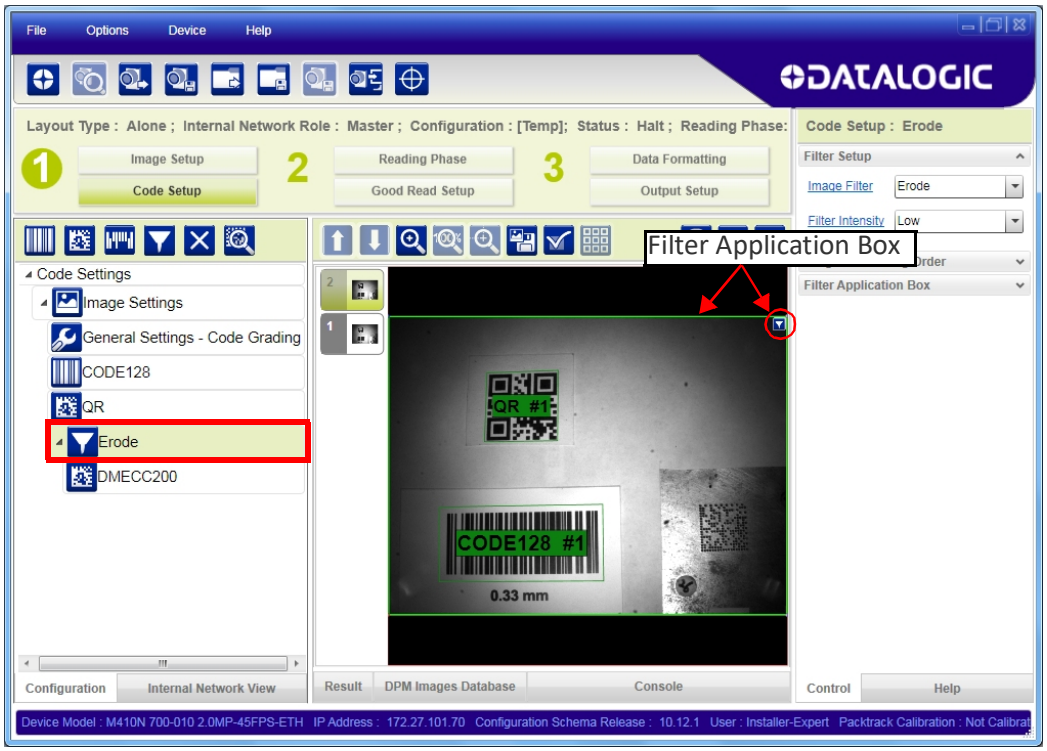
The following steps are taken to apply the Close filter to Example 1:

1. From the Image Settings group, add the Code Symbologies that must be decoded: Code 128, DMECC200 and QR Code. The Code 128 and QR Codes are successfully decoded; Code Localization boxes as well as code bounding boxes are colored green.

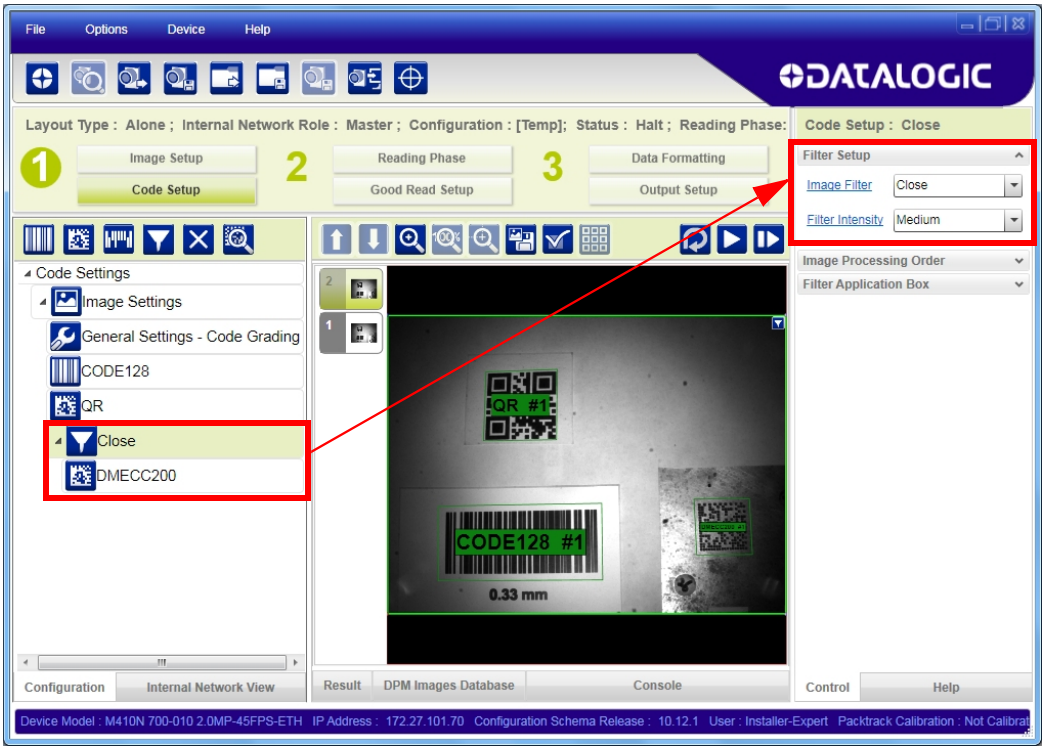


2. Select the Data Matrix ECC 200 code and add the Image Filter. Notice that the Data Matrix DPM code is not successfully decoded, Code Localization box is colored red.





3. Select the desired Image Filter Type and Intensity for correct decoding.

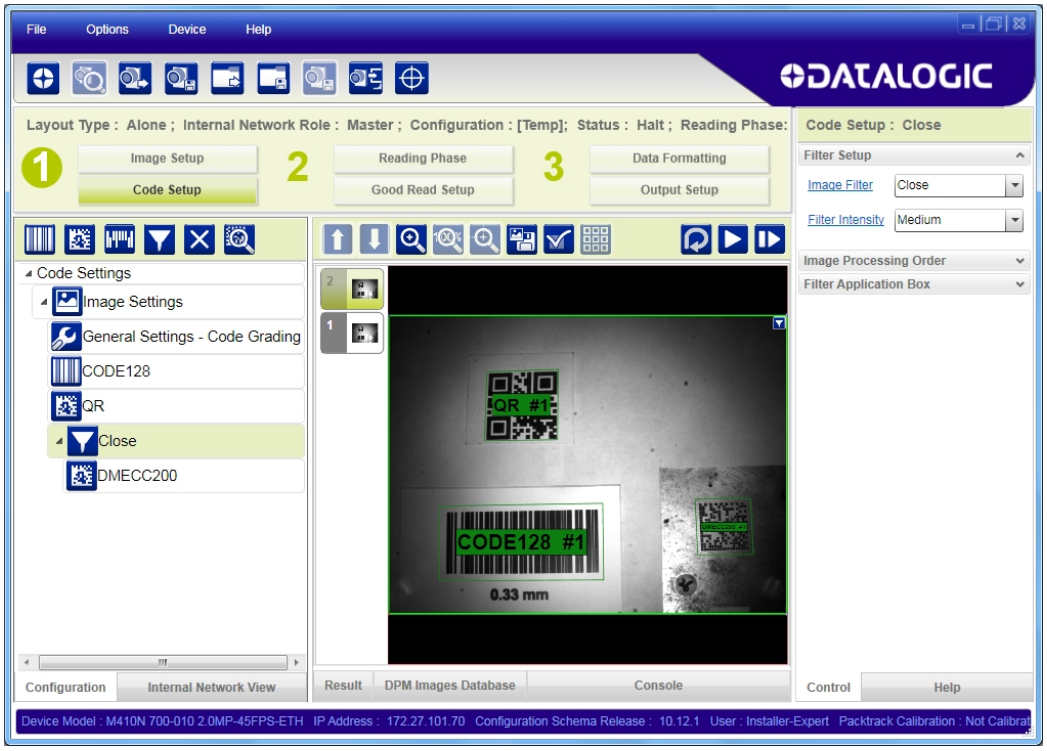


The Data Matrix DPM code is now successfully decoded. The Code Localization box and the Code Bounding box are both colored green.

Example 2

This example is based on the previous one except that each of the three code symbologies will always be located in a specific physical area of the image and must not be decoded out of its expected position.

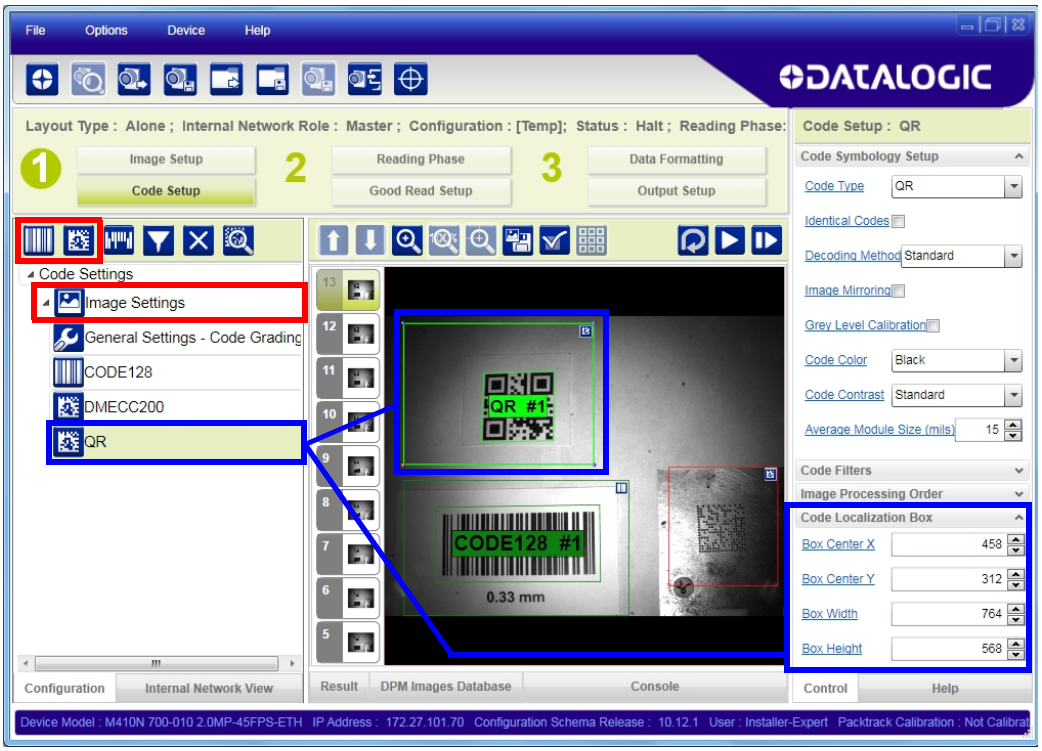
In this case, not only do we need to apply an Image Filter to the Data Matrix ECC 200 DPM code, but we need to set the Code Localization and Filter Application box sizes and positions to their respective image area to prevent decoding a code if it is in the wrong position.



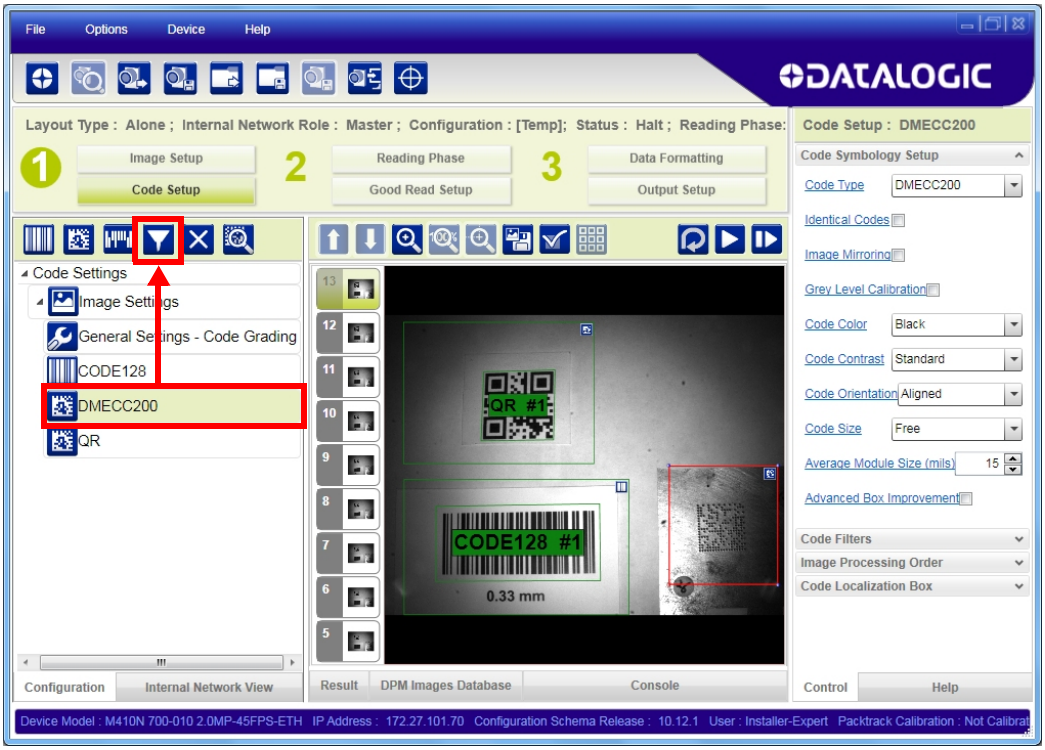
The following steps are taken to apply the Close filter to Example 2:

1. From the Image Settings group, add the Code Symbologies that must be decoded: Code 128, DMECC200 and QR Code.

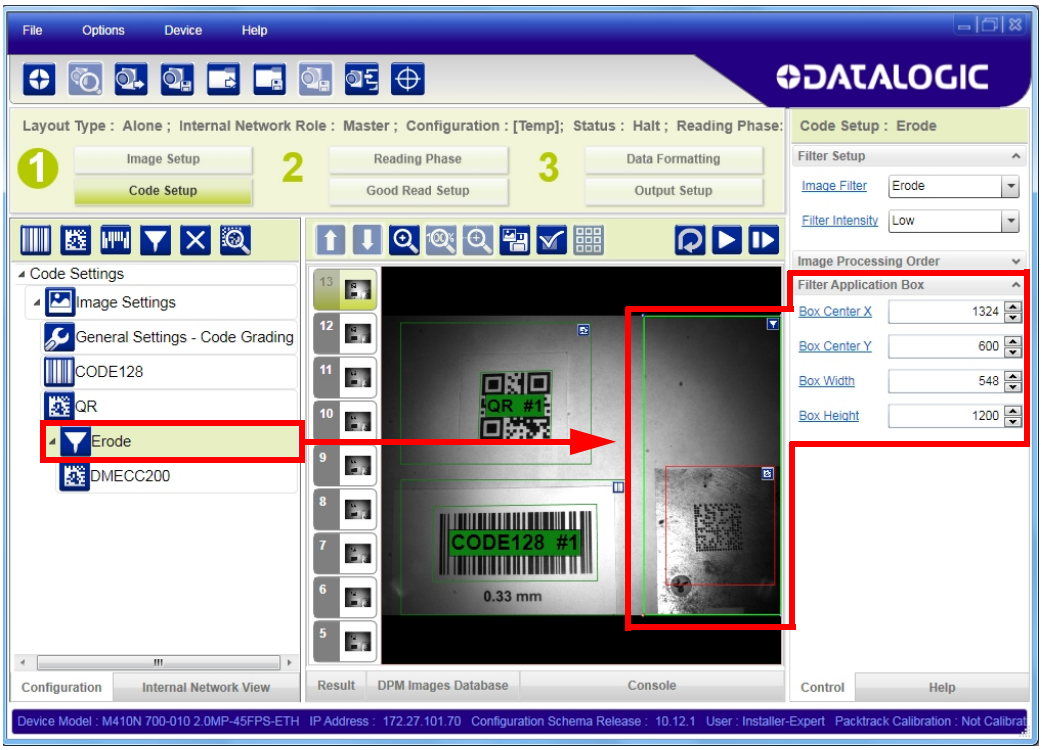
2. For each one of them, resize and position its code localization box (dragging its borders with the mouse or editing the Code Localization Box parameters) to the image area where the code will be found.



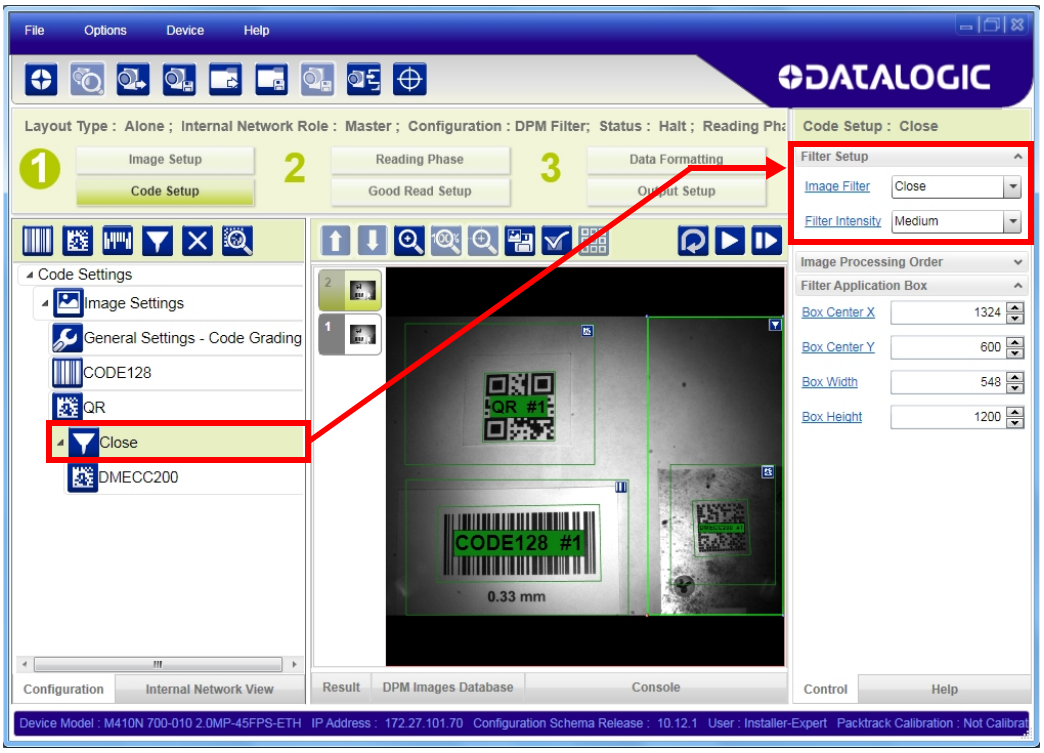
3. Select the Data Matrix ECC 200 code and add the Image Filter.



4. Resize and position the filter application box (dragging its borders with the mouse or editing the Filter Application Box parameters) to the image area where the filter will be applied. The relative Code Localization box must fit inside its Filter Application box.



5. Select the desired Image Filter Type and Intensity for correct decoding.



The Data Matrix DPM code is now successfully decoded only within the expected position. The Code Localization box and the Code Bounding box are both colored green.

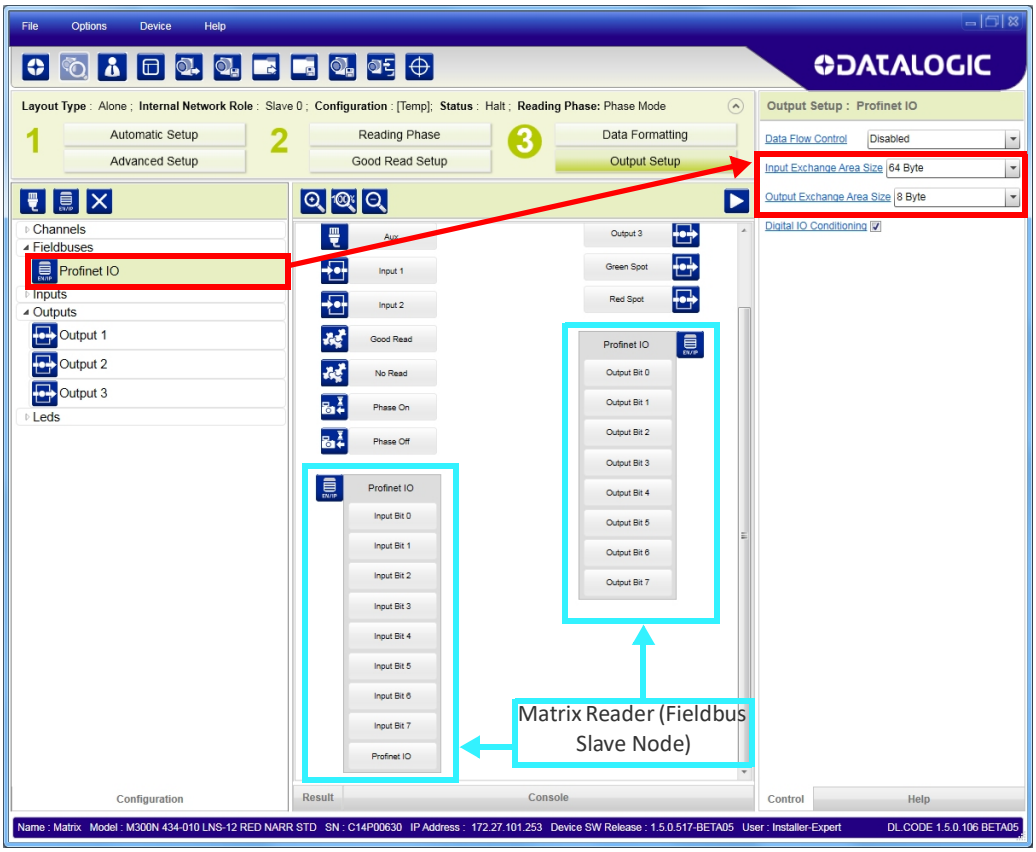
MATRIX CONTROL BY FIELDBUS CHANNEL

The Matrix reader can be controlled by signals coming from the Fieldbus Master as well as echoing its input signals to the Fieldbus Master.

Fieldbus Input/Output Representation in DL.CODE

For HMS Fieldbus and the embedded Profinet-IO channels, communication with the Matrix reader takes place through Input/Output Exchange Areas. The size of these areas must be correctly defined in the relative parameters (see below). See also "Digital IO Conditioning" on page 61 to account for Digital IO Conditioning.

The Input and Output Exchange Area Size parameters refer to the Fieldbus Master, Input **to** the Master, Output **from** the Master.



NOTE

All other representations in DL.CODE show the Fieldbus input and output channels from the Matrix reader perspective (Fieldbus Slave Node). Therefore, Fieldbus slave node Input Bits are from the Fieldbus Master and Fieldbus slave node Output Bits are to the Fieldbus Master.

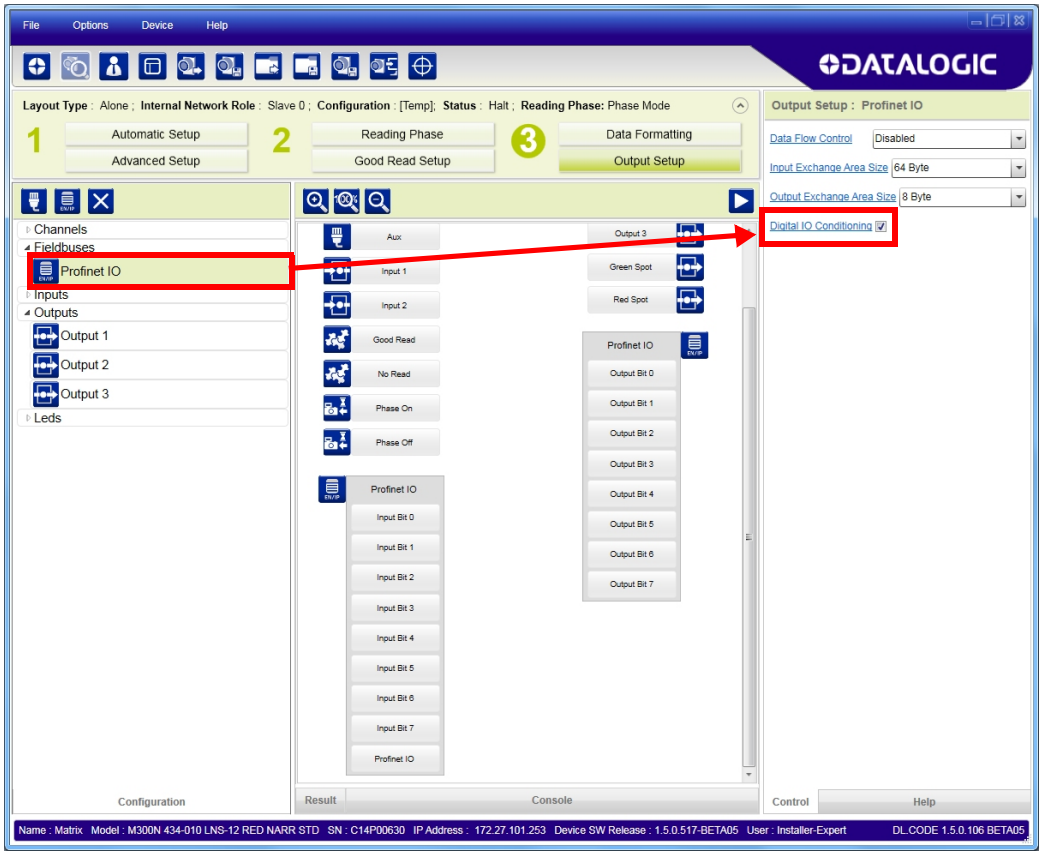
NOTE

When Profinet IO is enabled, the device communication is limited to the local network even if a Gateway is set.

Digital IO Conditioning

When checked (enabled by default), this parameter reserves the first byte of the Input/Output Areas for the Host to receive device Input echoes, drive the device Reading Phase and/or drive the device Outputs.

In this case application data or Data Flow Control begin at the second byte of the Input/Output Areas. For more details on Data Flow Control refer to the "DAD Driver" document included in the DL.CODE installation package.

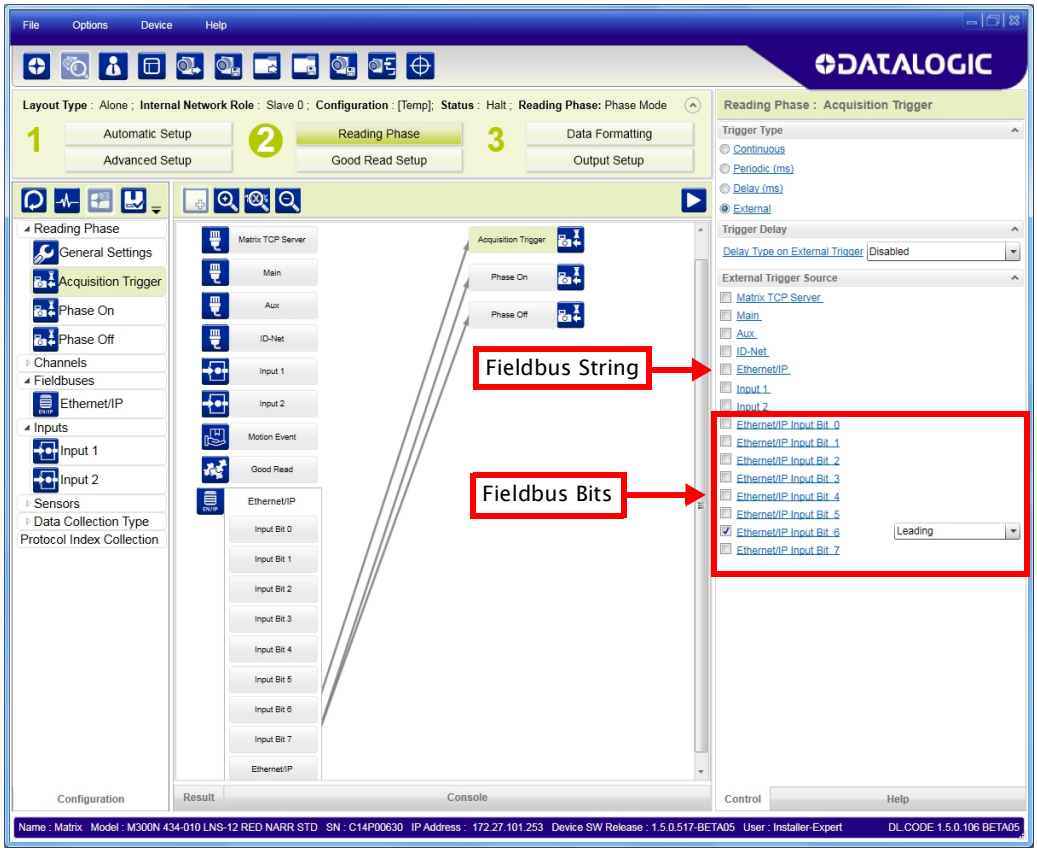


NOTE

By disabling Digital IO Conditioning, application data or Data Flow Control begin at the first byte of the Input/Output Areas. Any Digital IO Conditioning configurations (Input echo, Reading Phase or Output control from the Fieldbus Host) will be ignored.

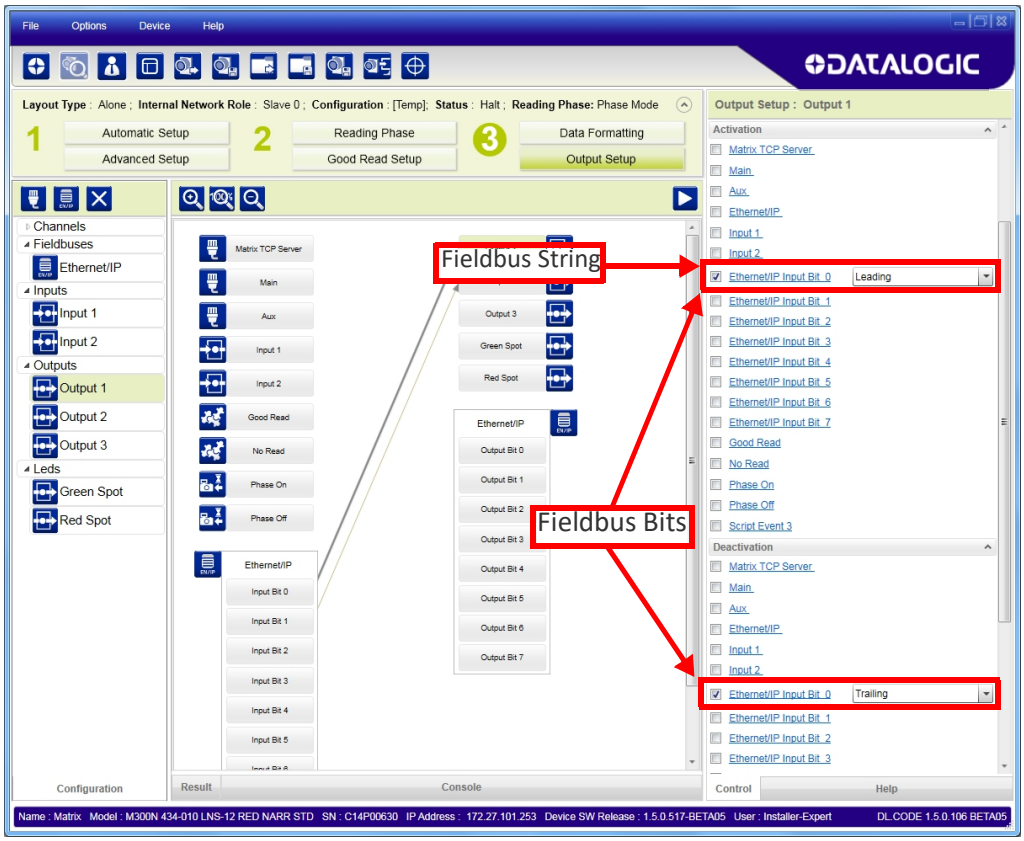
Fieldbus Reading Phase Control

The Fieldbus Master can control the reading phase by assigning either communication strings or individual communication bits to reading phase parameters. These bits are received on the Matrix Fieldbus channel as Input Bits.



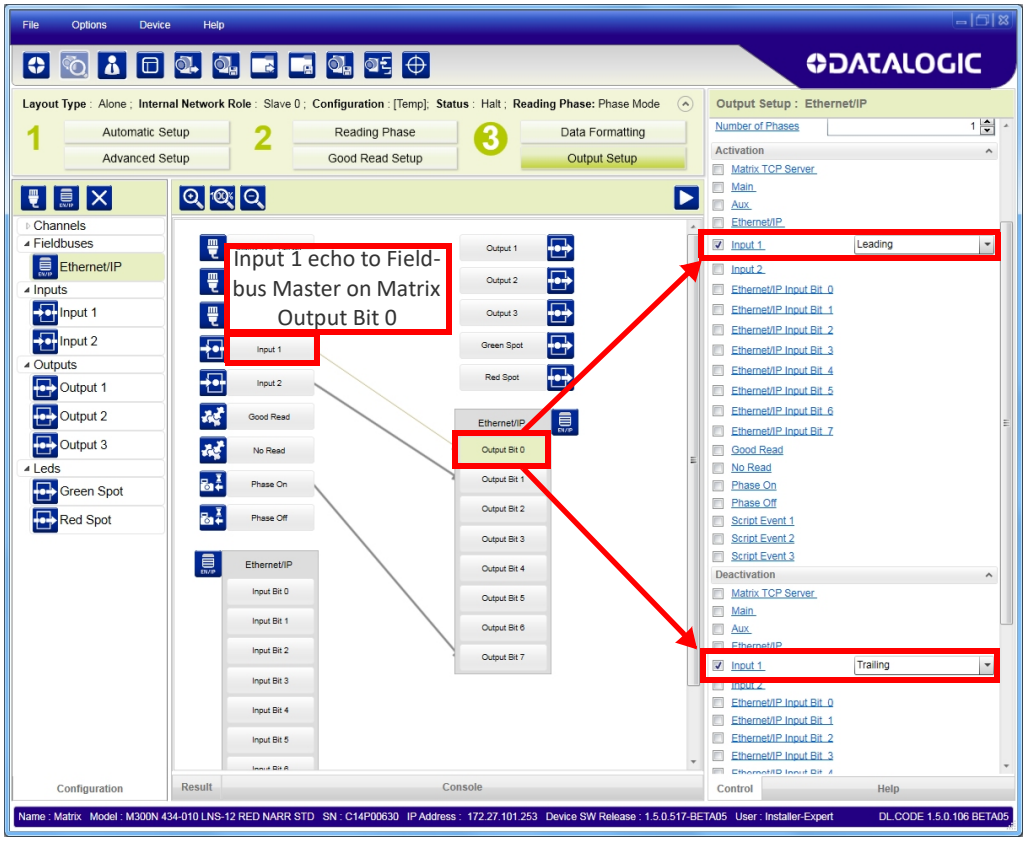
Fieldbus Digital Output Control

The Fieldbus Master can also drive the Matrix reader's Digital Outputs by assigning either communication strings or individual communication bits to the Digital Output Activation and Deactivation parameters. These bits are received on the Matrix Fieldbus channel as Input Bits.



Digital Input Echo to Fieldbus

The Fieldbus Master can receive the Matrix Reading Phase and Input signal echoes by assigning them as sources to the Fieldbus individual communication Output Bit Activation and Deactivation parameters. These bits are sent on the Matrix Fieldbus channel as Output Bits.



BACKUP AND RESTORE THROUGH DL.CODE

DL.CODE allows Backup and Restore to be performed to the configuration PC via file or to an external storage device such as BM100.

It can be performed for Single Reader and Internal Network (Master/Slave) configurations.

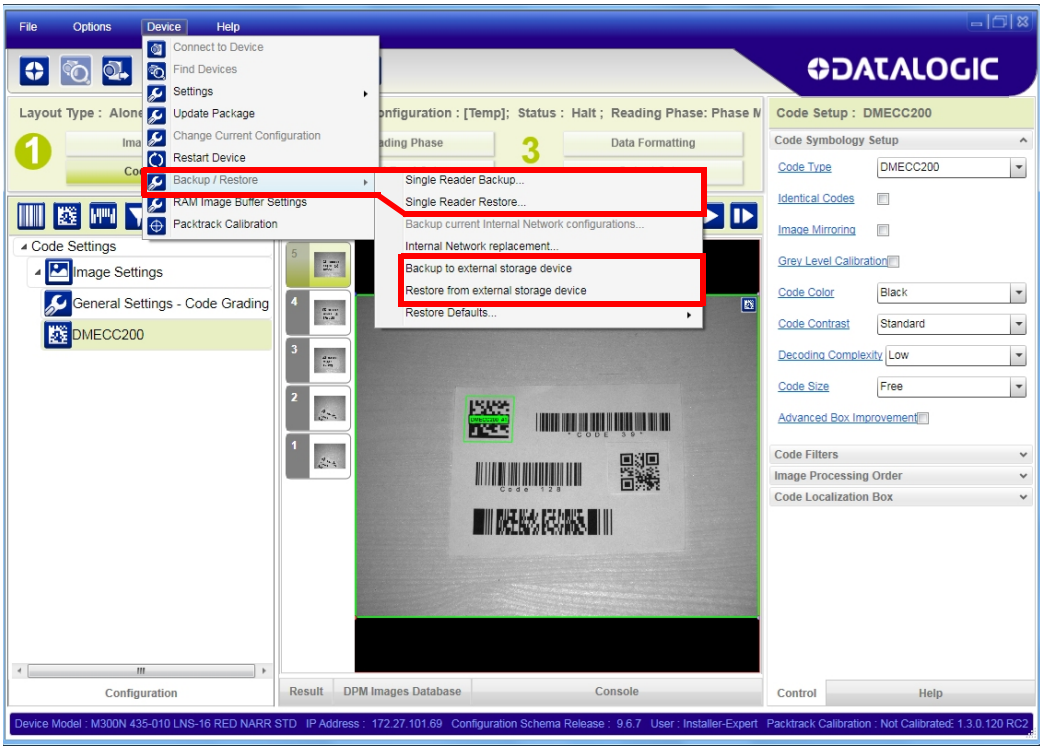
Backup and Restore functions allow performing Complete Configuration and Environment parameter storage for Single Reader and ID-NET (Master/Slave) network devices as well as device firmware. Backup and Restore can be applied to any reader connected through a device having external backup memory, regardless of the reader's network configuration.

Backup to and Restore from external device is supported by DL.CODE for all reading devices when connected to:

- CBX + BM100
- QLM-Series Gateways

NOTE

Before executing a Backup to a BM100 backup module make sure the Write Protection switch is set to Unlocked.



Backup

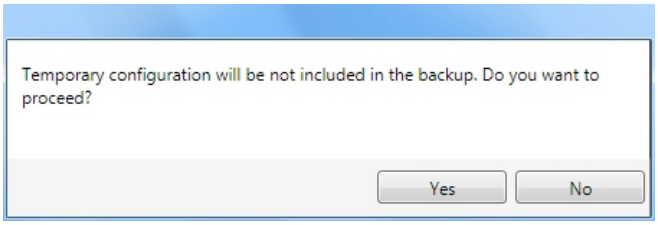
To perform a **Backup**:

1. From the DL.CODE Device menu, select either **Single Reader Backup** (to file on PC) or **Backup to external storage device**.

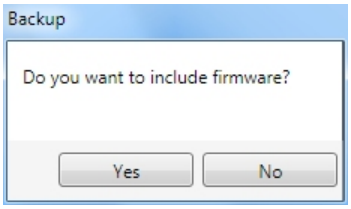
NOTE

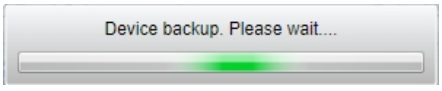
For ID-NET network Backup, select the Backup current Internal Network configurations selection.

You will be reminded that configuration in temporary memory will no be saved so you should save the configuration to the reader before performing Backup:

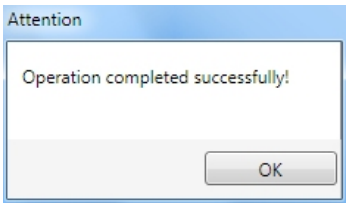


If you are performing a backup to a file, you will be asked whether to include the firmware or not.





At the end of the backup, DL.CODE shows a message indicating successful completion.



Restore

CAUTION

The restore procedure cannot be performed on different models, but only on one same model.

To perform a **Restore**:

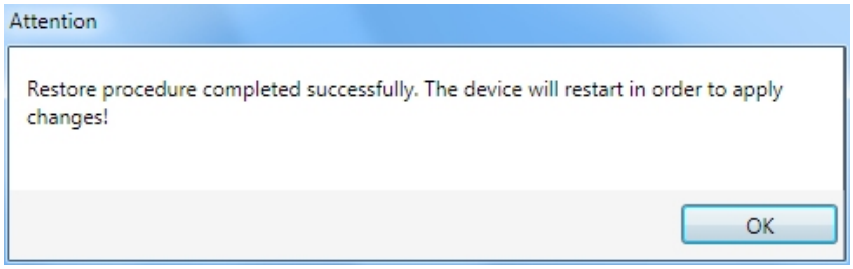
1. From the DL.CODE Device menu, select either **Single Reader Restore** (from file on PC) or **Restore from external storage device**.

NOTE

For ID-NET network Restore, select the Internal Network replacement selection.

If restoring an ID-NET network though the Master, this may take a few minutes.

At the end of the restore, DL.CODE shows a message indicating successful completion.



Replacement

CAUTION

The replacement device must be the exact same model as the device it is replacing.

The **Restore** function also provides easy and secure Single Device Replacement:

1. Remove the device to be replaced.
2. Connect the new device (make sure the new device has been previously set to factory default).
3. Run the Restore procedure by selecting either **Single Reader Restore** (from file on PC) or **Restore from external storage device** item (see: Restore procedure).

NOTE

In case of Backup or Restore operation failures, error messages will be displayed in the Monitor Diagnostic page.

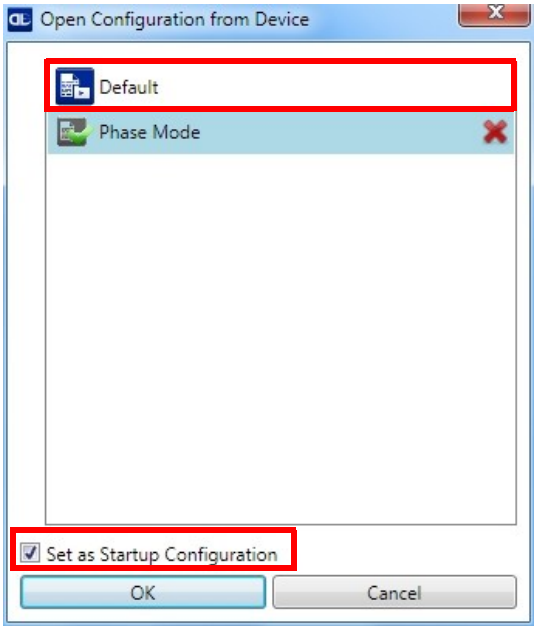
RESTORE DEFAULTS

The device parameters are divided into two main classes, Configuration and Environment which are affected differently by the Restore Defaults commands.

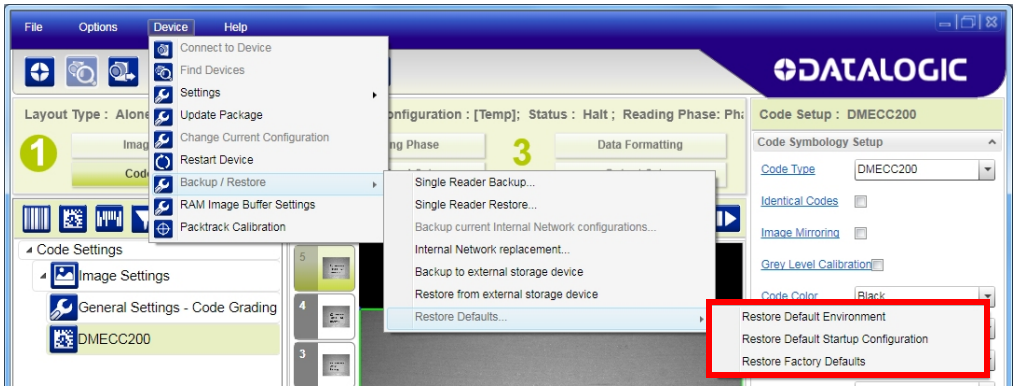
- The Configuration parameters are the ones set in the various steps of the configuration process and are specific to each application. When multiple configurations (jobs) are saved on a single device, these parameters can be different from one configuration to the next.
- Environment parameters regard the device Identity and Position in a Network (Ethernet, ID-NET, etc.) and are not influenced by the Default (or any other) Configuration present in memory.

Restore Default Startup Configuration

The Default configuration is always present on the reader and in fact it is not modifiable and cannot be deleted. It can always be restored by simply selecting it from the Open from Device configuration list.



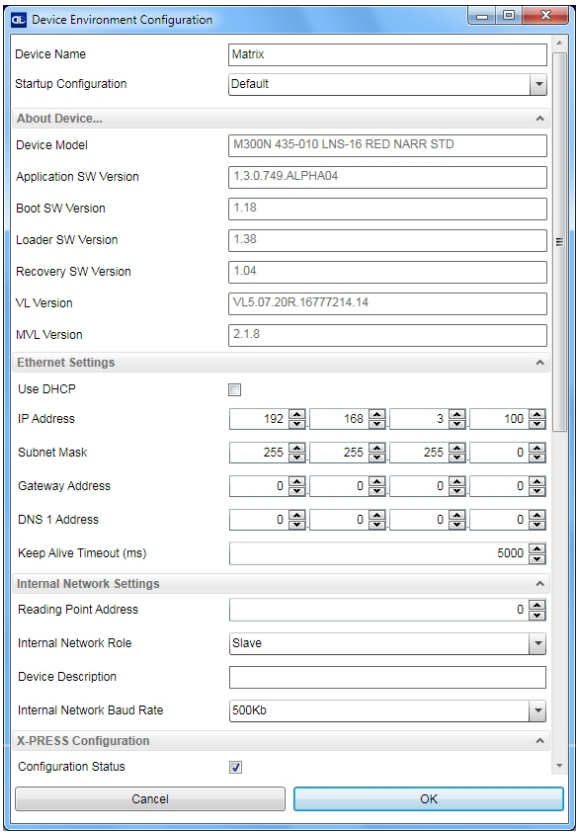
The same action can be performed from the Device menu >Backup/Restore > Restore Defaults > **Restore Default Startup Configuration**. The Default Configuration will be set to run at startup and the reader will be reset.



Any previously saved configurations on the device will remain in memory, but the Default configuration is set as the startup configuration.

Restore Default Environment

Restore Default Environment returns all Environment parameters to their factory default settings. The default IP address will be restored as well as all the parameters managed in the Device Environment Configuration window.



The Factory Default static IP address for all Matrix N Family readers is:

IP Address = 192.168.3.100

Any previously saved configurations on the device will remain in memory, but the Default configuration is set as the startup configuration.

Restore Factory Defaults

In order to return a device to its absolute Factory default parameters (for example device replacement) it is necessary to use the **Restore Factory Defaults** command. You will be prompted to confirm.

All Environment parameters will be restored to Factory default values **and any existing configurations stored on the device will be erased**. The device will be reset and therefore start in run mode with the factory default configuration.

Software Reset

At any time the device can be reset by the Restart Device command in the DL.CODE Device Menu.

CAUTION

Remember to save the current configuration before restarting.

WEB MONITOR

The Web Monitor is a remote monitoring tool provided to visualize the Matrix reader in its run-time environment. You can access it from the DL.CODE Task area or directly from your browser by inputting the IP address of the reader. The reader must be available on the LAN.

This works on major browsers (also on smartphones and tablets), which support HTML5 (see the table below).

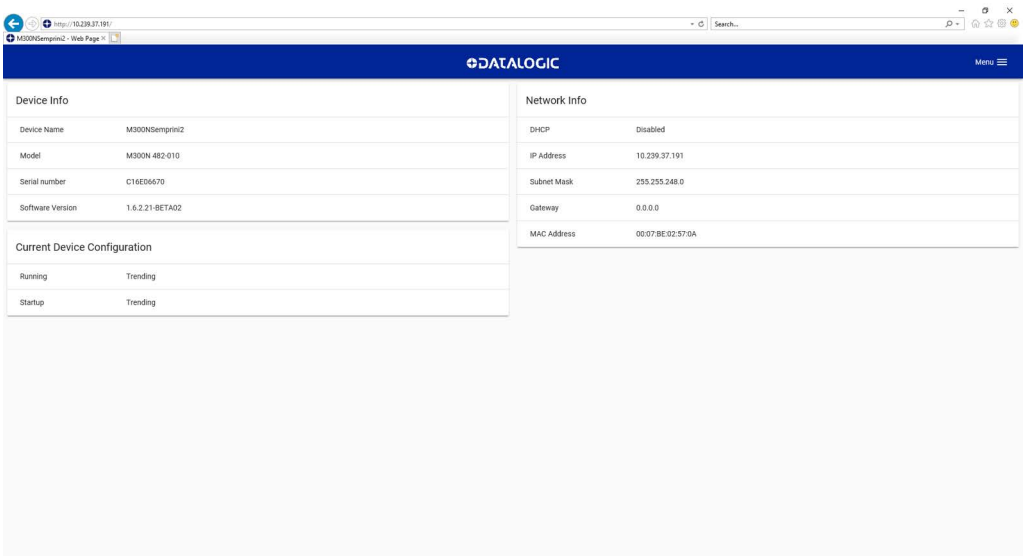
Chrome	Firefox	Edge	IE	Safari	iOS	Android	IE Mobile
latest	latest	14	11	10	10	Nougat (7.0) Marshmallow (6.0)	11
		13	10	9	9	Lollipop (5.0, 5.1)	
			9	8	8	KitKat (4.4.)	
				7	7	Jelly Bean (4.1, 4.2, 4.3)	

NOTE

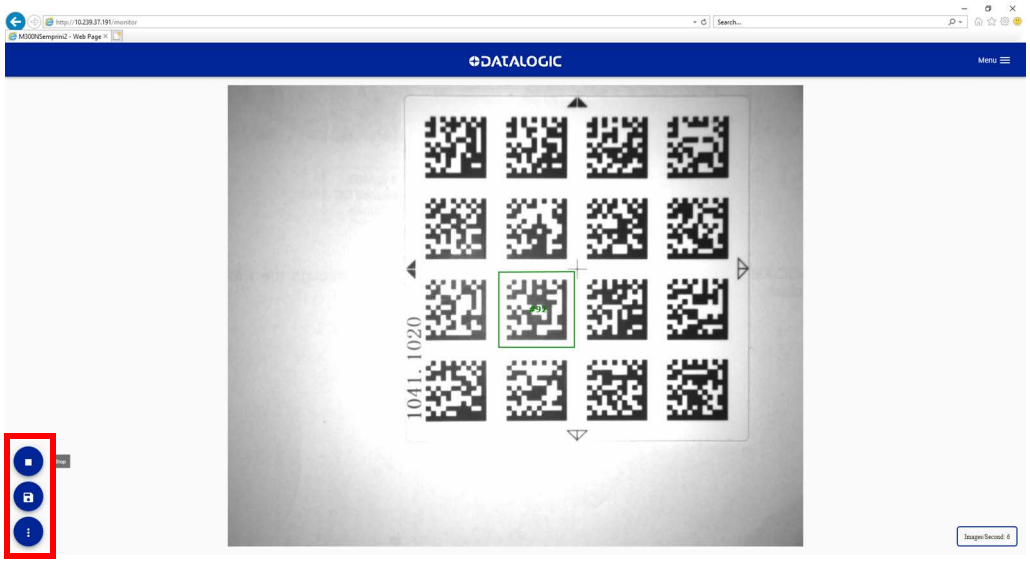
If you use Web Monitor with a simulator reader, make sure that port 80 of your computer is free. If port 80 is busy, you will not be able to access the Web Monitor page of the selected simulator.

The Web Monitor tool provides the following features:

Info page

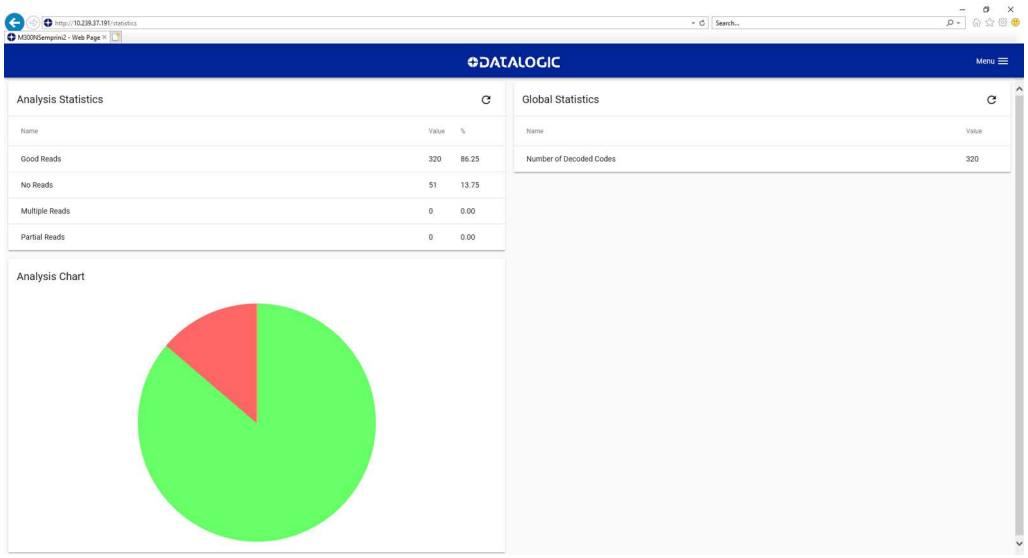


Monitoring Page

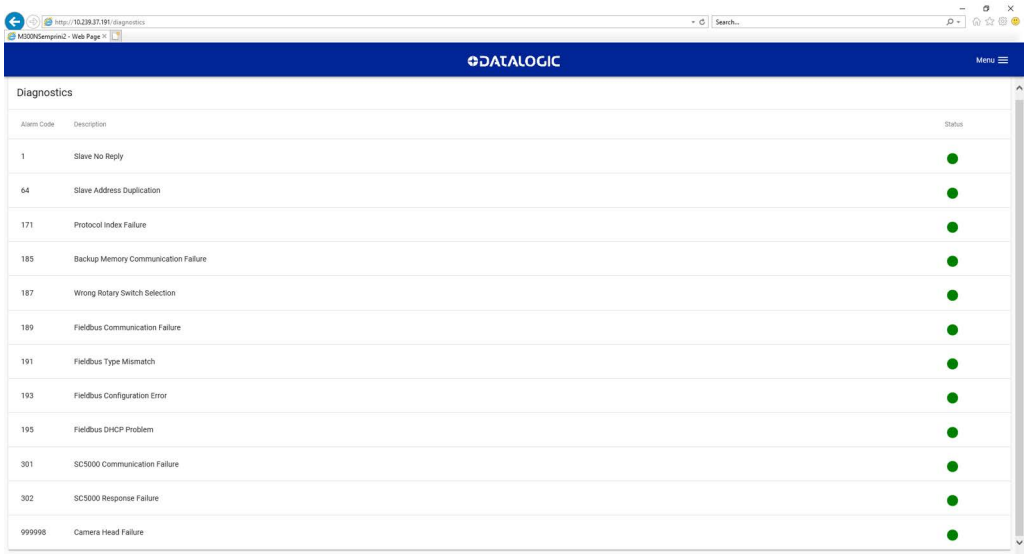


	The Stop/Play button allows you to stop image monitoring to save a particular image. The reader continues to run, only the monitored image is stopped.
	The Save image button automatically downloads the current image as a jpg file to the browser's default Download folder. The default naming syntax is image_*weekday month day year*.jpg (i.e. image_Wed Jan 23 2019.jpg)
	The options button opens an options panel on the Monitoring window which allows: - setting the quality of the image to monitor (higher quality images reduce framerate) - showing only positive results (good read), only negative results, or all results - showing Code List - showing Web Monitor images per second - showing x,y Coordinates to the Monitoring window. The coordinates report the mouse position over the monitored image (in pixels). To close the options panel, click on the image area.

Statistics Page



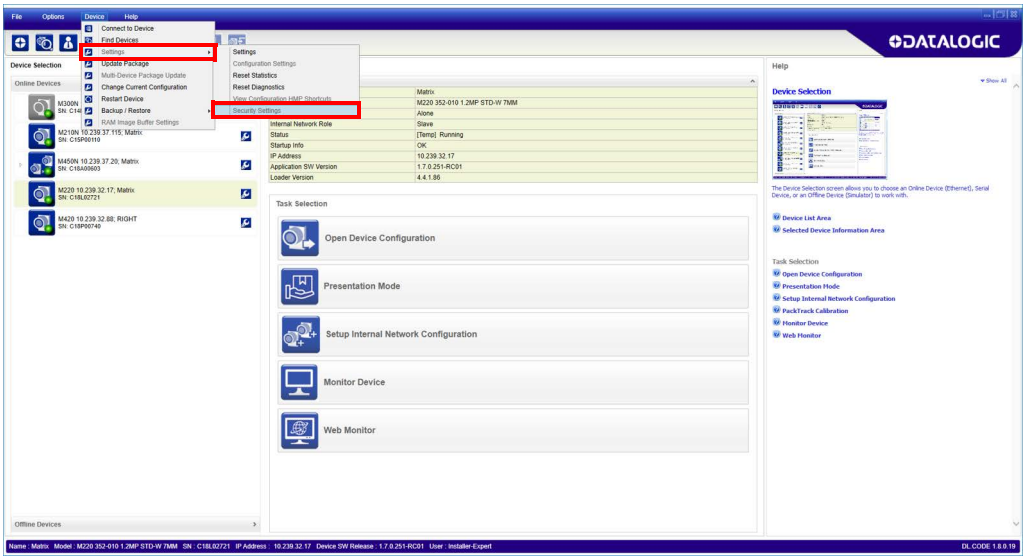
Diagnostics Page



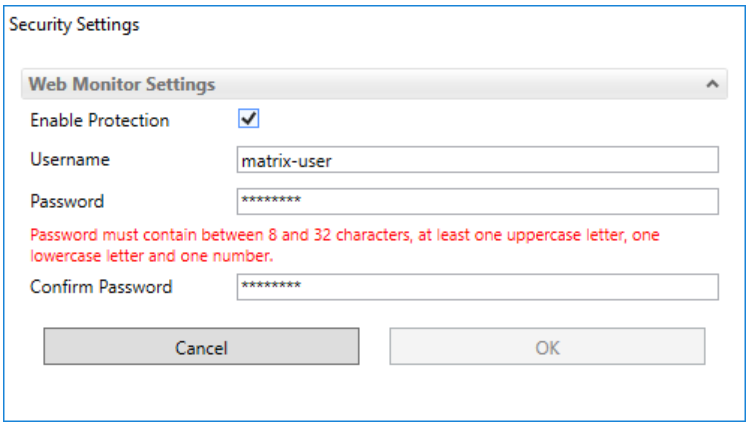
The Web Monitor tool also provides a page for **Code Grading Monitoring** and a **Configurator** to edit your Matrix reader settings. To access these features, a user name and password are required.

Please follow these steps:

1. On DL.CODE, go to Device>Settings>Security Settings.



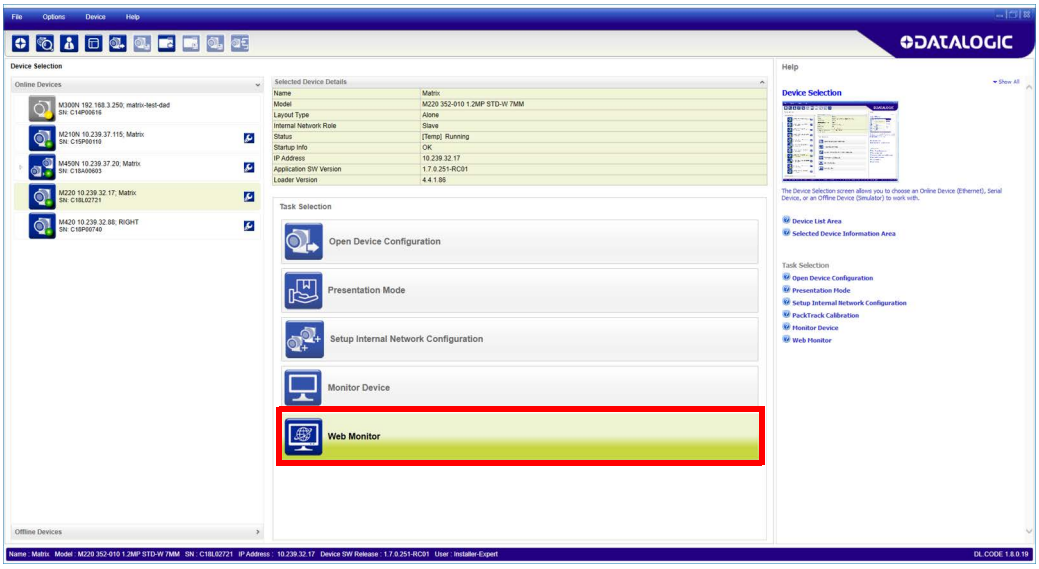
2. Set your password under the Web Monitor Settings tab.



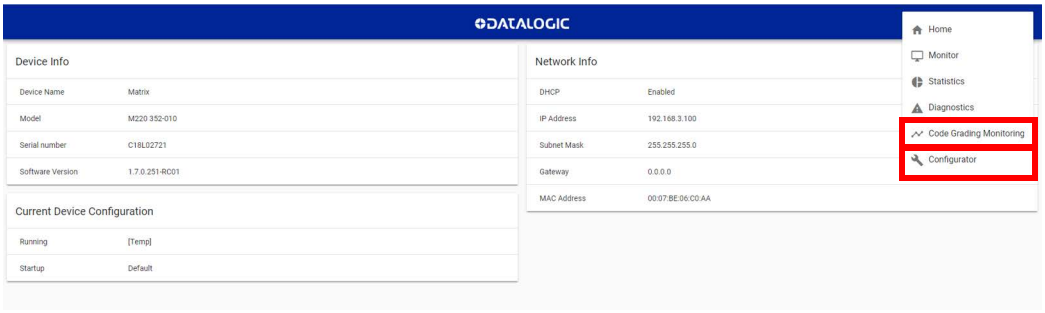
NOTE

A default user name is already set, while no default password is provided for security reasons.

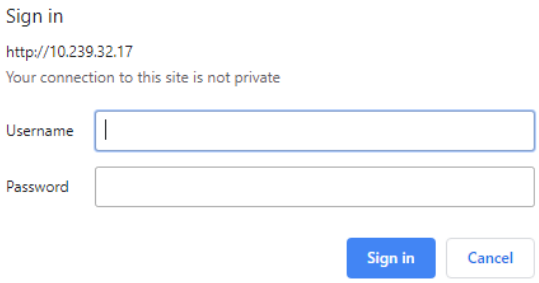
3. Open the Web Monitor by selecting it on the Task Area.



4. On the Web Monitor tool, open the Menu on the upper right-hand corner and select either Code Grading Monitoring or Configurator.



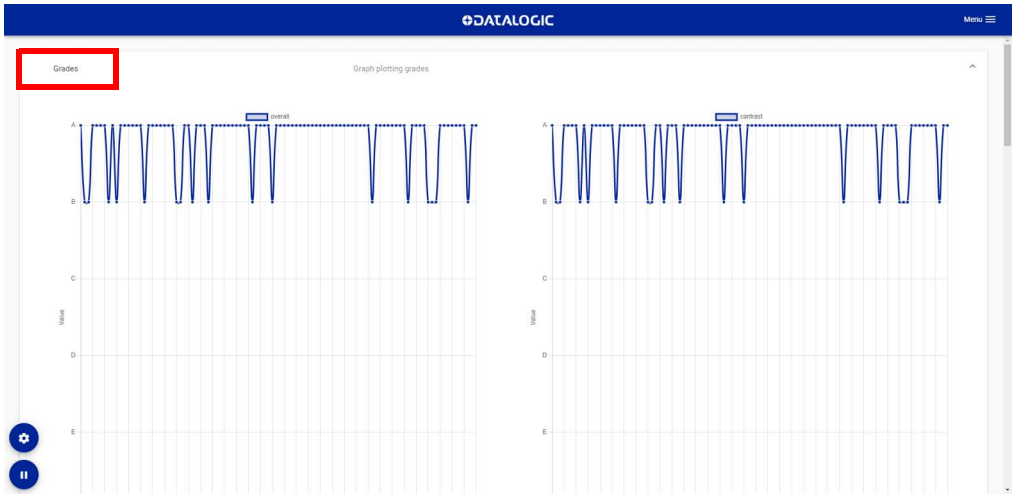
5. A pop-up window will be displayed prompting the user to enter user name and password.



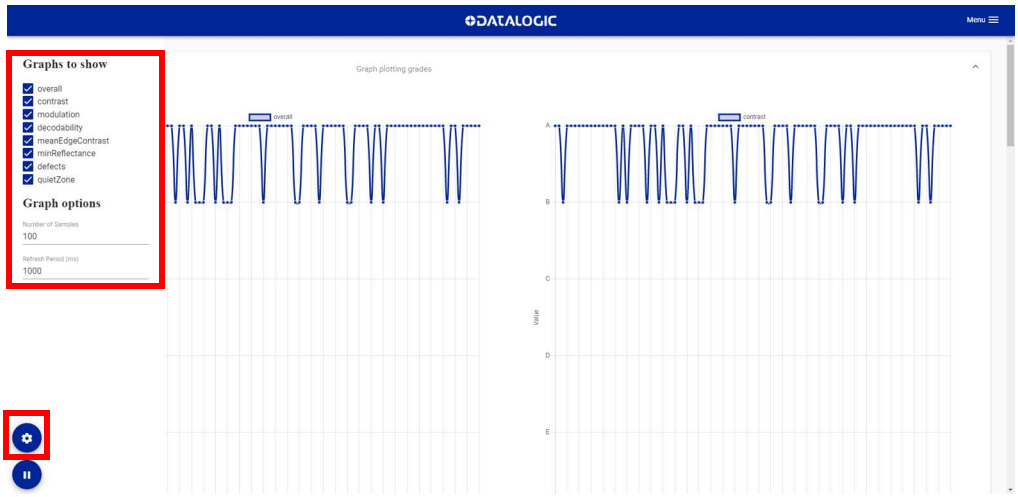
Code Grading Monitoring Page

To view Code Grading live results, the user first needs to enable code grading on DL.CODE (see Chapter 13, Code Grading).

On the Code Grading Monitoring Page the user can see live results in graphs. Results can either be viewed as Grades or Values:

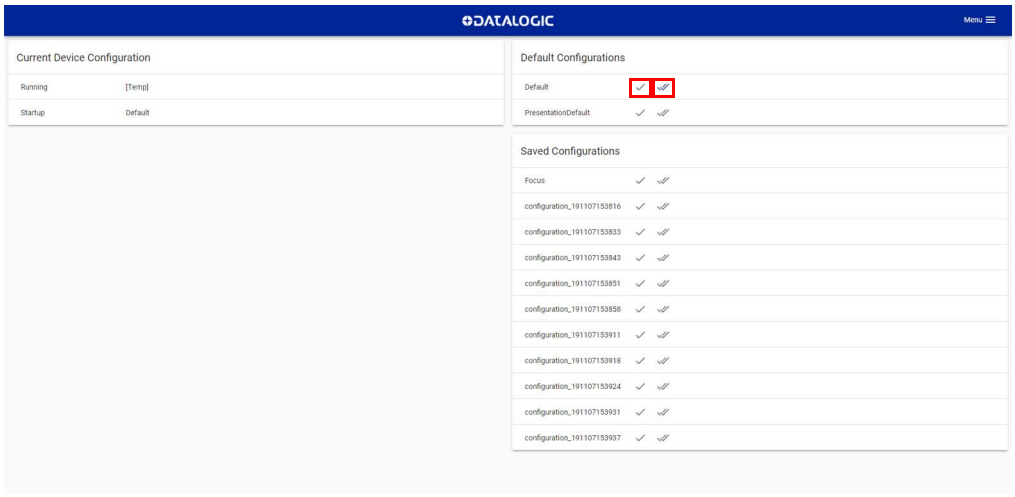


Each graph represents results for a code grading parameter. The graphs to display can be selected from the Settings icon:



Here the user can also set the number of samples and the refresh period (ms).

Configurator page



On the Configurator page the user can view the configuration currently running on the device and run another one by selecting it from the saved configurations, in particular:

- click on the single check to load the configuration;
- click on the double check to load the configuration and set it as the startup configuration.

The loaded configuration will display a blue single check. The startup configuration will display a blue double check.

CHAPTER 5
DATA COLLECTION METHODS

CODE COLLECTION

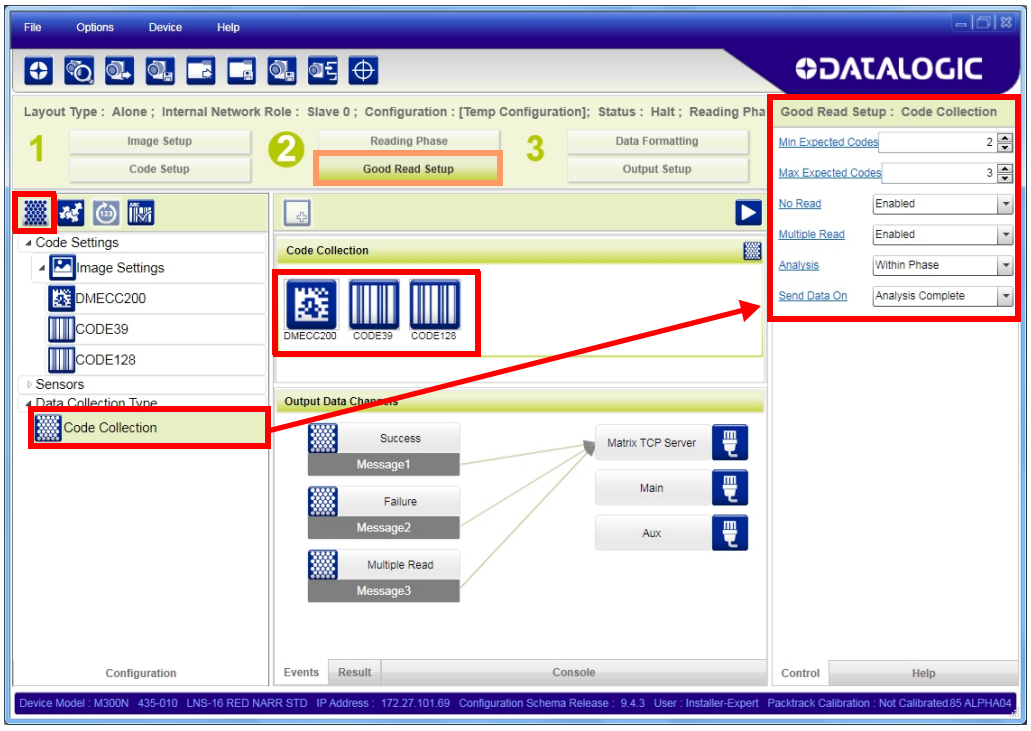
Valid Operating Modes: Continuous, One Shot, Phase Mode, PackTrack

In Code Collection mode the reader can collect several codes before providing a single output result.

The number of codes to be collected is set by the *Min Expected Codes* and *Max Expected Codes*.

The type of output message sent to the host depends on the *No Read* and *Multiple Read* parameter settings and can be modified and formatted in the Data Formatting Output Group.

Example:



As shown in the figure above, the following parameters are set:

The DMECC200, CODE39 and CODE128 code types are enabled and present in the Code Collection.

Min Expected Codes = 2; Max Expected Codes = 3; No Read is Enabled; Multiple Read is Enabled

Case 1:two or three codes (of the enabled code types) are read

Output:the content of **Message 1 Success** is sent to the Host. This corresponds to a Good Read message and by default contains the code content.

Case 2:one code is read

Output:the content of **Message 2 Failure** is sent to the Host. This corresponds to the No Read message because it doesn't satisfy the minimum expected codes criteria.

Case 3:four codes (of the enabled code types) are read

Output:the content of **Message 3 Multiple Read** is sent to the Host. This corresponds to the Multiple Read message because it exceeds the maximum expected codes criteria.

If the Multiple Read parameter is Disabled, then Case 3 is considered a Good Read and the content of **Message 1 Success** is sent to the Host containing the first three decoded codes.

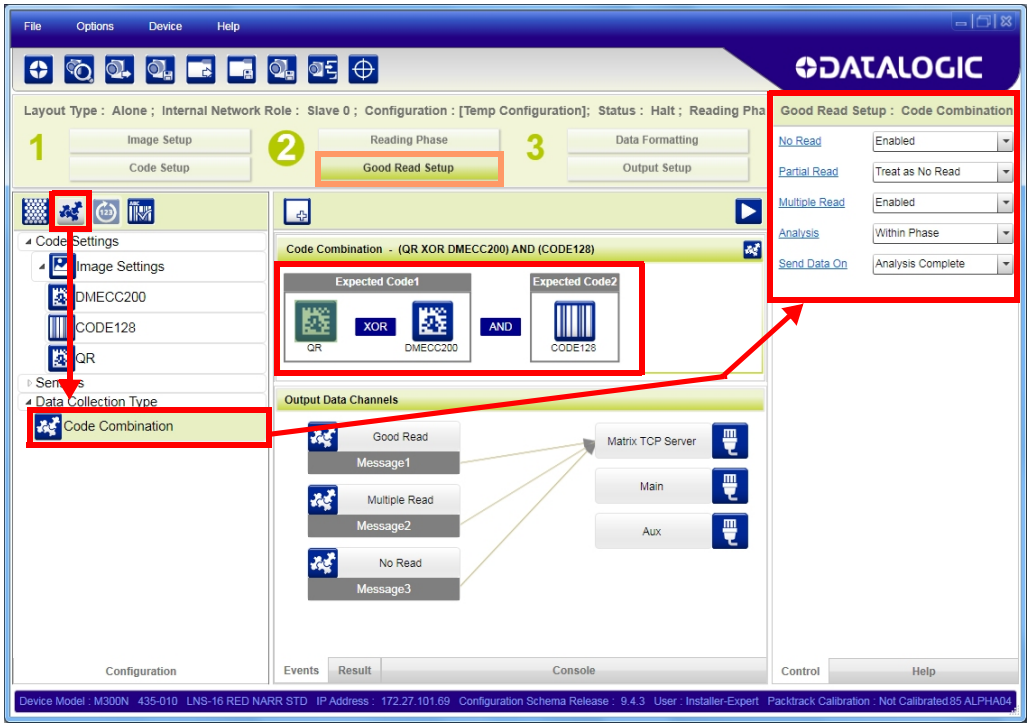
In Phase Mode, by setting ***Analysis*** to *Within Phase* only the reading phase is considered and a single result is provided to the user for each phase. On the other hand, by selecting *Within An Image* the reader searches for the defined number of codes within each acquired image and a result is provided to the user after each image decoding.

CODE COMBINATION

Valid Operating Modes: Continuous, One Shot, Phase Mode, PackTrack

In Code Combination mode the output results sent to the Host depend not only on the codes read but on meeting their configured logical combination criteria.

Example:



As shown in the figure above, the following parameters are set:

The DMECC200, QR Code and CODE128 types are enabled and present in the Code Combination with the following logical combination:

DMECC200 OR QRCode AND Code128.

No Read is Enabled; Multiple Read is Enabled; Partial Read is treated as No Read

Case 1:codes (DMECC200 AND Code128) or (QR Code AND Code128) are read

Output:the content of **Message 1 Good Read** is sent to the Host. This message by default contains the code content.

Case 2:only one of the three codes (DMECC200, QR Code or Code128) is read

Output:this is a Partial Read which, in this example, is treated as a No Read. The content of **Message 2 No Read** is sent to the Host.

Case 3:all three codes (DMECC200, QR Code and Code128) or (only DMECC200 and QR Code) are read

Output:the content of **Message 3 Multiple Read** is sent to the Host.

If the Multiple Read parameter is Disabled, then for Case 3:

- all three codes read (DMECC200, QR Code and Code128), is considered a Good Read and the content of **Message 1 Good Read** is sent to the Host containing one of the 2D codes (whichever was decoded first) AND Code128.
- reading (only DMECC200 and QR Code), is considered a Partial Read which, in this example, is treated as a No Read and the content of **Message 2 No Read** is sent to the Host.

In Phase Mode, by setting ***Analysis*** to *Within Phase* only the reading phase is considered and a single result is provided to the user for each phase. On the other hand, by selecting *Within An Image* the reader searches for the defined combination of codes within each acquired image and a result is provided to the user after each image decoding.

NOTE

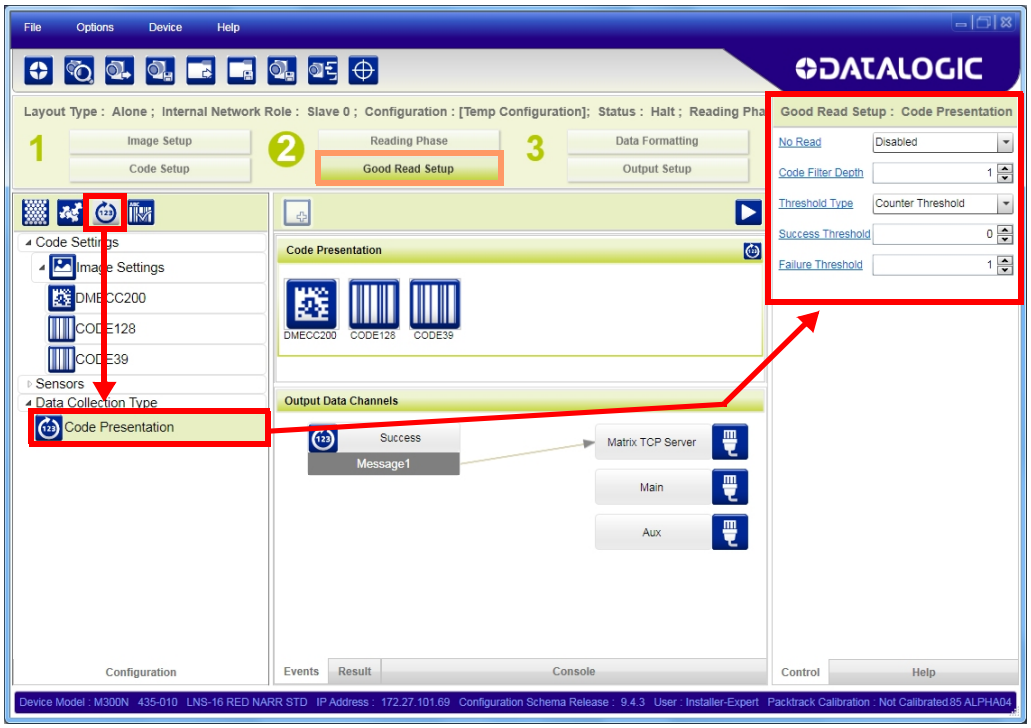
In Code Combination mode, the configured expression can contain up to a maximum of 50 codes.

CODE PRESENTATION

Valid Operating Modes: Continuous, One Shot, Phase Mode (Motion Sensor)

In Code Presentation mode typically a code is placed in front of the reader manually and the successful output results are sent to the Host. Usually No Reads are disabled and the configuration should correctly manage Multiple Reads.

Example:



As shown in the figure above, the following parameters are set:

The DMECC200, CODE128 and CODE39 types are enabled and present in the Code Presentation:

No Read is Disabled; Code Filter Depth is 1, and multiple reads are handled by the Acquisition Counter Threshold: Success Threshold=0, Failure Threshold=1.

Case 1:any code or combination of codes (of the enabled code types) is read in a single acquisition.

Output:the content of **Message 1 Success** is sent to the Host containing one occurrence of all of the decoded codes. This message by default contains the code content.

Case 2:any code or combination of codes is presented to the reader repeatedly

Output:this is a multiple read condition which is managed as follows:

Success Threshold=0, the same code (by content and type) cannot be read in successive acquisitions.

Failure Threshold=1, the same code (by content and type) can only be read after at least 1 acquisition without a code.

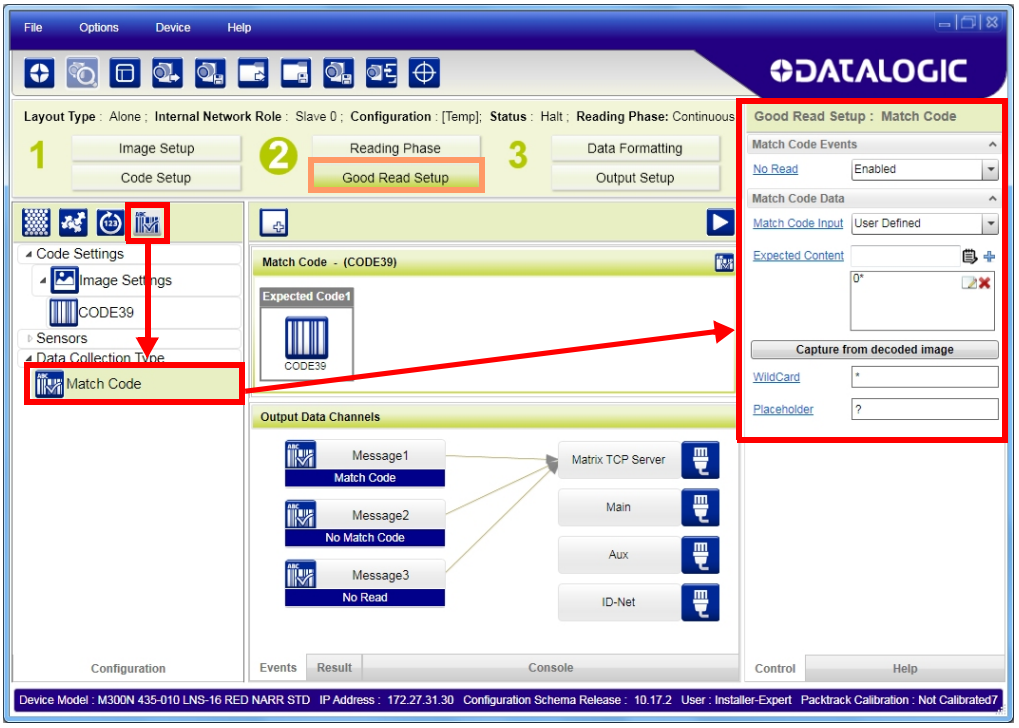
MATCH CODE

In Match Code mode the output results sent to the Host depend on whether the codes read meet the match code criteria or not.

User Defined

Valid Operating Modes: Continuous, One Shot, Phase Mode

Example:



As shown in the figure above, the following parameters are set:

CODE39 type is enabled and present in the Match Code:

No Read is Enabled; Match Code Input is User Defined where the Expected Content = “0*” (all codes must begin with zero but can be followed by any string – wildcard).

Case 1:a CODE39 type is read and its content begins with 0 (zero).

Output:the content of **Message 1 Match Code** is sent to the Host. This message by default contains the code content.

Case 2:a CODE39 type is read but its content does not begin with 0 (zero).

Output:the content of **Message 2 No Match Code** is sent to the Host.

Case 3:no CODE39 type is read.

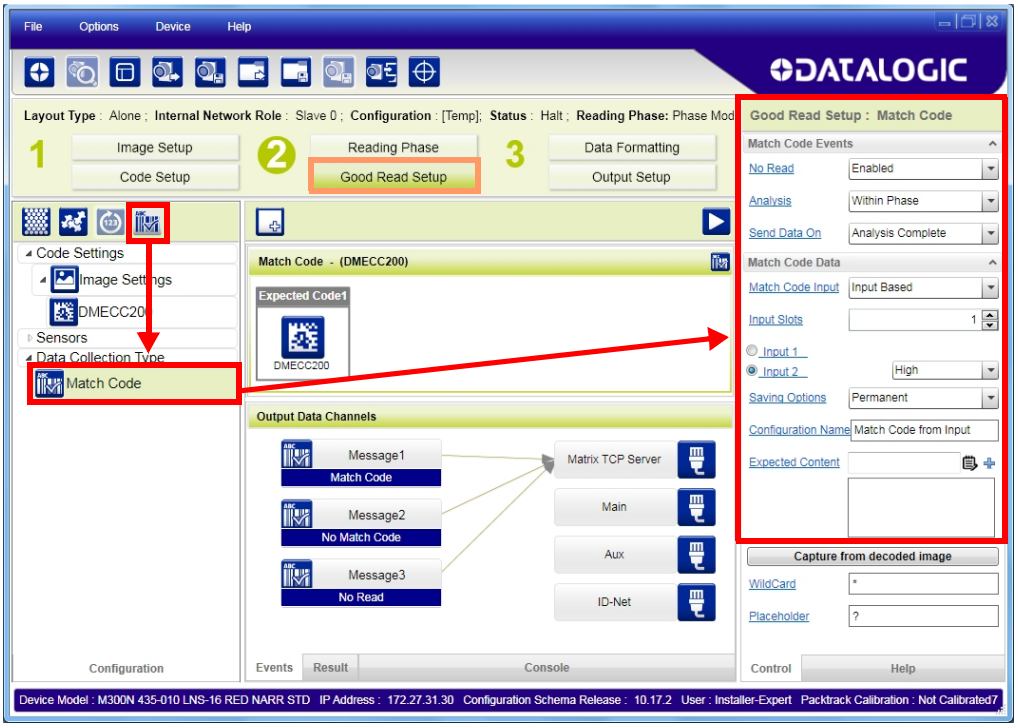
Output:the content of **Message 3 No Read** is sent to the Host.

Input Based

Valid Operating Modes: One Shot, Phase Mode

The reading phase is typically controlled by Input 1 (Trigger) while Input 2 is used for Match Code saving. An alternative is to control the reading Phase by sending strings from one of the available communication Channels.

Example:



As shown in the figure above, the following parameters are set:

Data Matrix ECC 200 type is enabled and present in the Match Code:

No Read is Enabled; Match Code Input is Input Based (captured by the Input 2 High state), Saved to Permanent memory with configuration job name = “Match Code from Input”.

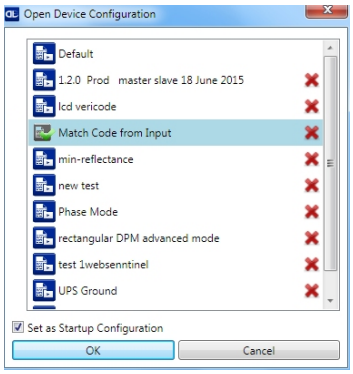
NOTE

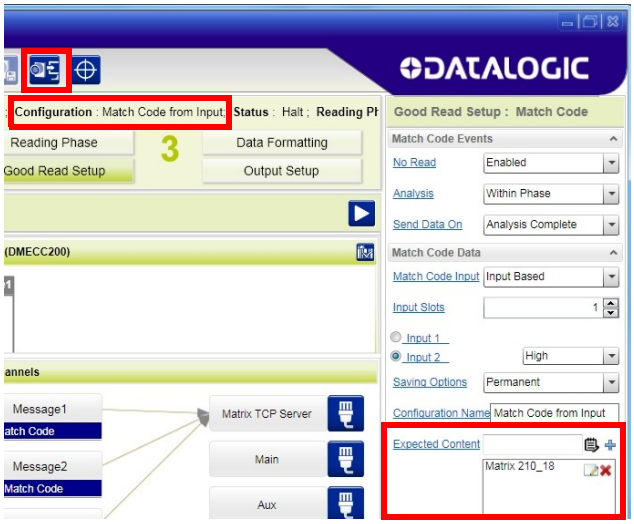
Capturing the Match Code on Input must be done either in Monitor or in RUN mode (disconnected from DL.CODE). You cannot capture the match code on input from the DL.CODE configuration environment.

Switch to Monitor and trigger a reading phase while Input 2 is in the High state (for this example).

The code will be read, saved in the Expected Content list and the entire job will be saved to the configuration name “Match Code from Input”.

Return to the Configuration Environment.





If you choose the Temporary Saving Option, you will need to save the job manually from the configuration environment as with all other configurations.

Case 1:a DMECC200 type is read and its content matches a code in the Expected Content list.

Output:the content of **Message 1 Match Code** is sent to the Host. This message by default contains the code content.

Case 2:a DMECC200 type is read but its content does not match a code in the Expected Content list.

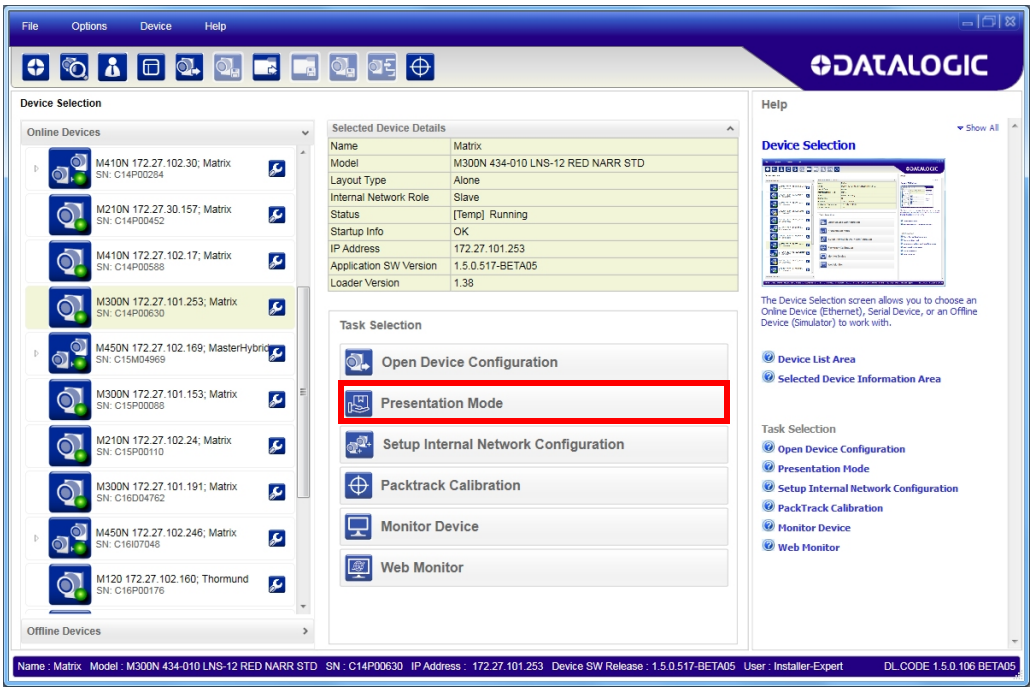
Output:the content of **Message 2 No Match Code** is sent to the Host.

Case 3:no DMECC200 type is read.

Output:the content of **Message 3 No Read** is sent to the Host.

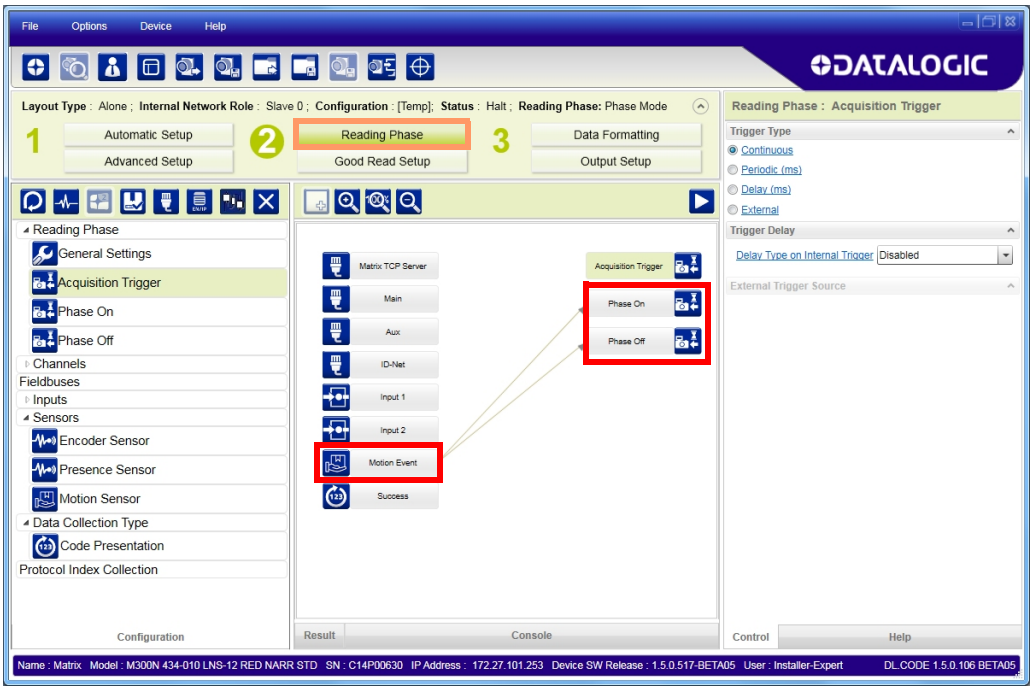
PRESENTATION MODE

Presentation Mode is a pre-configured case of the Phase Mode Operating Mode. It uses a software Motion Sensor to constantly search for changes in the images during the Idle state. When images change (indicating motion), the reading phase is activated to automatically detect any codes that will be presented to the device.



Selecting this task opens DL.CODE with the following default parameter settings:

- Phase Mode Operating Mode
- Phase On and Phase Off connected to Motion Event
- Data Collection Method is Code Presentation
- Code Filter Depth is 1
- Threshold Timeout 2 sec
- Data Matrix symbology enabled.
- Output Message to the Host is sent to the TCP Channel
- Green Spot is illuminated on successful read



This default is useful for kiosk or cash register applications where a single code is presented to the reader one-at-a-time by hand. The movement is typically < 0.5 m/s and the Focusing Pointer system is enabled during the Idle state to show where the code must be presented. The illuminator is off in the Idle state to avoid constant flashing. Either the code is returned or no message is returned.

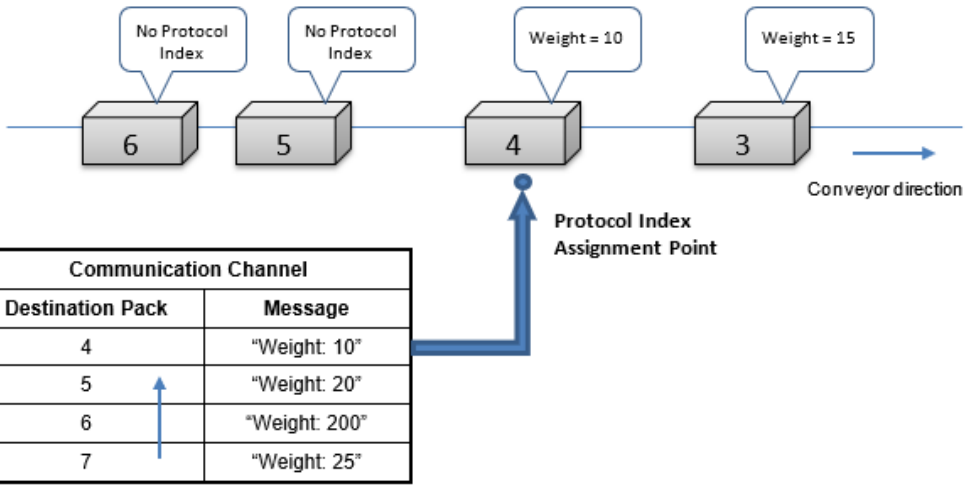
Another type of application is an In-line application where packages are introduced on a relatively slow moving conveyor (< 1 m/s). Typically the Code Combination Data Collection method is used. Either the codes are returned or a No Read message is returned.

The Presentation Mode can also be used with the Code Collection Data Collection method.

CHAPTER 6
PROTOCOL INDEX

OVERVIEW

Protocol Index is a feature allowing a custom message (string) sent from a remote Host to be assigned to a specific reading phase or pack. A typical application is assigning a message containing the weight, dimensions, etc., to each package passing through the system.

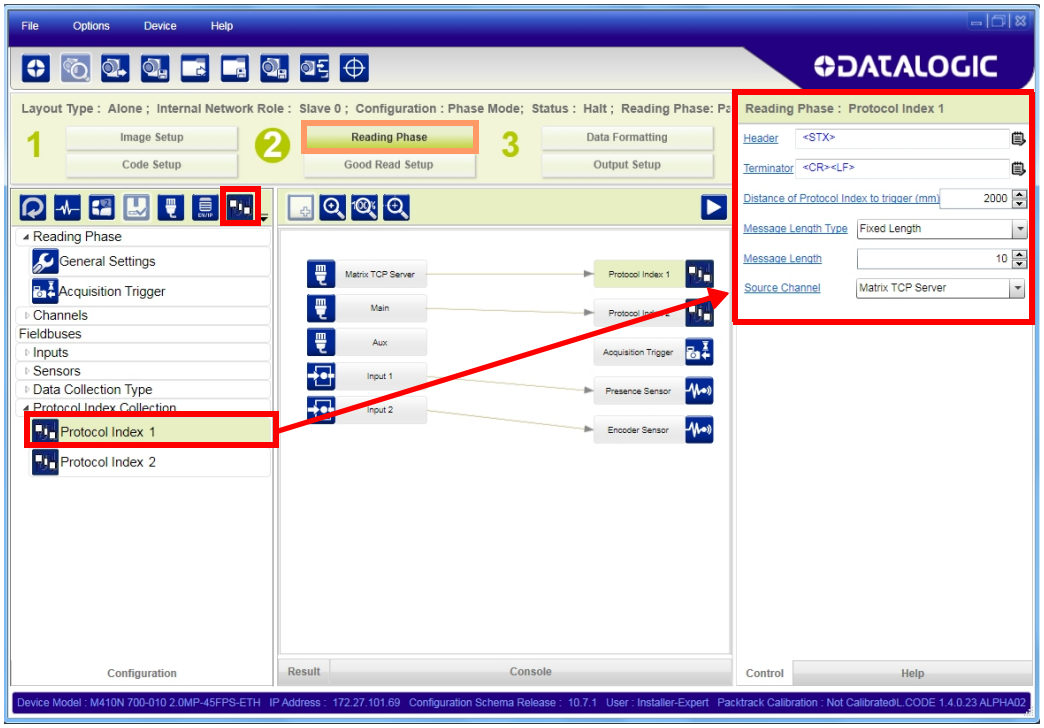


Communication Channel	
Destination Pack	**Message**
4	"Weight: 10"
5	"Weight: 20"
6	"Weight: 200"
7	"Weight: 25"

Protocol Index can be used either in Phase Mode (during the active reading phase) or in PackTrack operating mode (between the PS Offset and the Tx Line Distance).

Up to 10 different Protocol Indexes can be configured but each one must be transmitted over a dedicated communication channel (source). They can all transmit at the same or different assignment points.

COMMUNICATION CHANNELS	
Allowed	**Not allowed**
Serial	Modbus TCP Client embedded
Client TCP, port 1023	Ethernet/IP embedded
HMS Fieldbus	
Profinet-IO embedded	
Modbus TCP Server embedded	



Header and *Terminator* strings must be configured whenever a new Protocol Index object is added to a configuration. Either *Header* or *Terminator* can be blank but not both at the same time.

Typically if the Protocol Index Message Length Type is Variable Length, then both Header and Terminator will be needed.

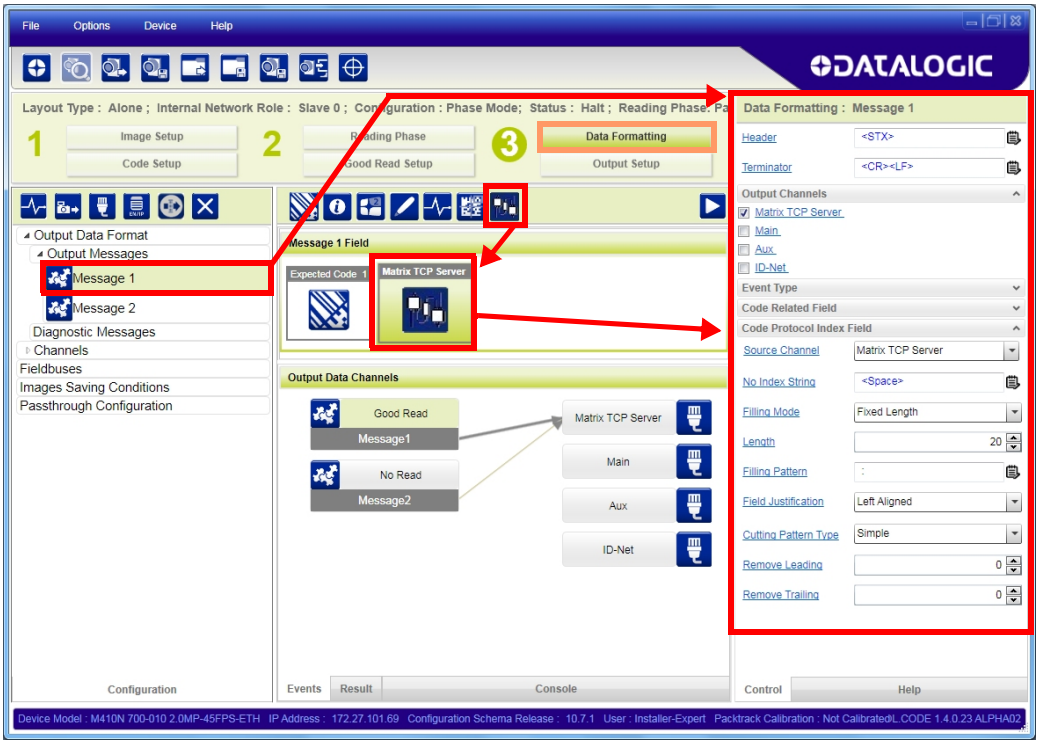
If the Message Length Type is Fixed Length, then at least the Header or the Terminator is necessary.

NOTE

Header and Terminator must be used for protocol index whenever an input channel (e.g. TCP/UDP client server) is also used as input of other services (e.g. phase/trigger generation). This is to avoid mixing triggers and protocol index messages.

INCLUDING PROTOCOL INDEX IN THE OUTPUT MESSAGE

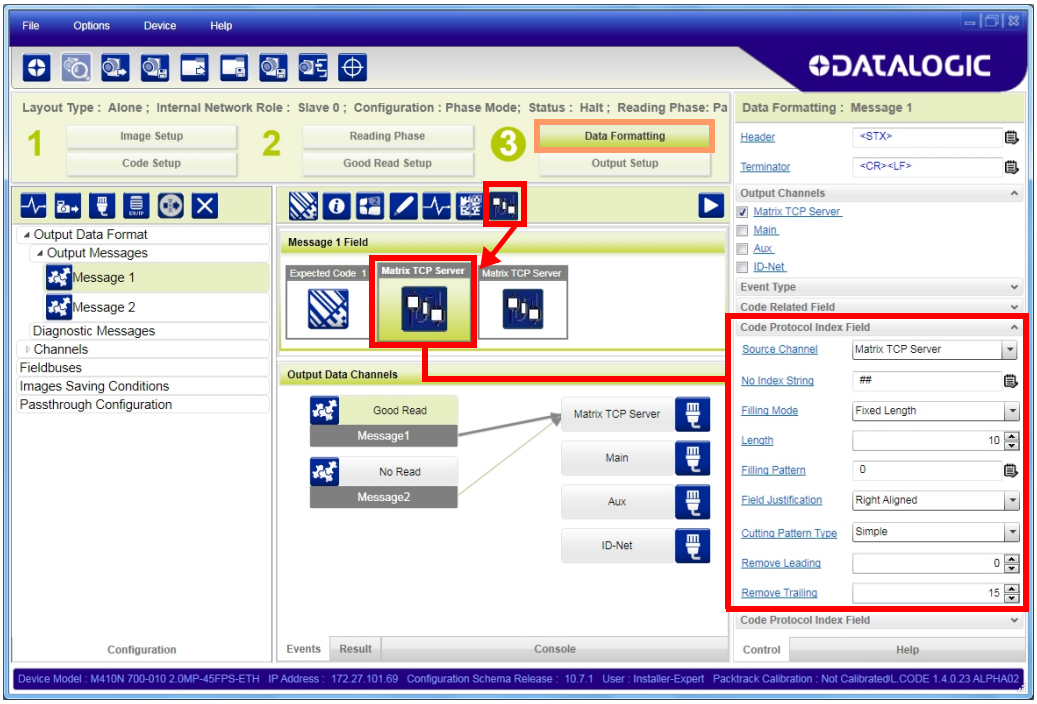
This message (in whole or in part), is also usually included in the output message. Each PI message can be added to the output message by including a Protocol Index Field.

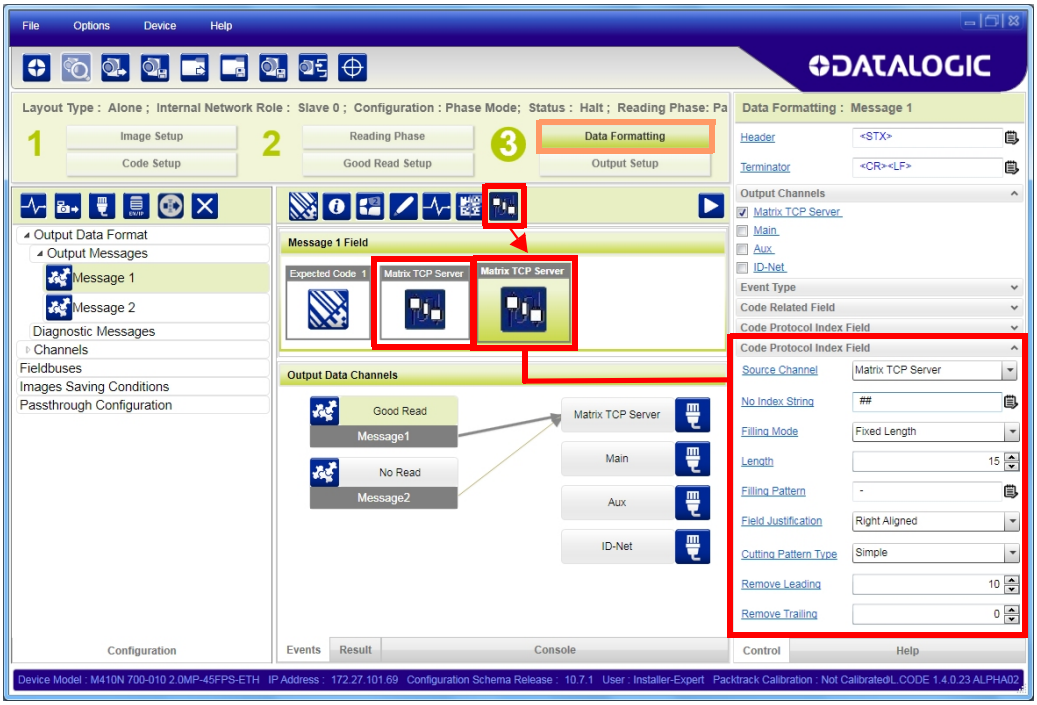


It is usually good practice to add the same PI message to both the Good Read and No Read output messages.

PARSING A COMPLEX PROTOCOL INDEX MESSAGE

A single complex Protocol Index message coming from a single channel can be received from a Host and can be divided into separate Protocol Index messages in the Data Formatting output message.





For further details on the Protocol Index Message, see the parameter descriptions in the DL.CODE Help On Line Protocol Index page.

CHAPTER 7
MESSAGE FORMATTING

Message Formatting is extremely flexible to allow a high level of customization depending on the application requirements. This however makes formatting more complex.

NOTE

A new Script Formatting tool based on javascript is provided to personalize the output message to a higher level than what is available using the standard Output Message tools. This tool is recommended only for people experienced with scripting. For details see "Script Formatter" on page 100**.**

Here we will break down the main characteristics of the Output Message and describe the standard Output Message tools.

Information relative to code reading is transmitted in standard formats on the device's selected interface. We refer to this as the OUTPUT MESSAGE. The general format of the OUTPUT MESSAGE is:

<HEADER><RESULTS><TERMINATOR>

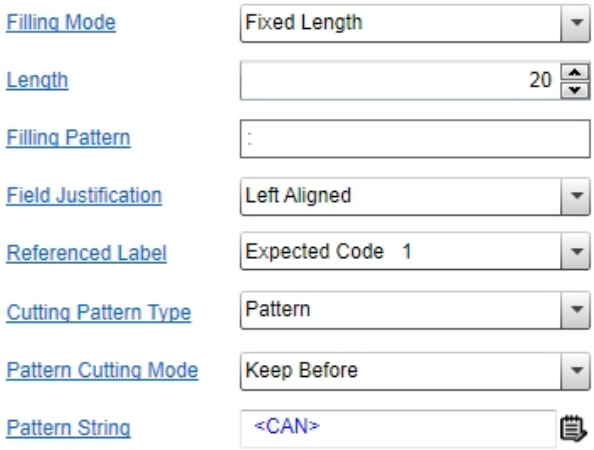
The RESULTS component is composed of several optional fields which are used in different combinations to create different output messages (like Successful Read, No Read, Statistical Data, Diagnostic Data, etc.). These fields can be inserted into the output message in any order. A list of the formatting fields is given in the table below.

Fields	Icon	Meaning
Code Related		Report various code related information types in the output message: Code Content, Number of Characters, Code Symbology (AIM ID), Simbology, Pixels Per Element, Average Module Size (mils), Symbol Size, Decoding Time, X-Coordinate, Y-Coordinate, Angle, Slave Number, Bounding Box. Each Code Related field can only contain one information type, but you can include multiple Code Related fields in the output message.
Global Statistics		Include Global Statistical Counters in the output message.

Fields	Icon	Meaning
Global Reading		This field offers different types of information depending on the data collection Analysis Mode and on the Operating Mode. When analyizing within an image, the Image Processing Time can be included in the output message to monitor performance. Typically used for Troubleshooting or fine tuning during installation. For Phase mode, several phase related counters are available. For PackTrack mode you can add the Pack ID to the output message.
Custom		Define custom strings to be included in the output message. Typically used to customize Failure messages like No Read or Multiple Read.
Diagnostics		Monitor individual Diagnostic Failure events by including them in the output message. Typically used for Troubleshooting. These can also be included independently from the Code Reading events by defining them in the Diagnostic Message. In this case they will be sent at regular intervals depending on the defined timeout.
Code Quality Grading		Include code quality trending to monitor print quality of code labels.

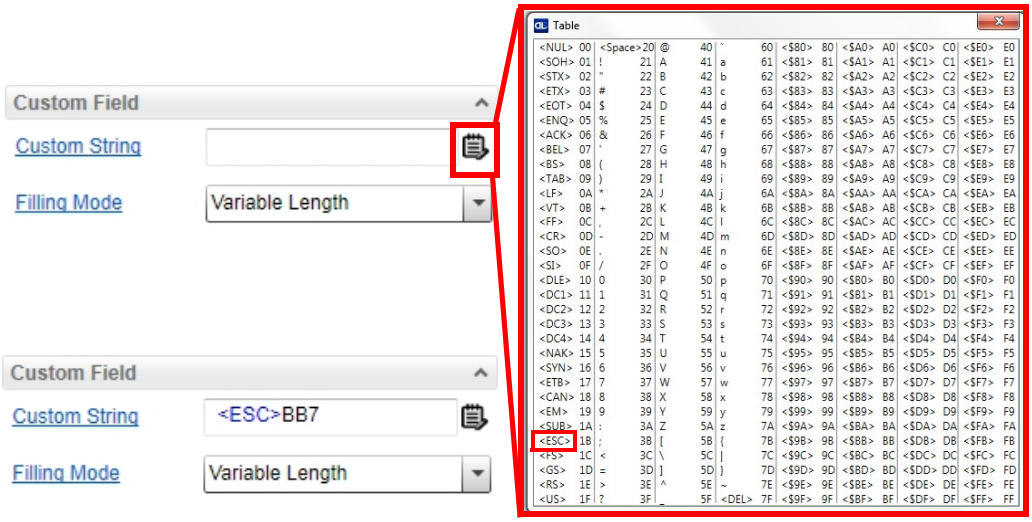
FIELD LENGTH MANAGEMENT

All field types by default are Variable Length fields but they also support Fixed Lengths with cutting and filling mode options.



INPUT STRINGS

All Input String fields have a table icon to the right of the field which allows you to input all ASCII characters including non-printable characters.



Just single-click on the desired character to insert it into the string.

While it is quicker to type normal characters directly from your keyboard, non-printable characters must be entered using the table.

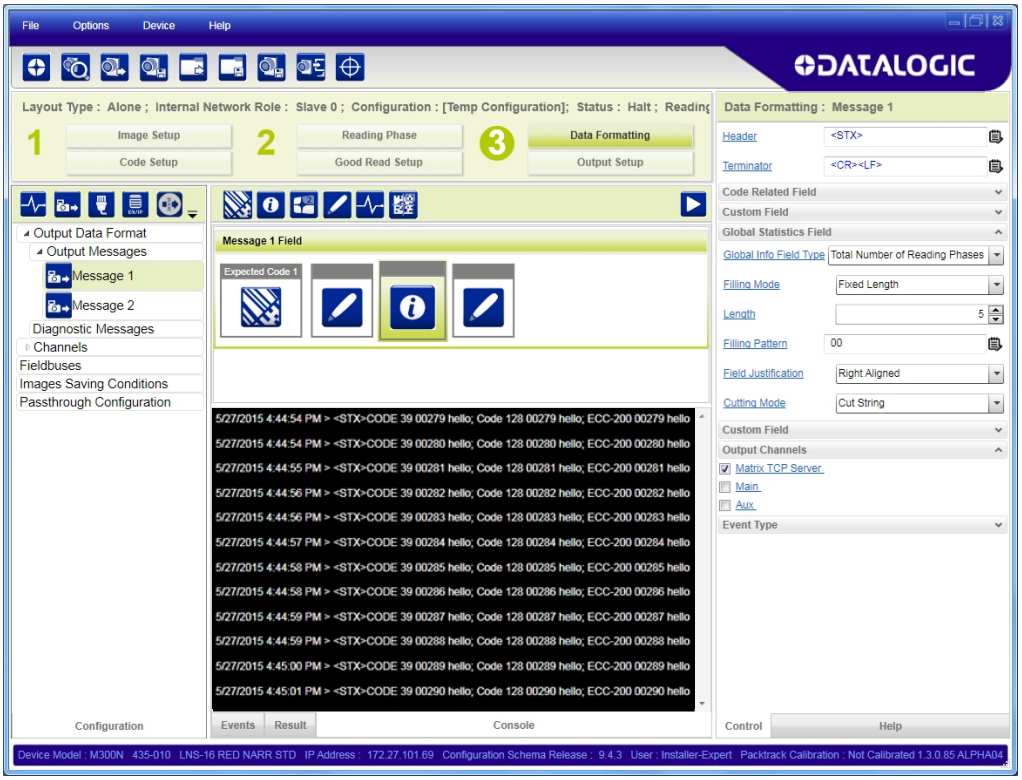
FIELD SEPARATOR

When Operating Mode is set to Code Collection or Code Presentation, the Field Separator character or string is used to separate each formatted code message within the complete output message #n. For example:

[Header] [formatted Code1] [Field Separator] [formatted Code2] ... [Terminator]

Each defined message #n can have a different Field Separator.

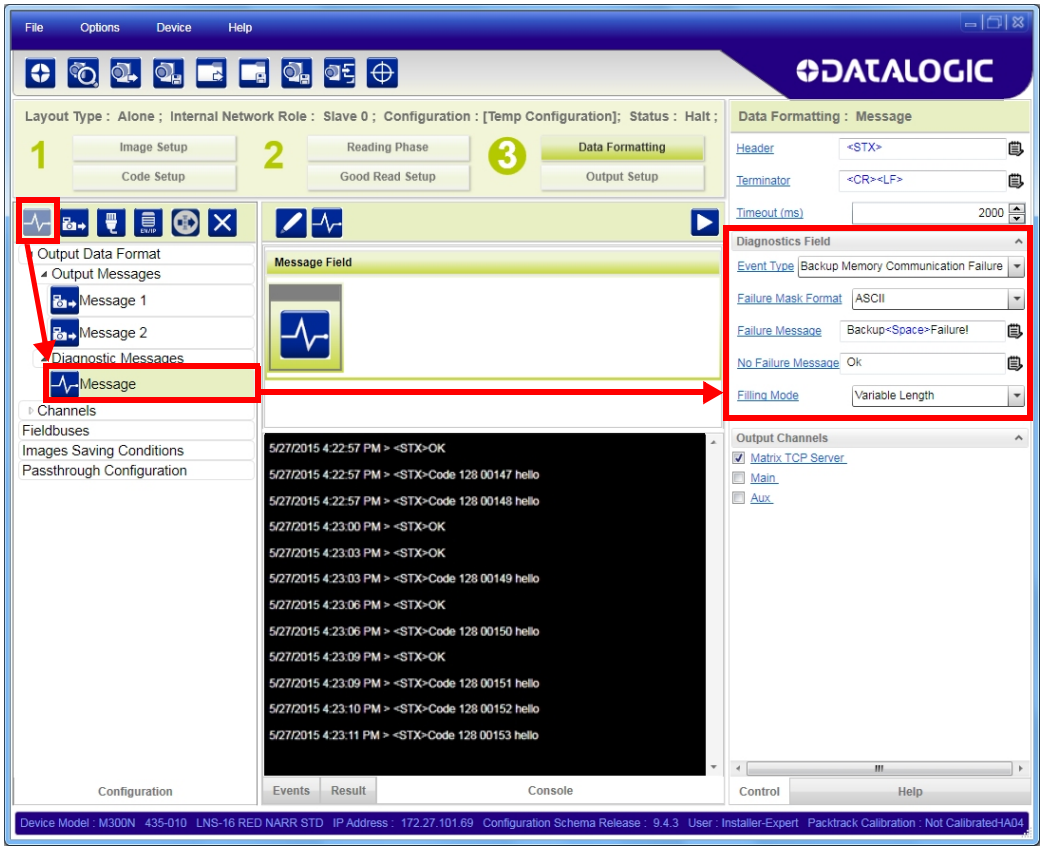
Example:



INDEPENDENT DIAGNOSTIC MESSAGES

As a troubleshooting tool or for error monitoring a Diagnostic Message can be sent independently from the code reading message.

For demonstration purposes, the figure below shows the Diagnostic Message for a **Backup Memory Communication Failure** being monitored and sent every 3 seconds on the output independently from the code reading output message.

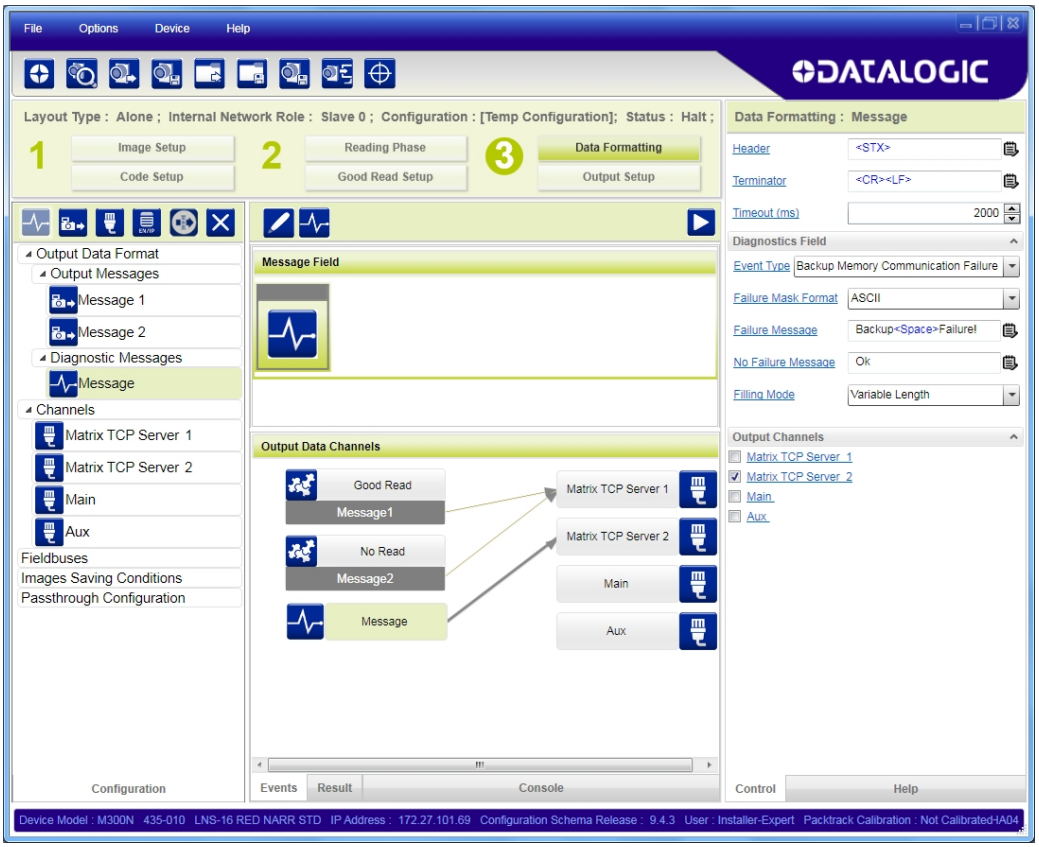


Typically only the Failure string is defined to avoid unnecessary message traffic.

Only upon failures, the Failure Mask sends a fixed 32-character mask. Each character represents the Standalone/Master device + 31 Slaves. The ASCII mask sends 0 =OK and 1 = Failure. The Binary mask sends non-printable characters.

The mask is appended to the Failure Content message (if defined).

This monitoring could be sent to another channel so as not to interfere with data communication, for example to another Matrix TCP Server connection, as shown below.



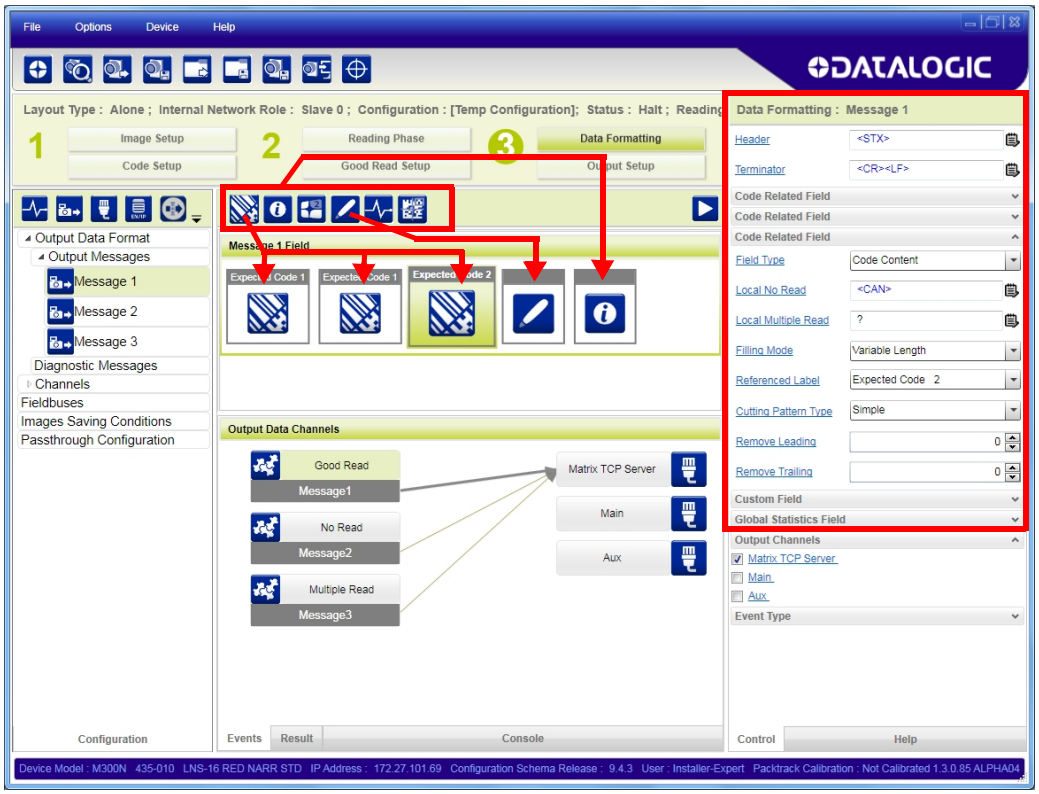
Several Diagnostic fields can be added to the Diagnostic Message for complete monitoring.

CODE COMBINATION MESSAGE FORMATTING EXAMPLE

In Code Combination the expected result is always known in terms of code reading.

To simplify these examples we will not consider Code Cutting or Justification, all fields are considered with the default value as Variable Length fields.

Recalling the example in "Code Combination" on page 79:



The DMECC200, QR Code and Code128 types are enabled and present in the Code Combination with the following logical combination:

DMECC200 OR QRCode AND Code128.

No Read is Enabled; Multiple Read is Enabled; Partial Read is treated as No Read

Click on the various field icons to add them to the Message Field Area to compose the message.

You can drag them into different positions to change the ordering of the message fields.

You can also delete them by selecting the field with the mouse (highlighted in green), and then delete it using your keyboard.

Then modify the Data Formatting parameters of each field in the parameters panel.

The Data Formatting Parameters are:

Message 1 = Good Read

Header String = <STX>

Terminator String = <CR><LF>

Referenced Label = Expected Code 1
Code Related Field = Code Symbology
Code Related Field = Code Content

Referenced Label = Expected Code 2
Code Related Field = Code Content

Custom Field
Custom String = <Space>->Space>

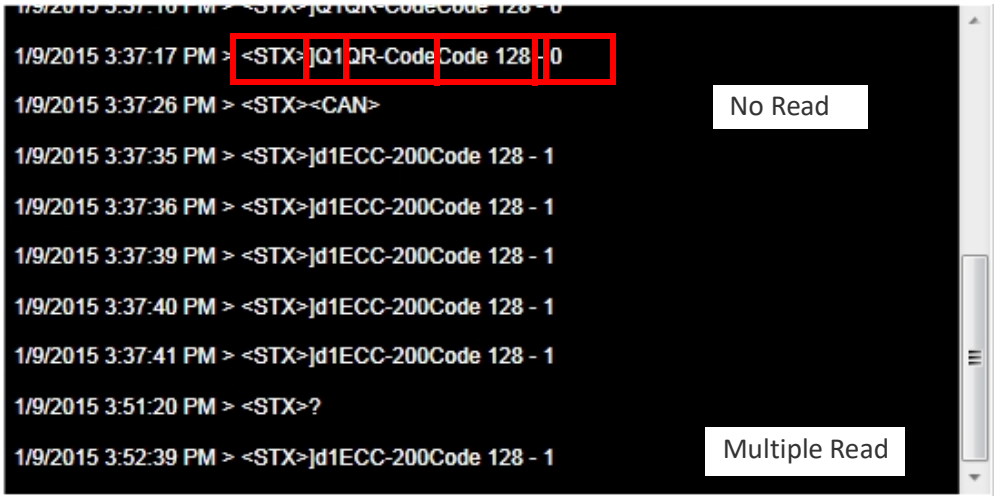
Global Statistics Field
Global Info Field Type = Total Number of No Reads

Message 2 = No Read
Custom Field
Custom String = <CAN>

Message 3 = Multiple Read
Custom Field
Custom String = <?>

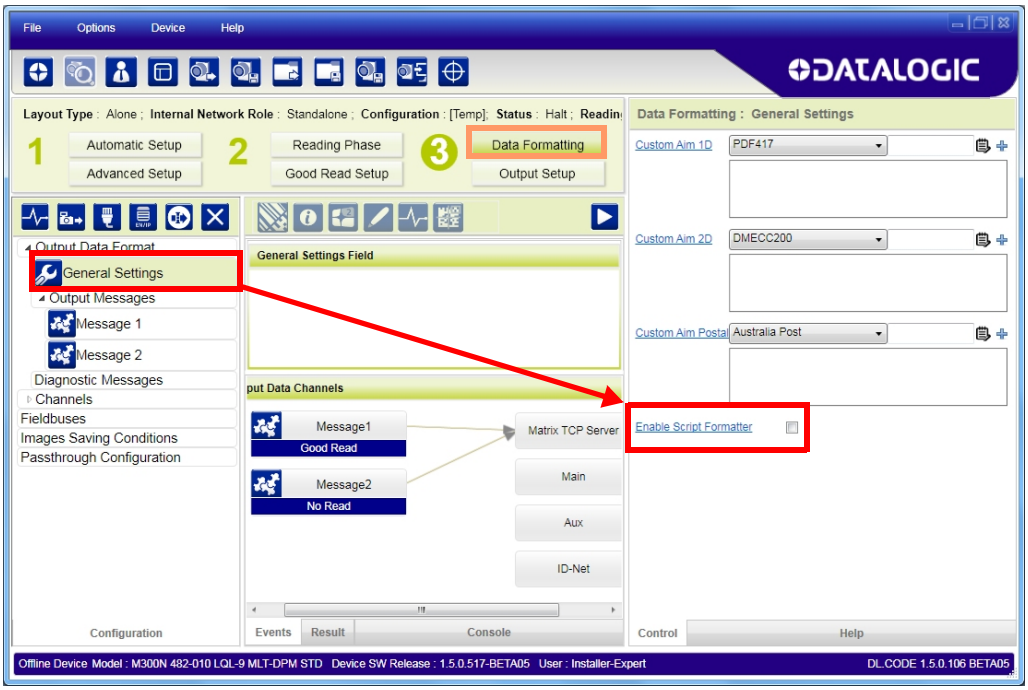
The Output Messages are:

Case 1: codes (DMECC200 AND Code128) or (QR Code AND Code128) are read
Output: the content of **Message 1 Good Read** is sent to the Host.

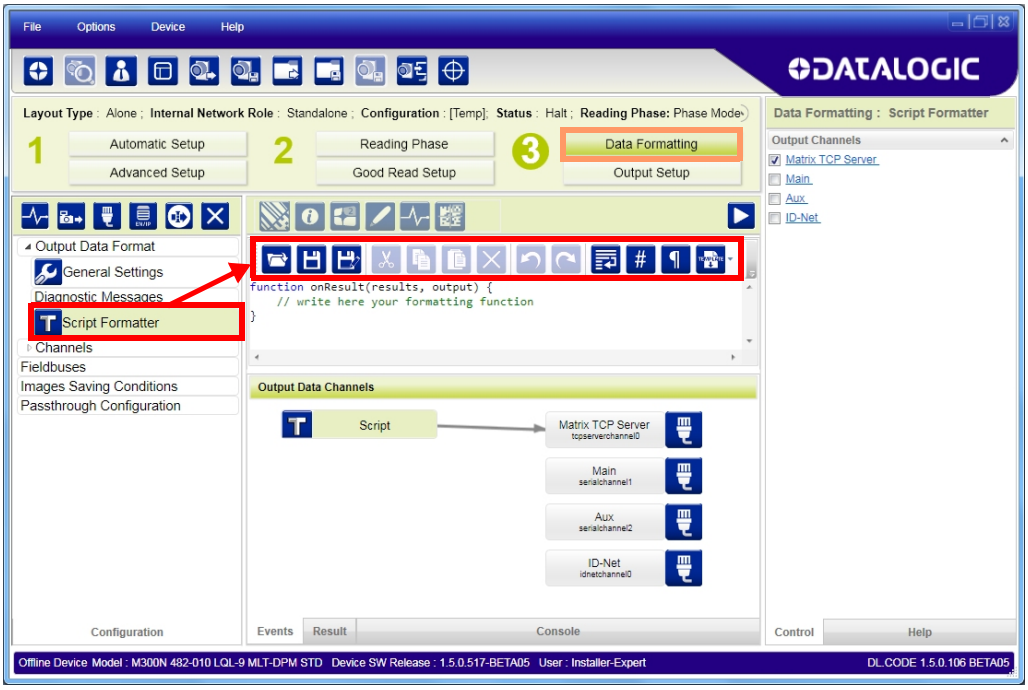


SCRIPT FORMATTER

Script formatting is designed to allow the highest level of output message configuration in place of the standard Output Message tools. It is recommended only for people experienced with scripting and can be enabled from the Data Formatting – General Settings branch.



The Script Formatter will now replace the Output Messages branch. It is complete with several editing tool icons to facilitate scripting within the DL.CODE GUI.



Script Formatter is based on an embedded JavaScript engine. For DL.CODE version 1.5.0 and later, the embedded JavaScript engine conforms to ECMAScript 5.0/5.1 Language Specification.

Global Objects

Inside the script engine, the script code makes it possible to access some global objects:

- Statistics (see "Statistics Object" on page 107) – an object to access device global reading statistics
- Diagnostics (see "Diagnostics Object" on page 108) – an object to access device diagnostic information
- Device (see "Device Object" on page 109) - an object to access device information

Callbacks

Function: onResult

This callback function is called every time the code analysis completes and allows for program controlled customized formatting of one or more messages based on decoding data results.

Argument	Type	Description
result	Result (see "Result Object" on page 102)	Allows access to all the decoding data
output	Output (see "Output Object" on page 106)	Allows access to all the output messages

Example:

This simple example accesses information wrapped inside the *Result* object to extract the content of all the decoded codes content and concatenate it into a space separated string. The generated string is then used as the output message by accessing the *Output* object.

```
function onResult(result, output) {
   var message = '';
   result.codes.forEach(function(code) {
   message += code.content + ' ';
   });
   output.setMessage(message ? message : 'no_read');
}
```

Objects Reference

Result Object

Provides functions to access decoding data.

Property	Type	Description
codes	Array	The array of all the decoded Code objects (see "Code Object" on page 103).
slots	Array	The array of all the configured Slot objects ("Slot Object" on page 105). The array contains a Slot object for each *Expected Code* in *Code Combination*. If the current Data Collection is not of type *Code Combination*, the property is set to **undefined**.
images	Array	The array of all the decoded images Image objects (see "Image Object" on page 105).
success	Boolean	This is set to **true** if the current result is a *Good Read*, *Success* or *Match*, otherwise it is set to **false**.
addresses	Array	The array of device addresses that decoded at least one code. Address 0 corresponds to the Master address.
readingCount	Number	The total number of decoded codes in the reading phase.
rejectedCodes	Number	The number of decoded codes removed from the output based on the analysis. An example is counting multiple read codes when the *Multiple Read* event is disabled.
imageAcquisitionTime	Number	Average time used to acquire an image (ms).
imageWaitTime	Number	Average time in which the device is busy between acquisition and decoding of an image (ms).
imageProcessingTime	Number	Average time used to decode an image (ms).
imageId	Number	The current image identifier. This is meaningful only if *Analysis* is *Within An Image*.
phaseId	Number	The current phase identifier. This is meaningful only if *Reading Phase* is set to *Phase Mode* and *Analysis* is *Within Phase*.
phaseDuration	Number	The duration of the last phase.
phaseOffDuration	Number	The duration of the last phase off period.
packId	Number	The current pack identifier. This is meaningful only if *Reading Phase* is set to *PackTrack Mode*.
dwCodes	Array	The array of all the decoded DWCODE™ objects (see "DWCode Object" on page 104).

Property	Type	Description
getProtocolIndex	String / Uint8Array	Retrieve protocol index data from the specified channel related to this result. The function returns an object with two properties: **content** and **binaryContent**. Both represent the protocol index data, the first as a printable string, the second as a Uint8Array.

Code Object

Provides access to the data of a single decoded code.

Property	Type	Description
content	String	The content of a code as a String.
center	Point	The center position of the code inside the image.
corners	Array	The four corners of the bounding box of the code as an array of Point objects (see "Point Object" on page 105).
addresses	Array	The array of device addresses that decoded this code. Address 0 corresponds to the Master address.
binaryContent	Uint8Array	The content of the code as an *Uint8Array*.
symbology	String	The code symbology name
aimId	String	The AIM identifier of the code symbology.
angle	Number	The code angle.
ppe	Number	The code pixels per element.
moduleSize	Number	The code module size.
decodingTime	Number	The decoding time in microseconds.
overrunTime	Number	The time in microseconds elapsed after tool timeout.
symbolSize	String	The code symbol size (if applicable).
quality	QualityMetrics	The Quality Metrics object (see "Quality Metrics Object" on page 104) that contains all the quality metrics info for the code.
imageIndex	Number	The index number of the image containing the code. This value is **undefined** if the code has been decoded on a slave device.
unreadableCode	Boolean	*true* if the code is an “unreadable code”, *false* it not.

DWCode Object

Provides access to the data of a single decoded DWCODE™.

Property	Type	Description
payloadRaw	String	The raw content of the code as String.
payloadCompatible	String	The compatible content of the code as String.
payloadEnhanced	String	The enhanced content of the code as String.
payloadNative	String	The native content of the code as String.
parsedContent	Object	If possible, a map with GS1 AIs and their contents.
unreadableCode	Boolean	*true* if the code is an "unreadable code", *false* it not.

Quality Metrics Object

Provides access to the quality metrics of a single decoded code. Each metric returns an object with two properties:

- **grade** - a string representing the quality grade (A, B, C, etc.)
- **value** - a numeric representation of the quality
- **decimalGrade** - the quality grade represented as a number between 0.00 and 4.00. Note that printing this number may result in a number with many decimal digits. Use *decimalGrade.toFixed(2)* to limit them.

Property	Type	Description
overall	Object	Overall quality grade.
decode	Object	Decode grade and value.
contrast	Object	Contrast grade and value.
modulation	Object	Modulation grade and value.
decodability	Object	Decodability grade and value.
meanEdgeContrast	Object	Mean Edge Contrast grade and value.
axialNonUniformity	Object	Axial Non Uniformity grade and value.
uec	Object	Unused Error Correction grade and value.
printGrowth	Object	Print Growth grade and value.
minReflectance	Object	Minimum Reflectance grade and value.
defects	Object	Defects grade and value.
fixedPatternDamage	Object	Fixed Pattern Damage grade and value.
gridNonUniformity	Object	Grid Non Uniformity grade and value.
reflectanceMargin	Object	ISO/IEC 15416:2016 Reflectance Margin (only value, grade is undefined).
quietZone	Object	Quiet Zone metric (value and decimalGrade are the same).

Point Object

This is an object with the *x* and *y* pixel coordinates of a point on an image. Values are relative to the upper left corner of the image (0,0).

Property	Type	Description
x	Number	The *x* coordinate (increases from left to right).
y	Number	The *y* coordinate (increases from top to bottom).

Slot Object

A slot corresponds to a single *Expected Code* in *Code Combination*. This object provides functions to access the Code objects (see "Code Object" on page 103) associated with the Expected Code.

Property	Type	Description
codes	Array	The array of all the Code objects assigned to this slot.
dwCodes	Array	The array of all the DWCODE™ objects (see "DWCode Object" on page 104) assigned to this slot.

Image Object

Provides access to the data of a single decoded image.

Property	Type	Description
acquisitionTime	Number	The acquisition time in microseconds.
waitTime	Number	The time in microseconds elapsed between end of acquisition and start of decoding.
processingTime	Number	The decoding time in microseconds.
imageId	Number	The current image identifier.

Output Object

Provides a function to set the output message to be sent to the output channel(s).

Property	Type	Description
setMessage	string, channel_id (optional)	Sets the output message. If no *channel_id* is specified, the message is sent to all connected channels, otherwise it is sent only to the specified channel (which has to be connected to the script message in any case). At the end of the formatting script execution, for each channel, the system sends its specific message, if it exists, otherwise it sends out the generic message.
setEvent1	none	Sets Script Event 1. If called, the event will be generated at the end of the script execution. From the Output Setup group, the event can be linked to any available output.
setEvent2	none	Sets Script Event 2. If called, the event will be generated at the end of the script execution. From the Output Setup group, the event can be linked to any available output.
setEvent3	none	Sets Script Event 3. If called, the event will be generated at the end of the script execution. From the Output Setup group, the event can be linked to any available output.
setFtpFileName	string, ftp_channel_id (optional)	Sets the name of the file where the FTP channel should save the content of the output message. If no ftp_channel is specified, the file name is used for all connected FTP channels, otherwise it is only used by the specified channel. If no file name is specified, the default “Unknown.tmp” file name is used.
saveImages	Array of Numbers	Specifies image IDs to save. This function makes it possible to specify the images to save.
	List of Numbers separated by commas	
	Array of Objects with id and filename properties	Specifies image IDs to save. In addition to IDs, it is possible to specify the image filenames.
	List of Objects with id and filename properties separated by commas	

NOTE

To use the saveImages function, the Image Saving Controlled by Script parameter must be enabled under Image Saving Conditions.

Statistics Object

Provides access to global device statistics.

Property	Type	Description
phases	Number	Number of Reading Phases.
goodReads	Number	Number of Good Reads.
noReads	Number	Number of No Reads.
multipleReads	Number	Number of Multiple Reads.
partialReads	Number	Number of Partial Reads.
phaseOverrun	Number	Number of Phase Overruns.
triggerOverrun	Number	Number of Trigger Overruns.
unexpectedPhaseOn	Number	Number of Unexpected Phase Ons.
images	Number	Number of Image Acquisitions.
fps	Number	Current frames per second.

This object provides the following functions:

Function	Type	Description
resetStatistics	none	Reset all statistics counter.

Diagnostics Object

Provides access to global device diagnostics. Each diagnostic returns an object with the following properties:

- **alarm** – a flag set to **true** if the alarm is currently active, otherwise it is set to **false**.
- **addresses** – an array of numbers where every number is the address of the device that has activated the alarm. If the alarm equals **false**, the array is empty. Address 0 corresponds to the Master address.

Property	Type	Description
slaveNoReply	Object	Slave No Reply diagnostic alarm.
slaveAddressDuplication	Object	Slave Address Duplication diagnostic alarm.
fieldbusCommunicationFailure	Object	Fieldbus Communication Failure diagnostic alarm.
fieldbusDHCPFailure	Object	Fieldbus DHCP Failure diagnostic alarm.
fieldbusConfigurationFailure	Object	Fieldbus Configuration Failure diagnostic alarm.
fieldbusTypeMismatch	Object	Fieldbus Type Mismatch diagnostic alarm.
wrongRotarySwitch	Object	Wrong Rotary Switch diagnostic alarm.
backupMemoryCommunicationFailure	Object	Backup Memory Communication Failure diagnostic alarm.
xrfSlaveNotDetected	Object	XRF Slave Not Detected diagnostic alarm.
protocolIndexFailure	Object	Protocol Index Failure diagnostic alarm.
sc5000CommunicationFailure	Object	SC5000 Communication Failure diagnostic alarm.
sc5000PresentationMessageResponseFailure	Object	SC5000 Presentation Message Response Failure diagnostic alarm.
noCameraHeadFailure	Object	No Camera Head Failure diagnostic alarm.

Device Object

Provides access to global device information.

Property	Type	Description
model	string	The complete model name of the device.
serial	string	The serial number of the device.
order	string	The order number of the device.
temperatures	Object	An object that stores the available temperatures of the device (Properties: sensor, motherboard, illuminator_driver, illuminator_leds, liquid_lens). Each property is a Number (the temperature) or undefined if no sensor is available on that specific device.
accelerations	Acceleration Object	All information from 3-axis accelerator, if available (if not, undefined)
operatingHours	integer	The number of operating hours of the device, if available (if not, undefined)
systemStarts	integer	The number of system starts, if available (if not, undefined)
systemReboots	integer	The number of system reboots, if available (if not, undefined)
applicationStarts	integer	The number of application starts, if available (if not, undefined)
applicationFailures	integer	The number of application failures, if available (if not, undefined)
readInput	name	Returns the named input line current value as integer (0/1). Names are “Input1”, “Input2”, and so on.
readOutput	name	Returns the named ouput line current value as integer (0/1). Names are “Output1”, “Output2”, and so on.

Acceleration Object

Provides access to acceleration data.

Property	Type	Description
raw	Array of 3 integers	The raw values form 3-axis accelerometer (X, Y, and Z)
forces	Array of 3 integers	The values of G forces (X, Y, and Z) in g
tilt	Array of 3 integers	The 3 components (X, Y, and Z) of the tilt angle
roll	Number	The roll angle
pitch	Number	The pitch angle

Adjustable parameters via HMP

The Script Formatter allows the user to define a set of parameters accessible in read and write from HMP with special commands. These parameters must be declared inside the Global Parameters object: this global object is always defined but it is an empty object by default. Via HMP it is possible to add/set/get properties inside the Parameters object.

NOTE

These script parameters are never stored in permanent memory!

Add/Set Parameters property

To add or set a property of the Parameters object via HMP, use the command SCRIPT_SET_PARAM.

> Command: SCRIPT_SET_PARAM<space><property_name><space><json_string><LF>
> Positive response: ACK<LF>
> Negative response: NACK<LF>

For example, suppose to have a script like this:

```
// Init a property inside the Parameters object to empty string.
Parameters.hmp_data = "";

function onResult(result, output) {
  // set the output message to the current value of Parameters.hmp_data
  output.setMessage(Parameters.hmp_data);
}
```

It is possible to change the output message with the HMP command:

```
SCRIPT_SET_PARAM hmp_data "next mesage content"<LF>
```

Get Parameters property

To get the value of a property of the Parameters obejct via HMP, use the command SCRIPT_GET_PARAM.

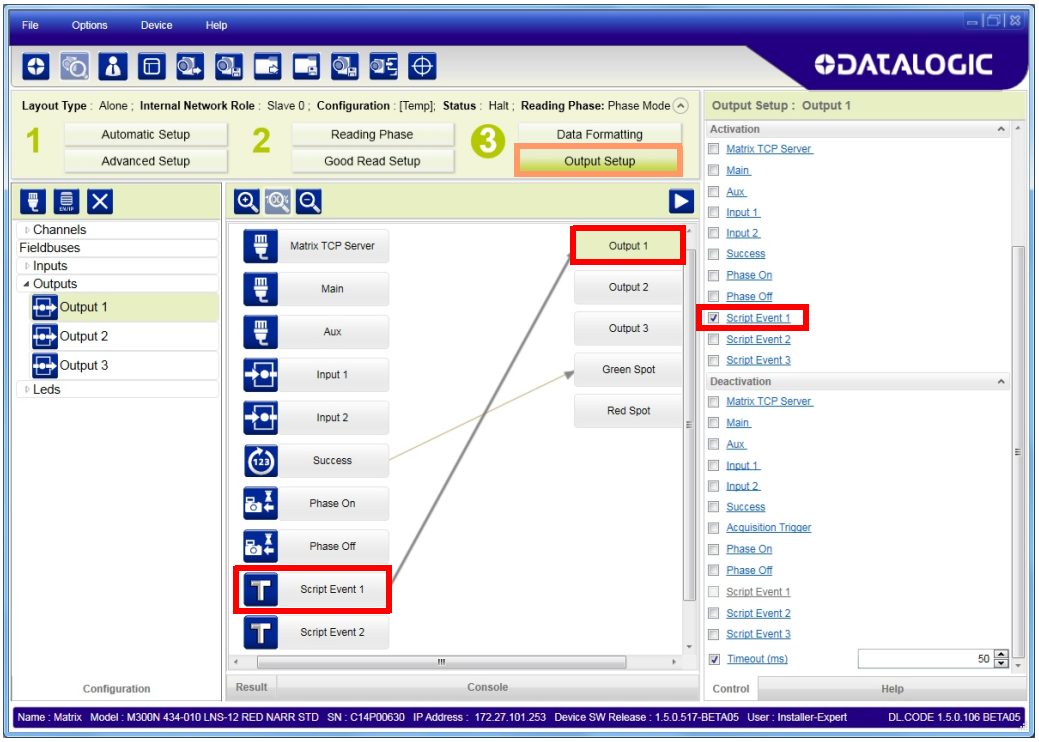
> Command: SCRIPT_GET_PARAM<space><property_name><LF>
> Positive response: ACK<json_string><LF>
> Negative response: NACK<LF>

For example, considering the previous example script, it is possible to get the current value of the hmp_data parameter with the HMP command:

```
SCRIPT_GET_PARAM hmp_data
```

Script Event Digital Output Control

As well as the output message configuration, the script can also contain Output object functions that can drive the reader's outputs.



CHAPTER 8
MONITOR

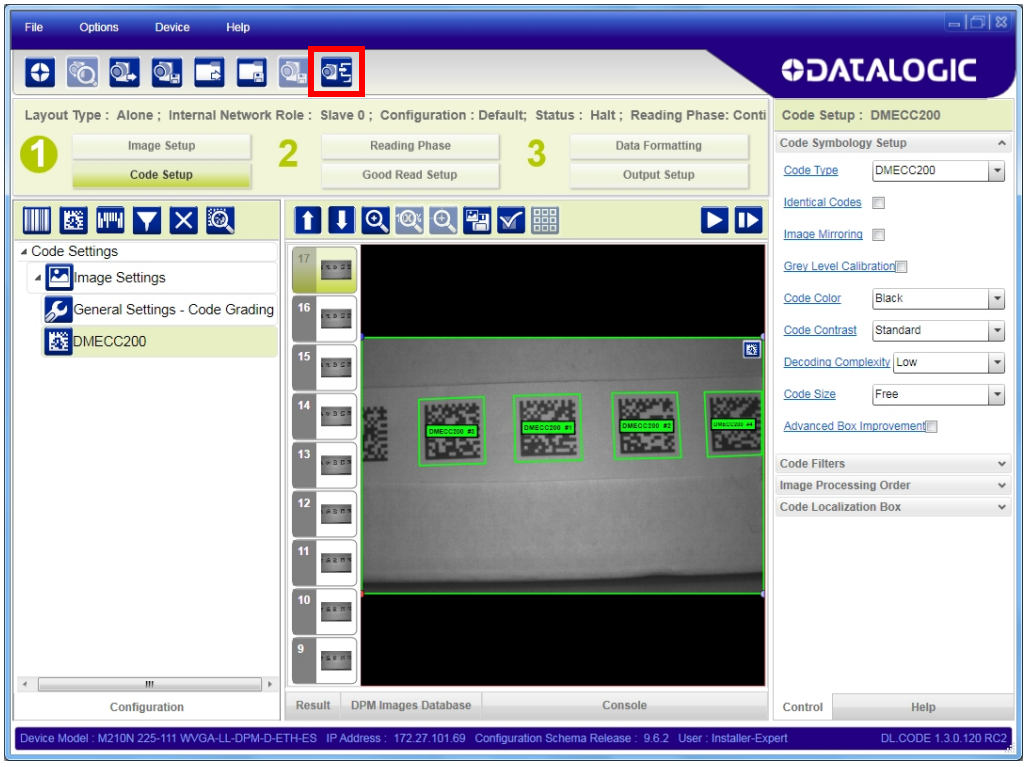
The Monitor feature is designed to check device operation from a remote PC even simultaneously with other monitoring PCs.

It can be used during installation or troubleshooting to check the device operation. The device operates with a minimum of DL.CODE overhead and therefore reading results are much closer to real-time performance.

Monitor also provides diagnostic alarm feedback.

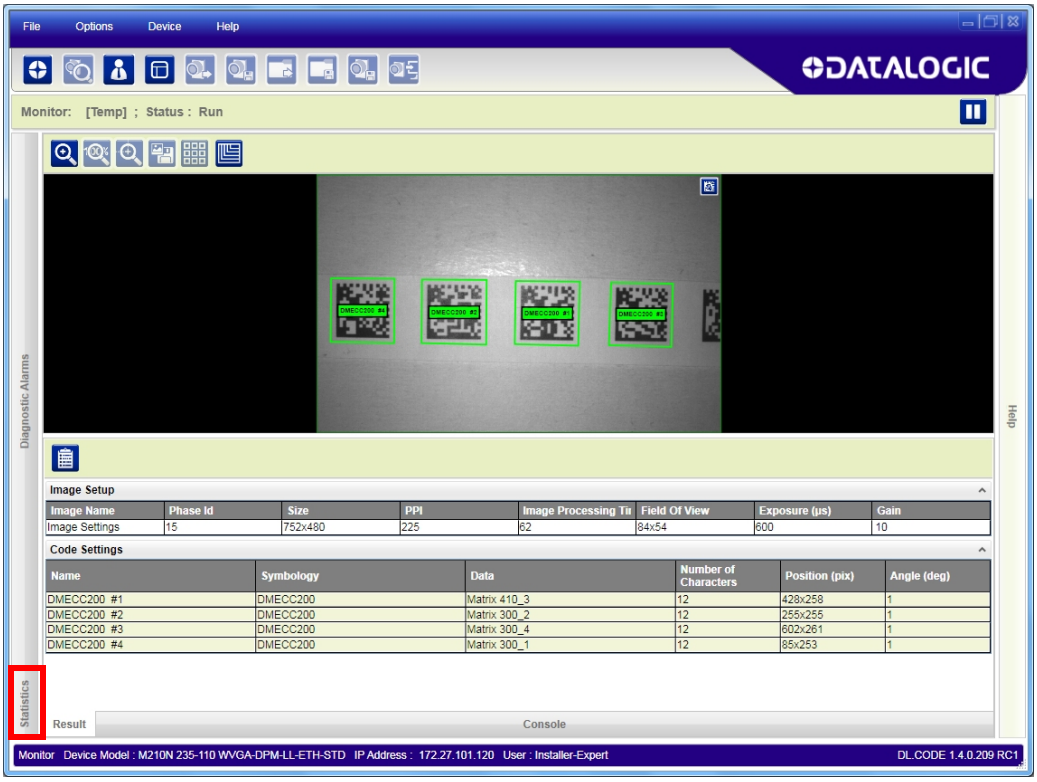
ACCESSING THE MONITOR

You can access the Monitor page through the File menu or the Monitor icon.



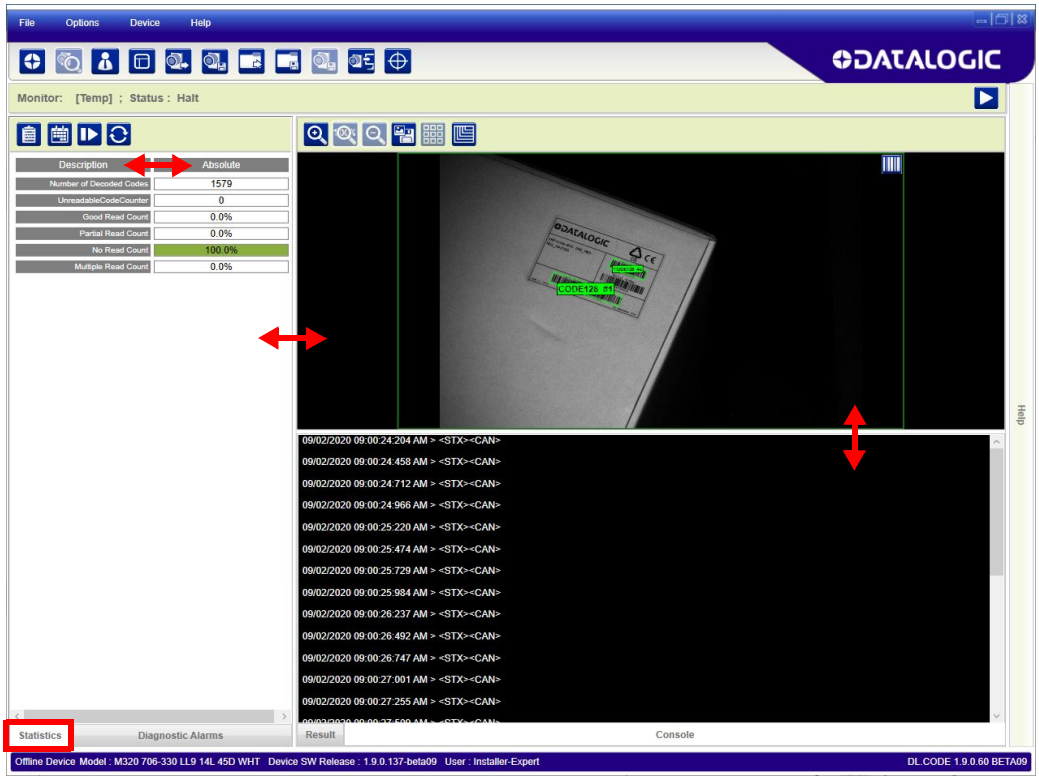
MONITORING STATISTICS

The Monitor loads with the reader in run mode. If the device is reading (in Continuous operating mode or One Shot/Phase Mode with trigger active), the reader will show image acquisition. The Results area underneath the image gives feedback on decoding results.

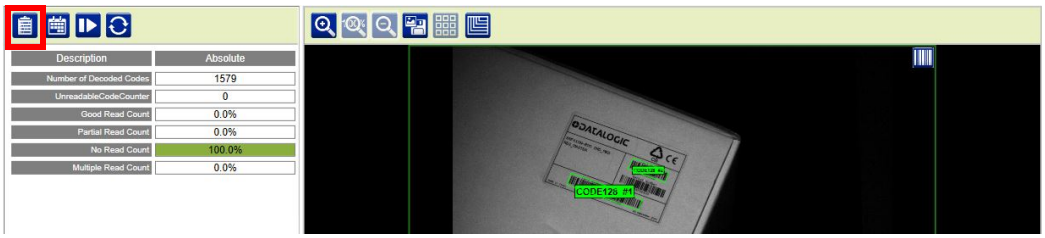


The Statistics page is collapsed to the left side of the DL.CODE window and can be opened by clicking on its tab.

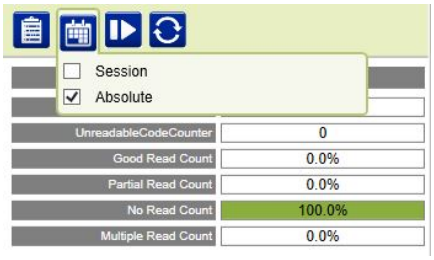
You can resize the various windows by dragging the edges with the mouse.



You can toggle between table view and chart view statistics by clicking on the icon.



You can also switch between Session and Absolute statistics.

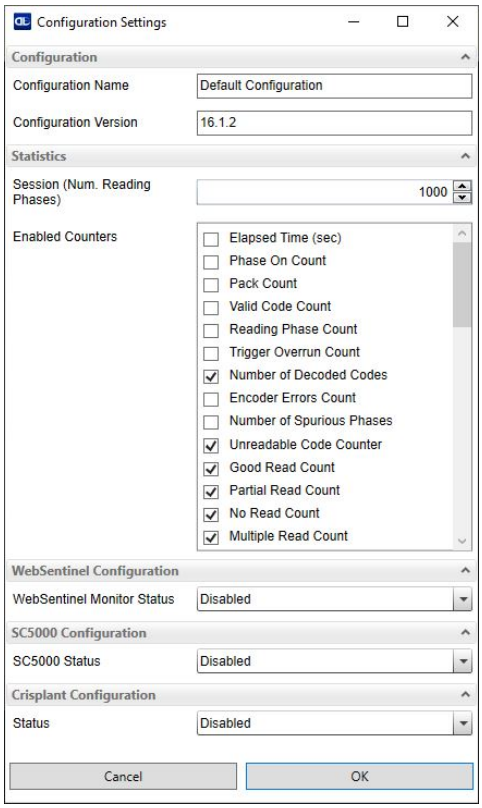


- **Session Statistics**: show only the fields selected that are represented as a percentage, rate, or average (i.e. Good Read, Partial Read, No Read, Average Pack Size, etc.) and refer to the last session or the last time the Statistics were reset up to the maximum Session number of reading phases. See below. A session ends if the device is connected to the DL.CODE configuration environment.
- **Absolute Statistics**: show all the fields selected in the Device menu > Settings > Configuration Settings > Statistics > Enabled Counters list from the last device power on or the last time the Statistics were reset.

Statistics Settings

From the Device menu > Settings > Configuration Settings you can set the number of reading phases to monitor for a Session (from 10 to 1000).

You can also choose which fields to visualize in the Monitor Statistics page.

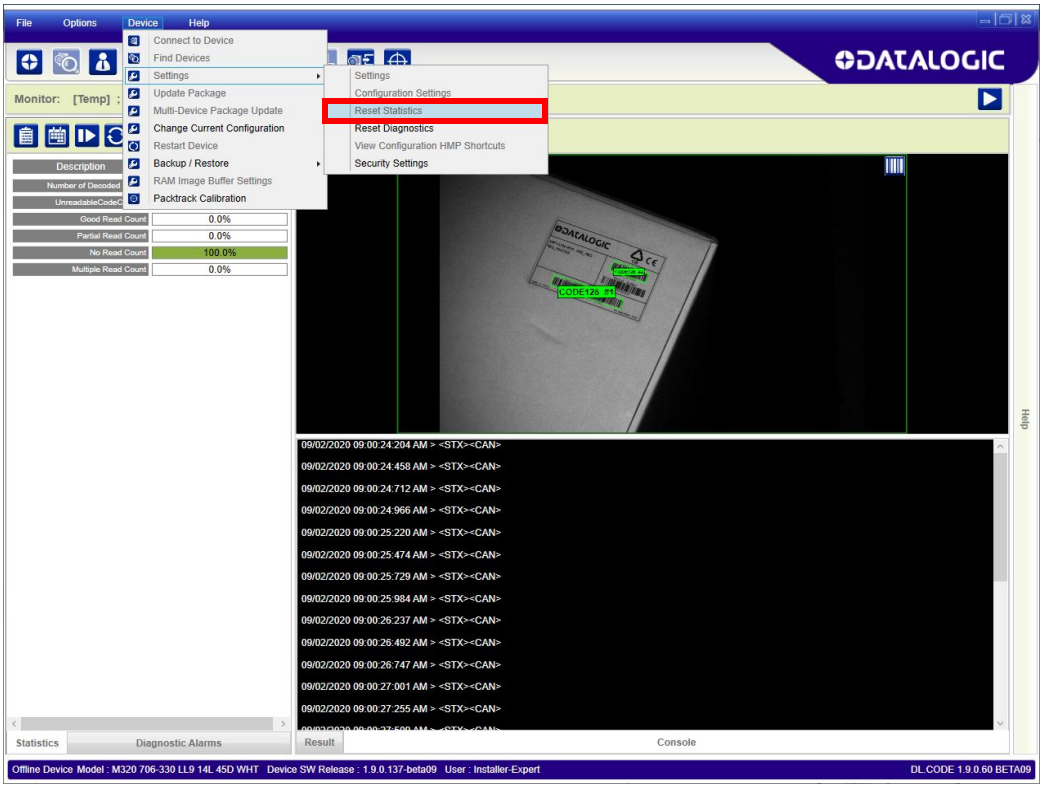


The counters that can be enabled include:

- **Elapsed Time (sec)**: the time elapsed from the start of image acquisition.
- **Phase On Count**: this counts all Phase On commands sent to the device.
- **Pack Count**: this counts all packs and parcels in systems where these are present in the reading area.
- **Valid Code Count**: this counts all valid codes decoded by the reader.
- **Reading Phase Count**: this counts all completed analyses (e.g. each phase in Phase mode or each image in Continuous mode).
- **Trigger Overrun Count**: this increases every time a trigger is ignored. This happens when the device cannot acquire an image when the trigger is received. Please note that in Monitor mode the device ignores more triggers than what normally happens.
- **Number of Decoded Codes**: this counts all codes that the reader has successfully decoded.
- **Encoder Errors Count**: this counts the errors reported by the encoder.
- **Unreadable Code Counter**: this counts all codes that the reader was not able to completely decode (only available for Code 39, Code 128, and Interleaved 2 of 5 symbologies). Note that only one unreadable code per frame can be detected.
- **Good Read Count**: the percentage of successfully decoded codes in the total of analyzed codes.
- **Partial Read Count**: this counts the analyses that resulted in a Partial Read.
- **No Read Count**: this counts the analyses that resulted in a No Read.
- **Multiple Read Count**: this counts the analyses that resulted in a a Multiple Read.
- **Successful Collection Count**: this counts the analyses that resulted in a Successful Collection.
- **Failed Collection Count**: this counts the analyses that resulted in a Failed Collection.
- **Match Code Count**: this counts the analyses that resulted in a Match Code.
- **No Match Code Count**: this counts the analyses that resulted in a No Match Code.
- **Phase Overrun Count**: this counts the events of Phase Overrun.
- **Unexpected Phase On Count**: this counts the events of Unexpected Phase On.
- **Multiple Phase On Count**: this counts the events of Multiple Phase On.
- **Valid Packs**: this counts valid packs and parcels in PackTrack mode.
- **Pack Too Short**: this counts the events of “Pack too short”. According to the PackTrack configuration, these can be ignored or cause the pack to be discarded.
- **Pack Too Close**: this counts the events of “Pack too close”. According to the PackTrack configuration, these can be ignored or cause the pack to be discarded.
- **Pack Too Long**: this counts the events of “Pack too long”. According to the PackTrack configuration, these can be ignored or cause the pack to be discarded.
- **Average Pack Size (mm)**: this measures the average size of packs in parcels.
- **Average Pack Distance (mm)**: this measures the average distance between packs.
- **Discarded Codes**: this counts all codes that are discarded and not decoded (e.g. codes exceeding the maximum number of characters set by the user).
- **Frame Rate**: this indicates the device acquisition frame rate.
- **Conveyor speed (mm/s)**: this indicates the conveyor speed in mm/s.
- **Encoder Frequency (Hz)**: this indicates the frequency (Hz) of the encoders used in the application.
- **Filtered Codes Count**: this counts all filtered codes.
- **Average Codes or Labels Found**: this indicates the average number of codes or labels found in each completed analysis.

- **Average Decoding Time (ms)**: this indicates the average time (ms) needed to decode all analyzed codes.
- **Image Acquisition Counter**: this counts the images acquired by the device.
- **Received Protocol Index**: this counts the Protocol Index messages received by the device.
- **Average Image Acquisition Time**: this indicates the average time (ms) needed for image acquisition.
- **Average Image Processing Time**: this indicates the average time (ms) needed to process an image.
- **Image Lost Counter**: this counts the number of images that were overwritten by newer images because there were no more buffers available.
- **Image Tx Failure Counter**: this counts the number of image transmission failures.

You can reset all the statistics (both Session and Absolute) clicking on the Reset Statistics icon or by selecting the Reset Statistics command from the Device menu > Settings.



MONITORING DIAGNOSTIC ALARMS

Any Diagnostic Alarms will show up as a warning light on the alarm panel. Most Alarms are relative only to Standalone or Master readers.

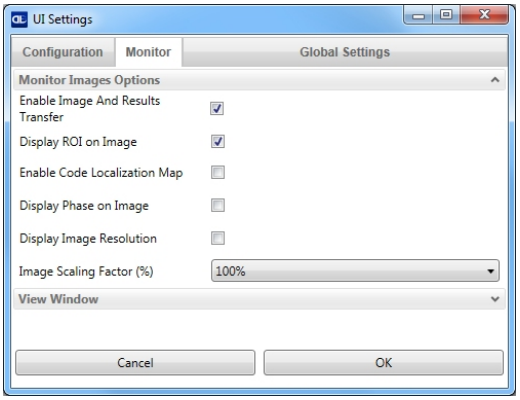
Alarm Code	Description	Meaning
1	Slave No Reply	No response to master from slave (shown in slave column).
64	Slave Address Duplication	Two or more slaves have the same address.
93	Focus Motor Failure	The focus motor is not working properly. Cycle power to the device. If the issue persists, contact Datalogic Technical Support.
100	Image Saving Tx Failure	This failure is triggered in case of communication problems with the server, so the problem could be due to the server or the network. For example: • Server unavailability or failure • Unavailability or failure of the network infrastructure • Full disk on the server • Antivirus not properly configured • Firewall issues, especially in passive mode (e.g. firewall blocks file transfer ports).

Alarm Code	Description	Meaning
124	High Temperature Warning	The device temperature is rising too high (max. device operating temperature is 50 °C). Check that the environmental temperature does not exceed 50°C (122 °F) and the device is properly mounted on a metal bracket. If the issue persists, contact Datalogic Technical Support.
171	Protocol Index Failure	The expected Protocol Index is not received. This alarm is activated when: • three or more consecutive packs/phases have no protocol index attached; • five or more packs/phases with no protocol index attached occur with a window of 100 packs/phases. When more than one configured Protocol Index is configured, the alarm condition is active whenever one of the errors above is true.
185	Backup Memory Communication Failure	There is a communication error between the reader and the external memory (BM100 module inside the CBX connection box or QLM Gateway).
187	Wrong Rotary Switch Selection	One or more of the selected BM100 Rotary Switch settings inside the CBX doesn't match the reader configuration memory.
189	Fieldbus Communication Failure	There is a communication error between the reader and the Fieldbus module inside the CBX connection box or QLM Gateway.
191	Fieldbus Type Mismatch	The Fieldbus module inside the CBX doesn't match the one saved in the reader configuration memory.
193	Fieldbus Configuration Error	A configuration error has occurred between the reader and the Fieldbus module inside the CBX connection box.
195	Fieldbus DHCP Problem	A communication problem has occurred between the DHCP server and the Fieldbus module inside the CBX connection box or QLM Gateway.
197	Low Temperature Warning	The device temperature is falling too low (min. device operating temperature is -10 °C). Check that the environmental temperature is not lower than -10°C (14 °F).
201	No XRF Slave(s) Detected	The XRF Master must have at least one XRF Slave.
301	SC5000 Communication Failure	There is a communication error between the reader and the SC5000 Controller.

Alarm Code	Description	Meaning
302	SC5000 Presentation Mes-sagge Response Failure	There is a configuration error between the reader and the SC5000 Controller, typically TX Line position or start position (address).
310	Excessive Bumps Warning	The device is detecting mechanical shocks of an intensity greater than 10G. Check that the frame where the device is mounted is not subjected to mechanical shocks.
999900	System Failure	Very severe failure on hardware system ini-tialization.
999901	Video Sensor Failure	Failure on video sensor initialization (eg. communication with sensor error).
999902	Illuminator Failure	Failure on illuminator initialization: illumina-tor not programmed or failure on eeprom information retrieval.
999903	Liquid Lens Failure	Failure on liquid lens initialization: focaliza-tion/thermal compensation tables missing, thermal sensor reading failure or voltage setting error on startup.
999904	I/O Failure	Failure on input/output or serial ports initial-ization.
999905	Thermal Sensor Failure	Failure on thermal sensor initialization: ther-mal sensor interrogation error.
999906	Generic Failure	Minor failure on system startup.
999998	Camera Head Failure	There is a communication error between the reader and its internal camera module.

MONITOR SETTINGS

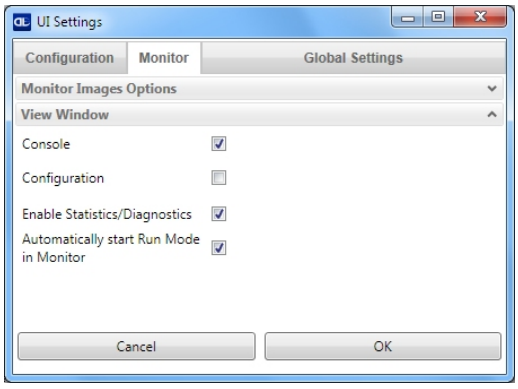
Monitor Images Options



Several options can be set for the Monitor window. They are set in the Options>UI Settings menu on the Monitor tab.

- **Enable Image and Results Transfer**: enables the image feedback in the Monitor window.
- **Display ROI on Image**: shows the bounding box around each code on the image.
- **Enable Code Localization Map** shows…
- **Display Phase on Image**: shows the Phase identification number in white letters over the upper left-hand side of the image.
- **Display Image Resolution**: shows the Image resolution in white letters over the upper left-hand side of the image.
- **Image Scaling Factor**: allows downsizing the image subsampling to increase the visualization performance.

View Window



You can select several options to view in the Monitor window.

- **Console**: shows the Console panel to view output messages.
- **Configuration**: shows the Configuration panel to allow viewing the configuration parameters (read-only). Configuration cannot be performed from the Monitor window.
- **Statistics/Diagnostics**: shows the Statistics and Diagnostic Alarms panels.
- **Automatically start Run Mode in Monitor**: when launching Monitor the window opens with the device in Run Mode. This is the default setting.

CHAPTER 9
IMAGE SAVING

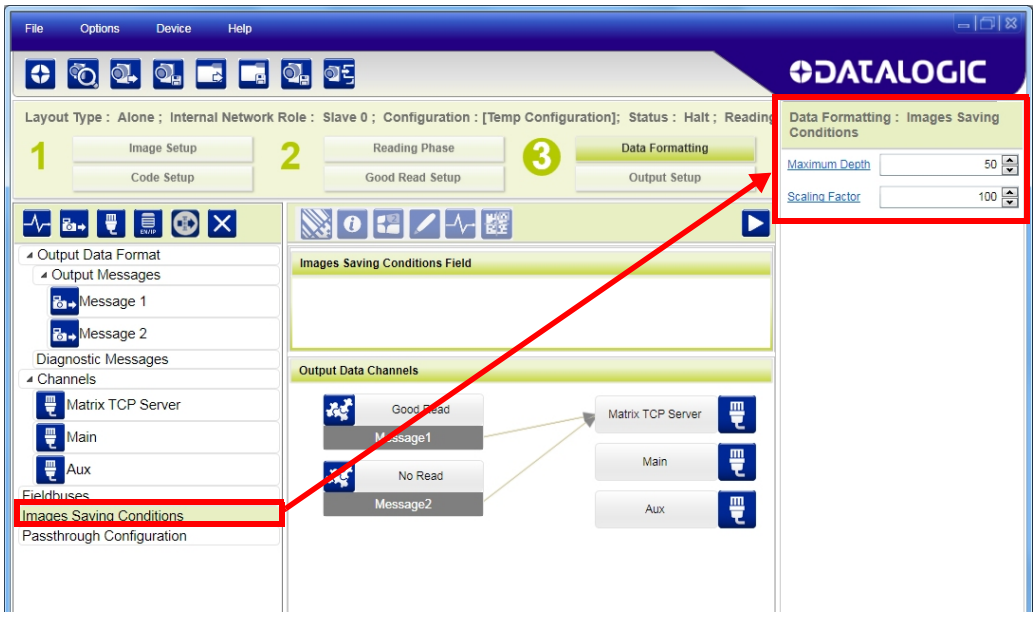
DEVICE IMAGE BUFFER

The Device Image Buffer allows saving captured images to the internal device memory (RAM) for data collection analysis. These images can be sent to a remote or local PC, an FTP server, or an SFTP server at run time; or they can be downloaded from the buffer when the device is offline (not in run mode).

CAUTION

The internal device memory has a circular buffer to avoid overrunning the available memory. To save images correctly to the internal buffer, you need to calculate the number of images the buffer can hold based on the image resolution for your device model. More images can be stored in memory by scaling down the image size.

This does not apply to Matrix 120 devices, which can save up to 2 images to the internal buffer.



NOTE

Transferring images remotely may slightly reduce the decoding rate. It is recommended to use this feature only when enough time is available to guarantee the decoding of all images.

Image Saving Conditions

Maximum Depth sets the total number of images that can be sent from or saved to the Device Image Buffer. Set this figure to a practical value taking into consideration the above mentioned CAUTION.

Scaling Factor allows reducing the size of images to be saved so that a greater number of images can be saved in the Device Image Buffer.

Pixel Subsampling reduces the image dimensions by subsampling the original image. The saved image will have 1 pixel every X pixel of the original image, where X is the parameter value, as reported in the table below.

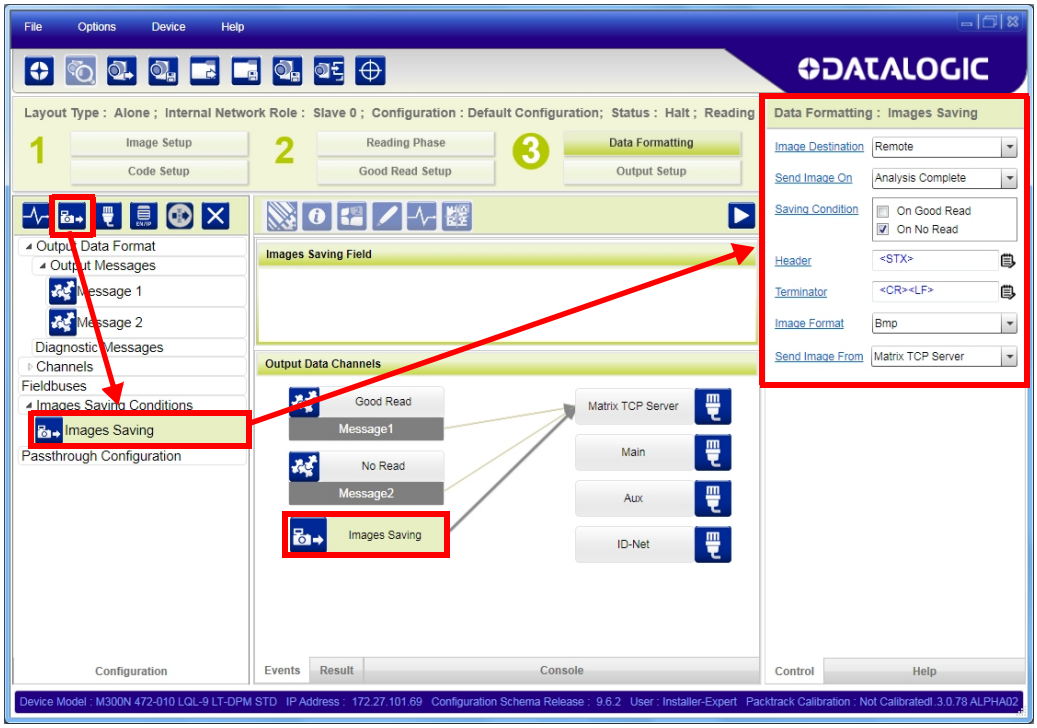
SUBSAMPLING VALUE	SAVED IMAGE WIDTH AND HEIGHT RESPECT TO THE ORIGINAL IMAGE	SAVED IMAGE SIZE IN PIXEL RESPECT TO THE ORIGINAL IMAGE
1 (default)	Same	Same
2	1/2	1/4
3	1/3	1/9
4	1/4	1/16
5	1/5	1/25
6	1/6	1/36
7	1/7	1/49
8	1/8	1/64

When width and height are not perfectly divisible by the pixel subsampling, they are rounded down (e.g. 100/3 = 33).

In case of image cropping, subsampling works on the cropped image.

Image Saving Using Matrix TCP Server/Client

To add an image saving condition to the configuration, click on the Add Image Saving Condition icon.



When selecting a Remote **Image Destination**, the default value is **Send Image From** Matrix TCP Server (on-board Ethernet) channel to an external TCP Client. If desired, a different dedicated Matrix channel can be added for image transfer.

NOTE

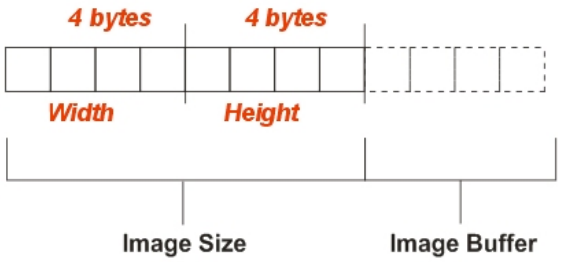
In Phase Mode, when only one communication socket is used to send Phase On and Phase Off commands as well as receive result strings sent by the device, the device might stop responding if the Client application does not process all sent data.

According to the selected **Saving Condition** parameter, after data collection analysis is complete, images can be sent at run time directly from the Device Image Buffer to a remote or local PC.

The Image Saving format is:

[Header] [Image Buffer] [Terminator]

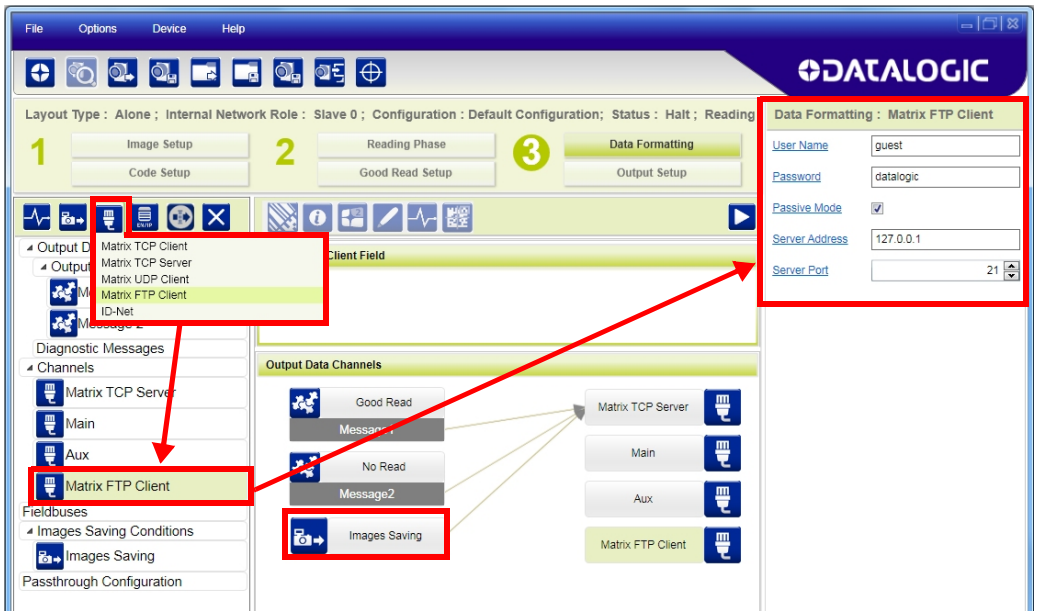
The Image Buffer contains the image data in the selected format (.bmp, .jpg, .png, .tif).



If the RAW data format is selected, the Image Buffer data is preceded by 8 bytes which indicate the image size: the first 4 bytes indicate the image columns (width) while the last 4 bytes indicate the image lines (height) in little-endian ordering.

Image Saving Using Matrix FTP Client

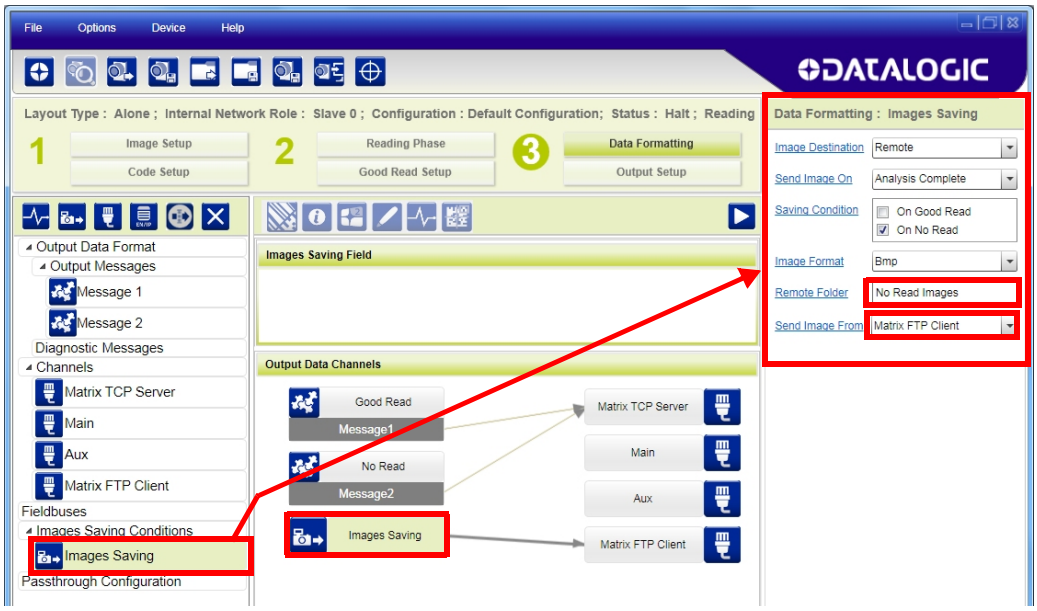
Images can be sent at run time directly from the device image buffer to an external FTP Server through a Matrix FTP Client channel. The Matrix FTP Client must be added as a new communication channel since it is disabled by default. Do this by selecting the channel icon.



- The Server Address must match the external FTP Server PC.
- The Username and Password must match the ones required by the external FTP Server PC.

Now select Images Saving and set the **Send Image From** parameter to Matrix FTP Client.

Input the name of the Remote Folder on the FTP Server to which the image files must be saved.



NOTE

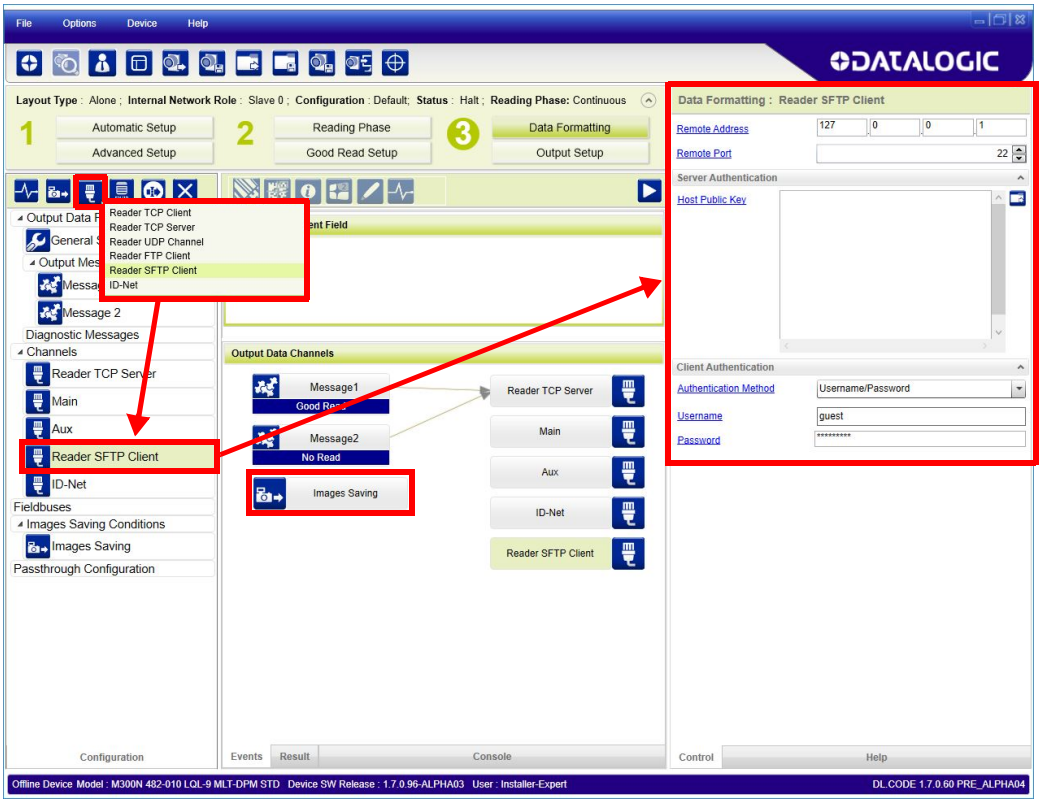
Some FTP Clients allow reducing the speed limit of image transferring. However, it is not advisable to set a speed limit below 4069 KB/s.

Image Saving Using Matrix SFTP Client

NOTE

Currently this feature is only available on Matrix 220 reader.

Images can be sent at run time directly from the device image buffer to an external SFTP Server through a Matrix SFTP Client channel. The Matrix SFTP Client must be added as a new communication channel since it is disabled by default. Do this by selecting the channel icon.



- The Remote Address must match the external SFTP Server PC.
- The available authentication methods are Username/Password and Public Key:
  -The Username and Password must match the ones required by the external SFTP Server PC.
  -The Public Key must be set by uploading a .crt or .pub file from the local PC.

Now select Images Saving and set the **Send Image From** parameter to Matrix SFTP Client.

Input the name of the Remote Folder on the SFTP Server to which the image files must be saved.

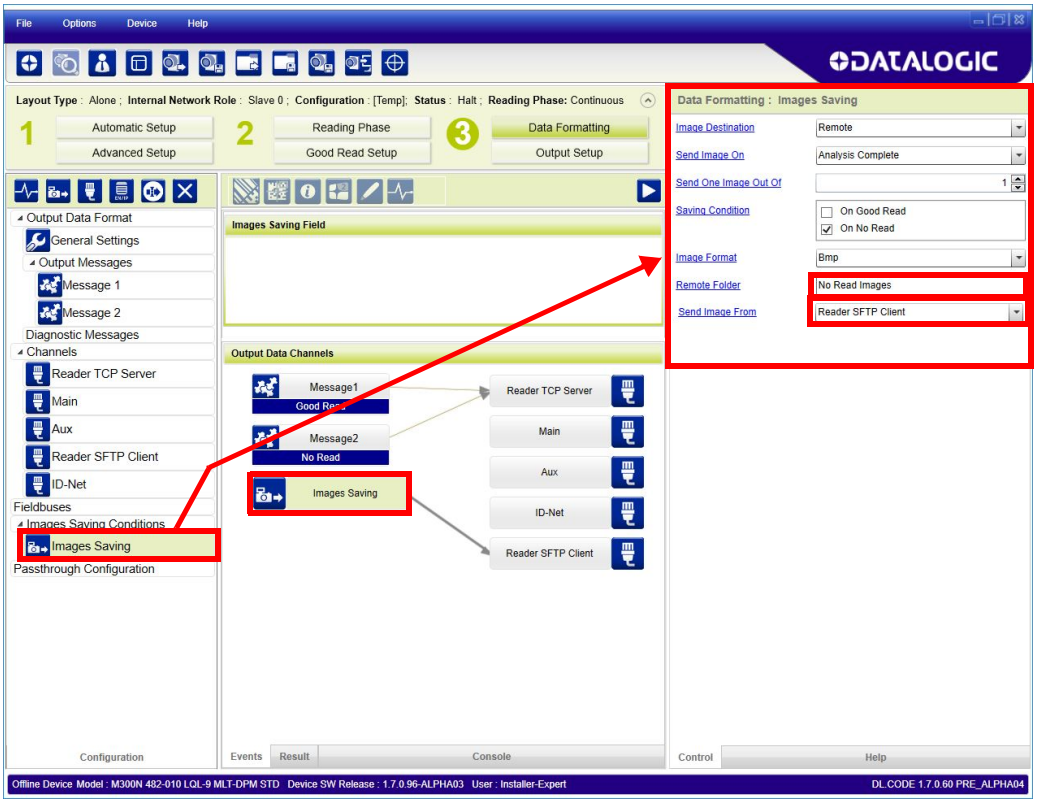


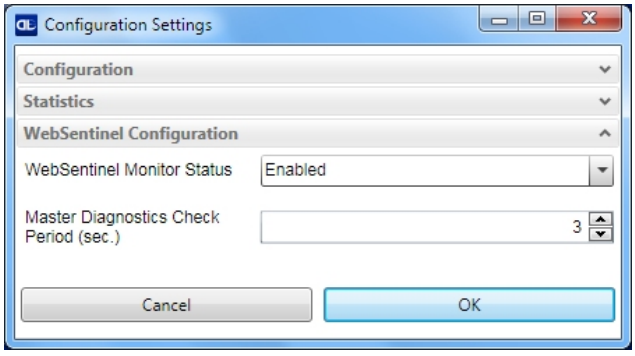
Image Saving On Demand to WebSentinel FTP Server

Images can be sent to the WebSentinel FTP Server through a Matrix FTP Client channel upon receiving a Download command from Datalogic WebSentinel™ through the Download buttons in the Event Search tab. See the description in the Datalogic WebSentinal User's Manual.

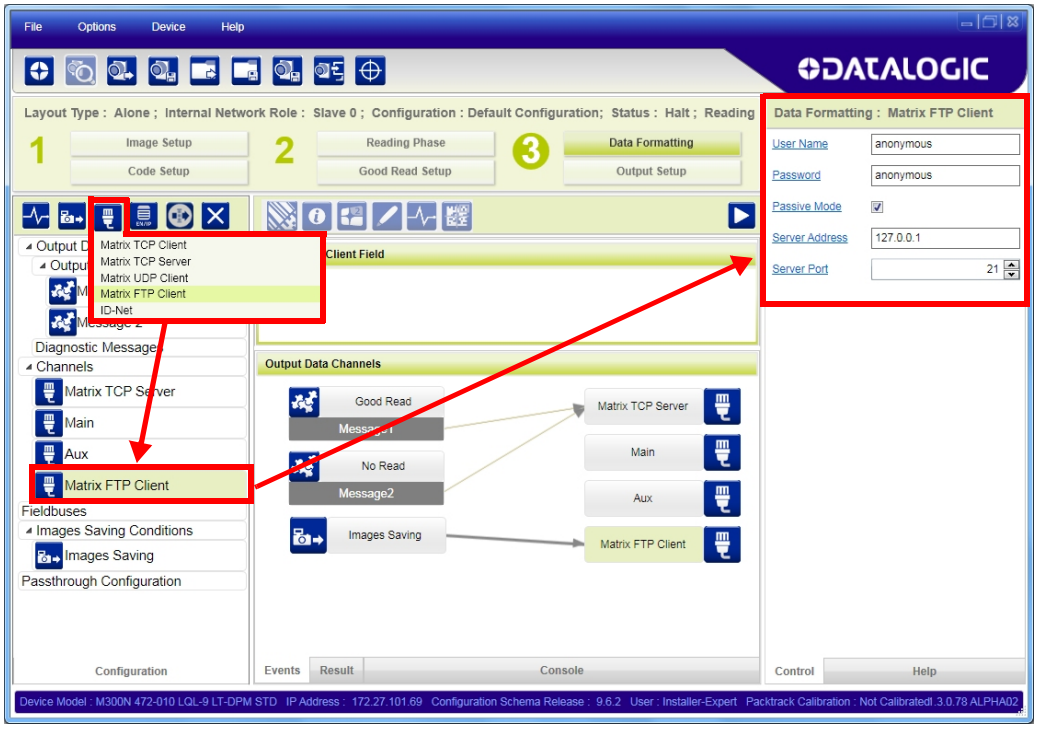
This selection applies only to One Shot and Phase Mode Operating Modes.

NOTE

In order for Datalogic WebSentinel to communicate with the Matrix array you must enable WebSentinel Monitor Status in the 130 Device>Settings>Configuration Settings menu.



The Matrix FTP Client must be added as a new communication channel since it is disabled by default. Do this by selecting the channel icon.



- The Server Address must match the external WebSentinel FTP Server PC.
- The Username and Password must match the ones required by the external WebSentinel FTP Server PC. For WebSentinel the defaults are **anonymous**, **anonymous**.

Now select Images Saving and set **Send Image On** to Demand and set the **Send Image From** parameter to Matrix FTP Client.

Input the name of the Remote Folder on the FTP Server to which the image files must be saved.

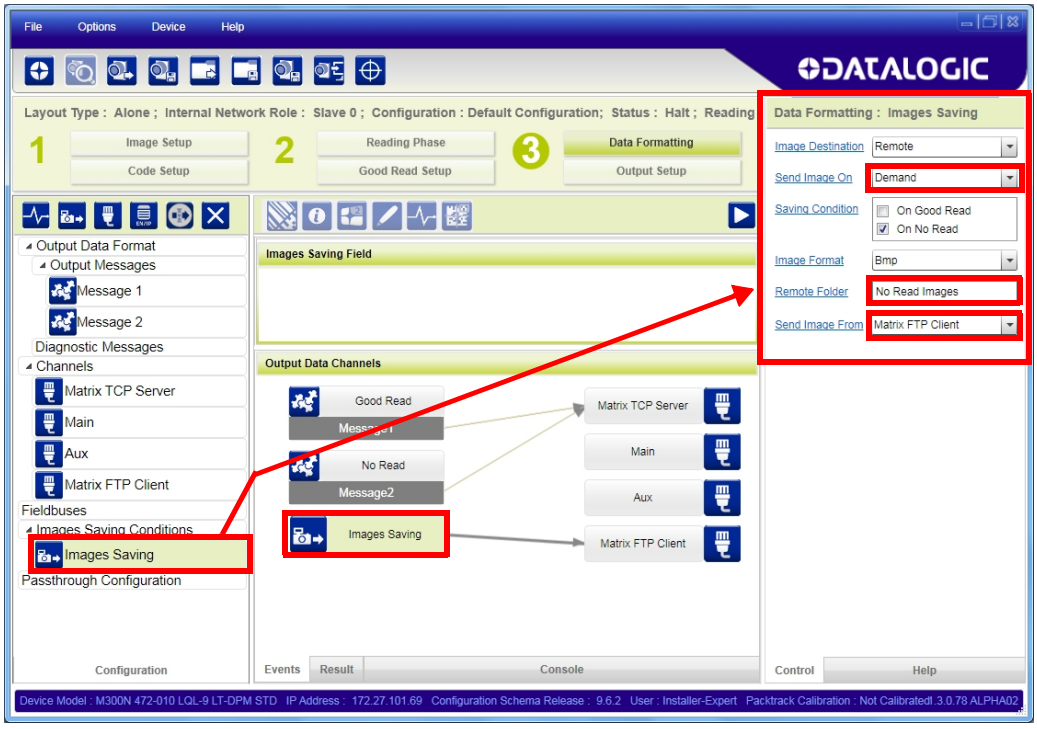


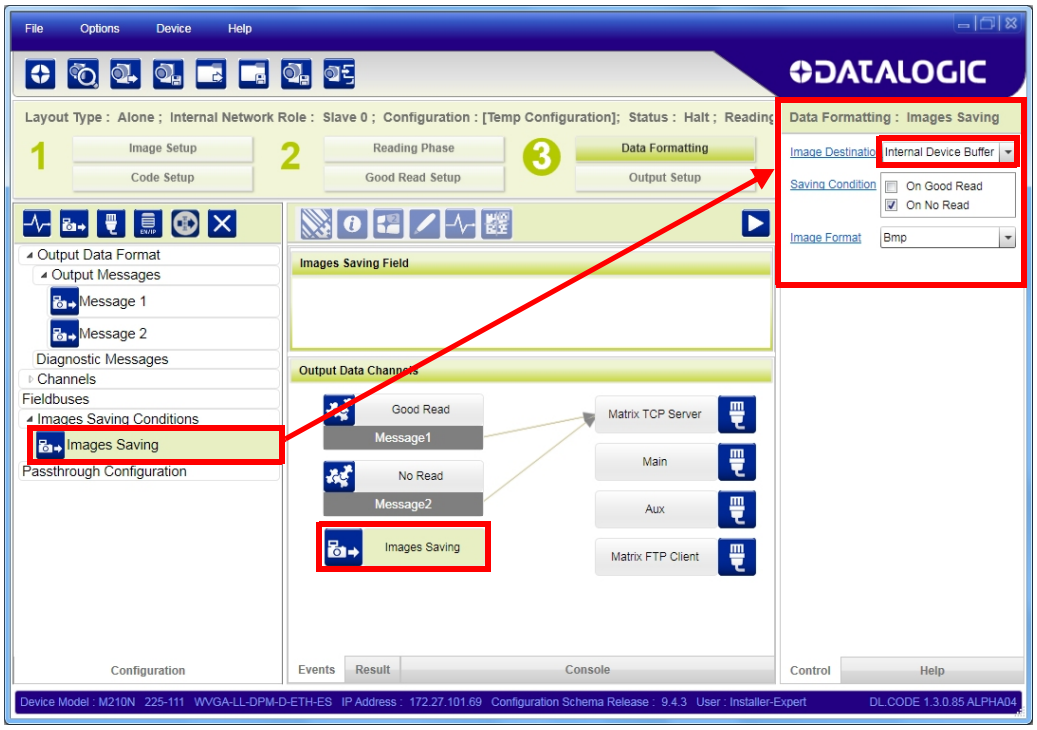
Image Saving On Demand Using Send Images HMP Command

Images can also be sent to a remote destination through a Matrix communication channel (typically to an FTP Server through a Matrix FTP Client channel), upon receiving a **Send Images** command from the Host. See the Matrix N Family Host Mode Programming manual for details on this command.

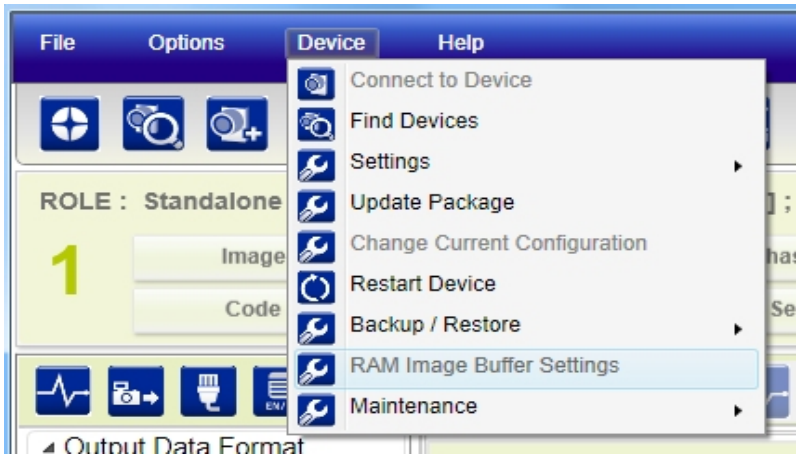
This selection applies only to One Shot and Phase Mode Operating Modes.

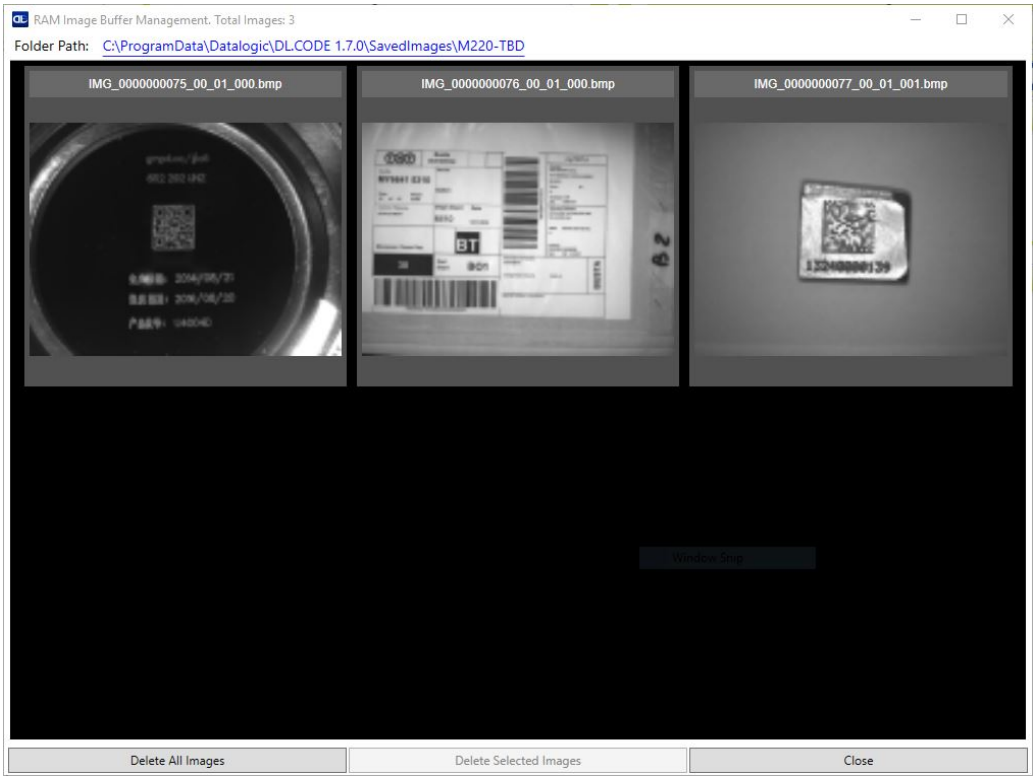
Image Saving Using Internal Buffer

Images can be saved to the Device Image Buffer and be downloaded to a PC when the device is offline.



To download the saved images, the device must be in offline mode. Through the Device menu>RAM Image Buffer you can access the **Image Buffer Management** window.



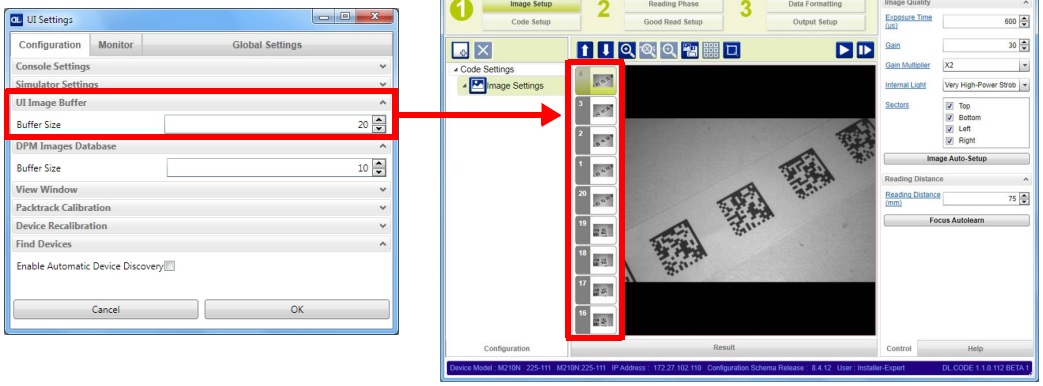


Upon opening this window, the images in the Device Image Buffer (RAM) are transferred to the DL.CODE Device Image Buffer Management window and the images are deleted from the device RAM.

You can save all the images in a zip file to the PC or you can delete the images from the DL.CODE Image Buffer Management memory.

UI IMAGE BUFFER

The UI Image Buffer manages the images captured through the DL.CODE Play and Capture features.



CHAPTER 10
MULTI DEVICE CONFIGURATION OPTIONS

NOTE

DL.CODE now supports several different multi device configuration types using the PASS-THROUGH configuration. In particular this feature allows MULTIDATA ID-NET network configurations to be made. Master/Slave SYNCHRONIZED ID-NET network configurations are also configurable as before.

PASS-THROUGH CONFIGURATIONS

Starting from software version 1.3.0, DL.CODE and Matrix N family readers support pass-through multi device configurations.

The pass-through configuration allows individually working devices (Alone), to collect data from other devices (also working Alone), and pass this data to a third device through a different communication channel. See the figure below as an example.

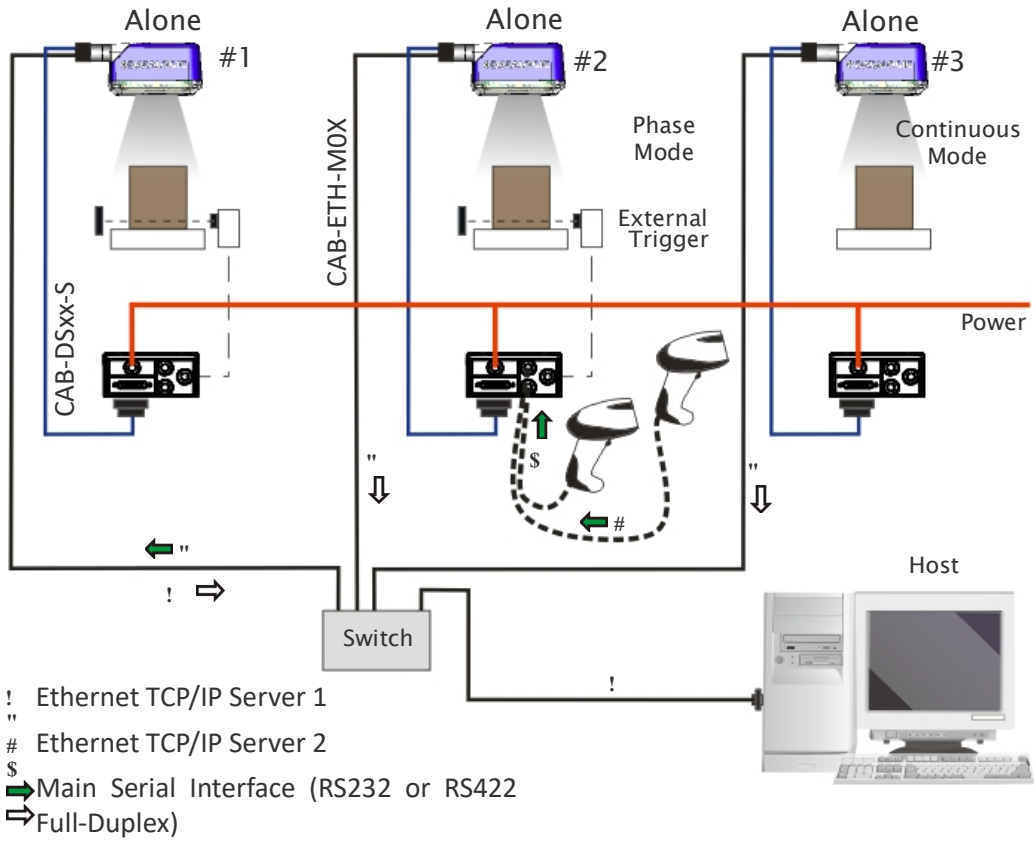
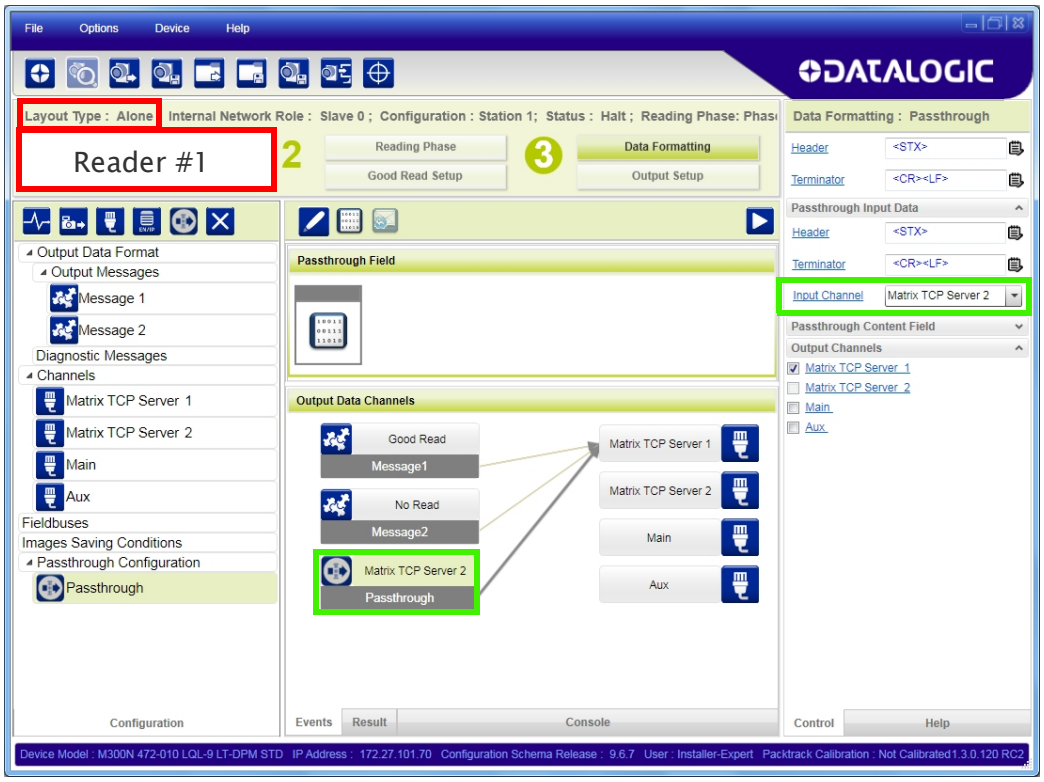
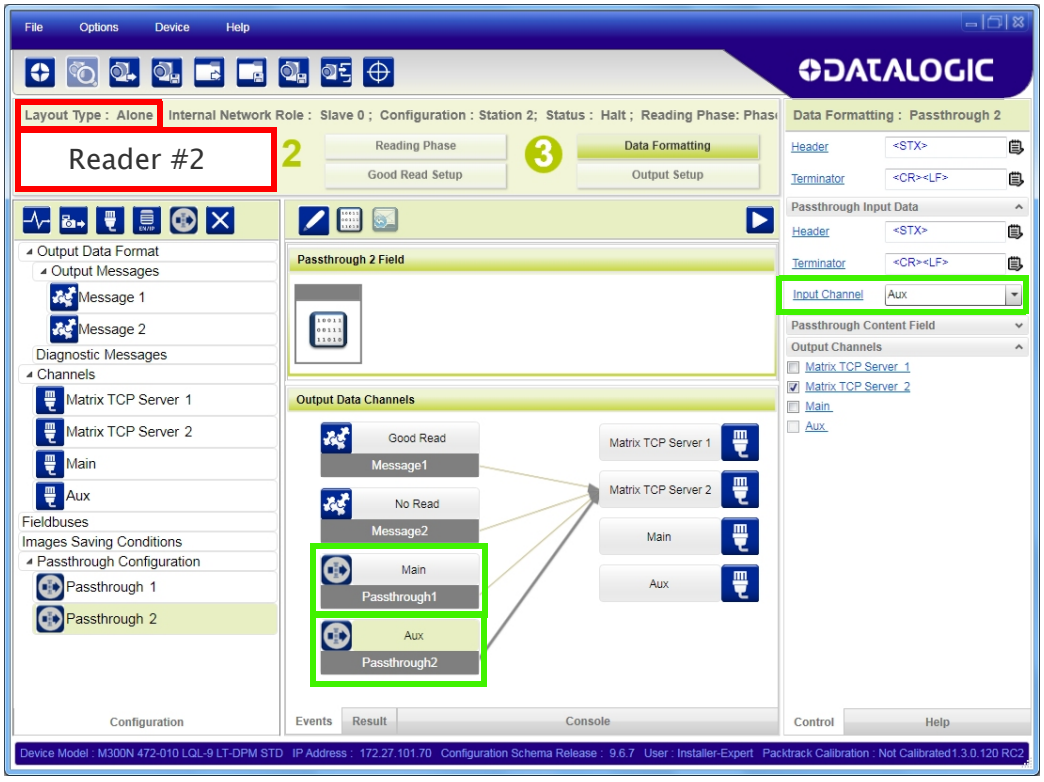
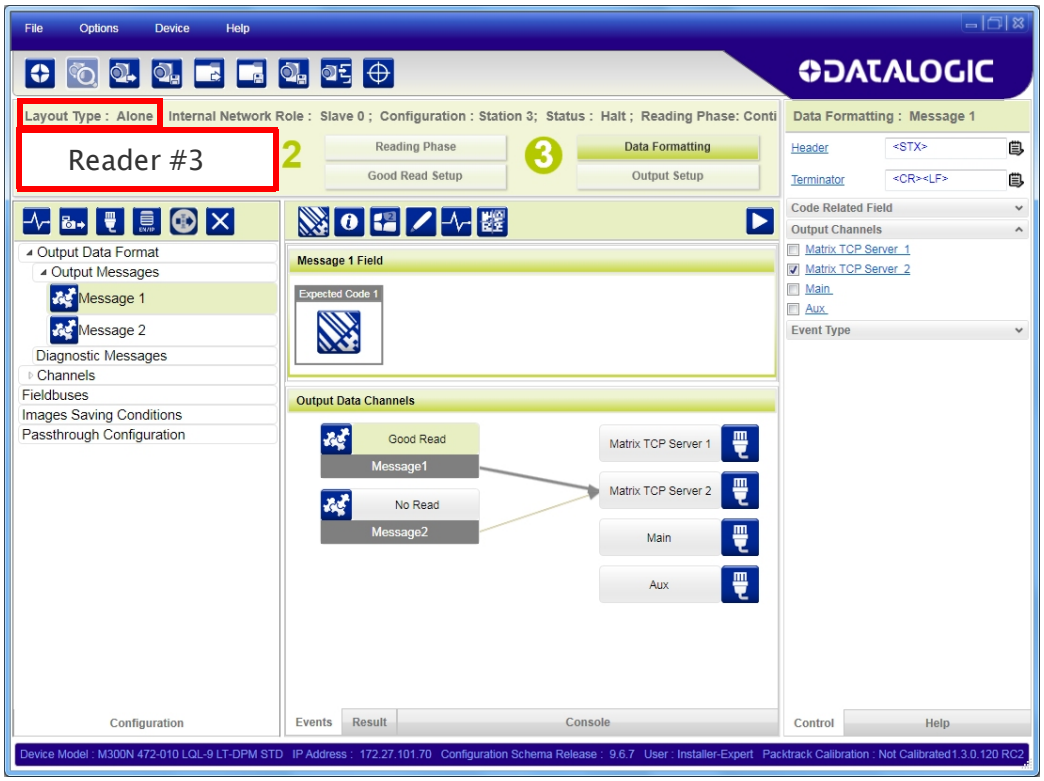


Figure 8 - Example Pass-through Layout

The following screenshots show the configuration settings for the three devices in the example above.







INTERNAL NETWORK CONFIGURATIONS

Internal Network configurations (also called Master/Slave configurations), are designed to collect data from several devices connected together in an ID-NET™ network and send data output to the Host system.

DL.CODE has a Net Autoset feature for the Internal ID-NET Network which automatically recognizes and assigns addresses to all connected Slave readers.

In order to automatically recognize the ID-NET Slaves, all devices must be physically installed and electrically connected (including ID-NET network wiring).

The general procedure is to:

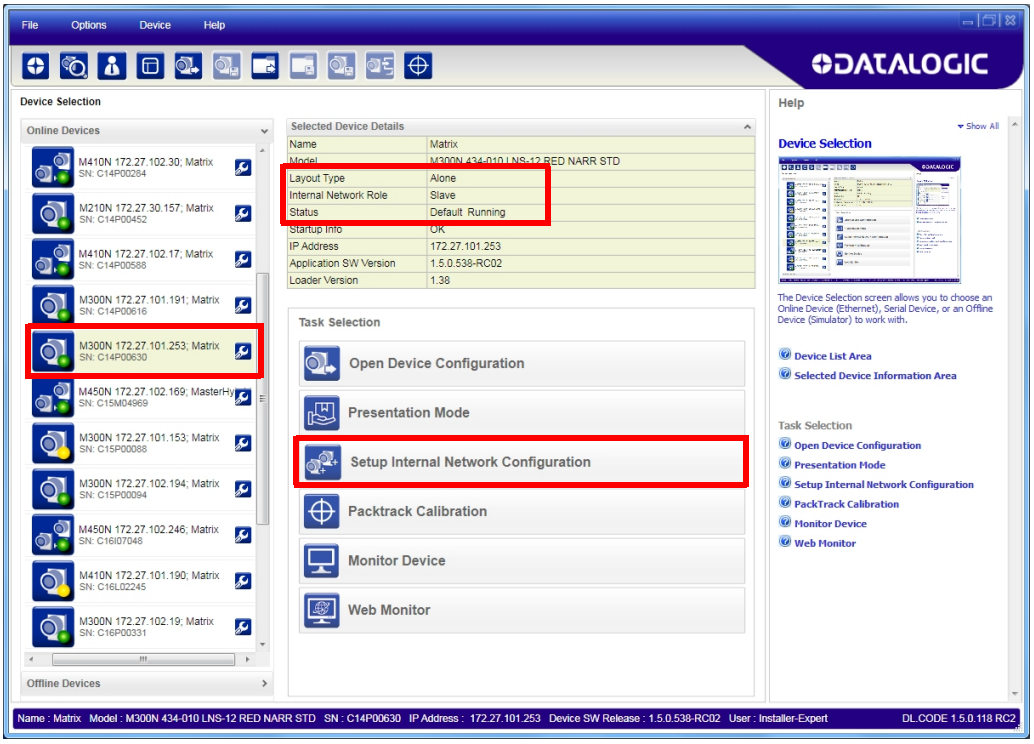
1. Mount all the readers (mechanical and electrical installation) with factory default settings (Layout Type = Alone, Internal Network Role = Slave).

2. Connect to the designated Master device in DL.CODE and open the Setup Internal Network Configuration. You will be prompted to change the device to Master. Click OK. The Slave units will automatically be recognized.

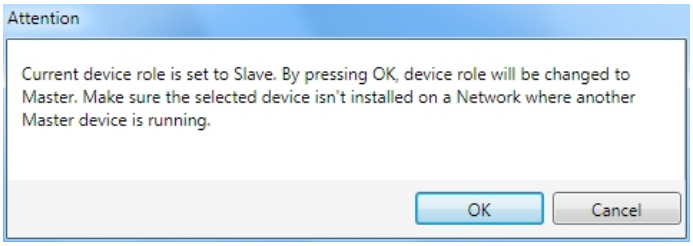
3. Depending on the application, select Multidata, Synchronized Phase Mode or Synchronized PackTrack Configuration.

Master Configuration

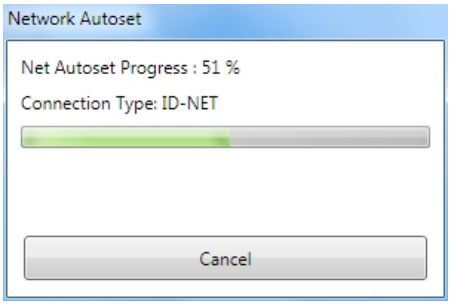
First start with the desired device to assign as ID-NET Master (current default setting is Slave). Click on Setup Internal Network Configuration from the Task area.



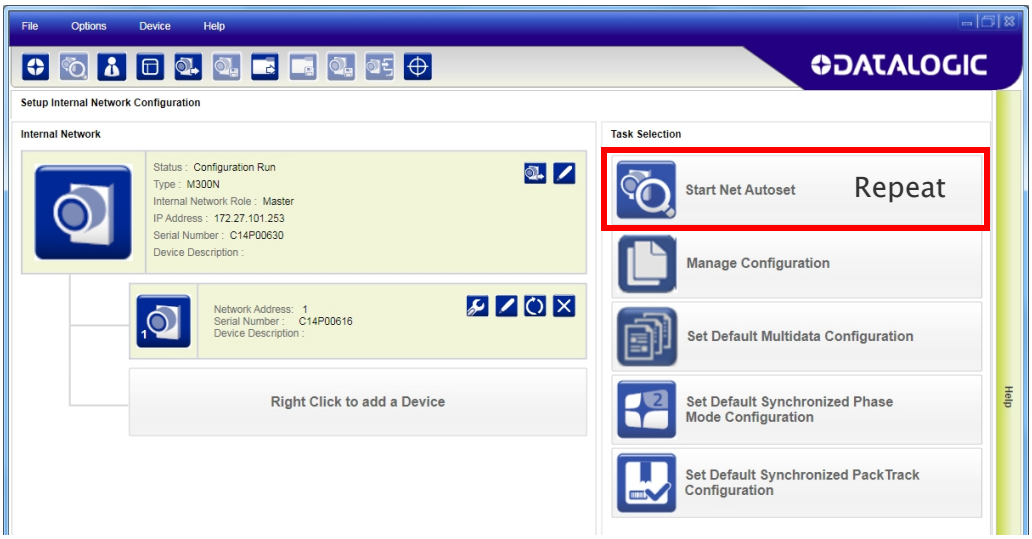
You will be advised that the device role will be changed to Master.



Click OK. The Net Autoset feature automatically starts to find Slave devices connected to the ID-NET network of the Master.

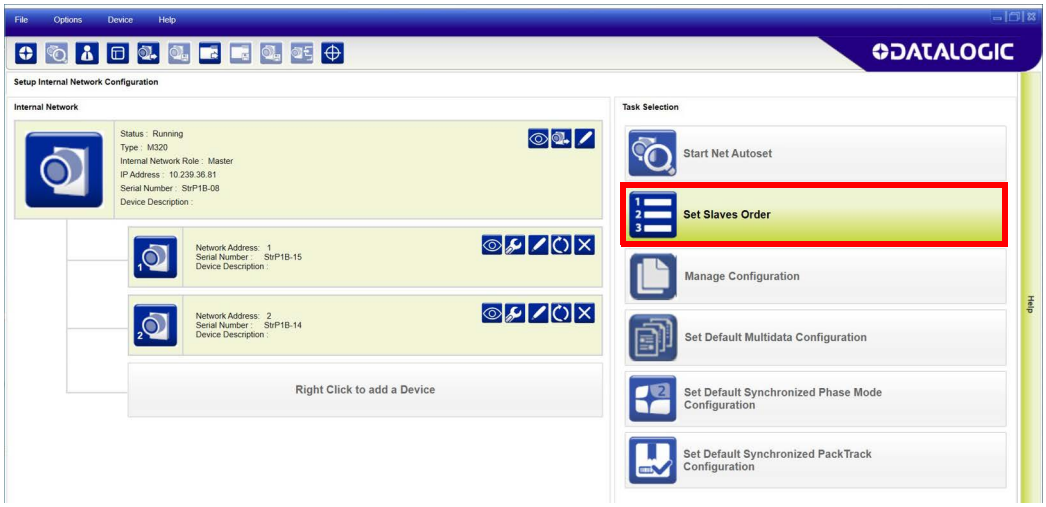


When finished, all the Slaves should have been correctly recognized. If not, verify all device connections and power and then repeat the operation by clicking on the Start Net Autoset button.



Slave Reorder and Wink Check Procedure

Slave devices connected to a Master device in an ID-NET network can be ordered as needed. From the Setup Internal Network Configuration window, click on the Set Slaves Order button.



NOTE

This feature is only available for Online Devices.

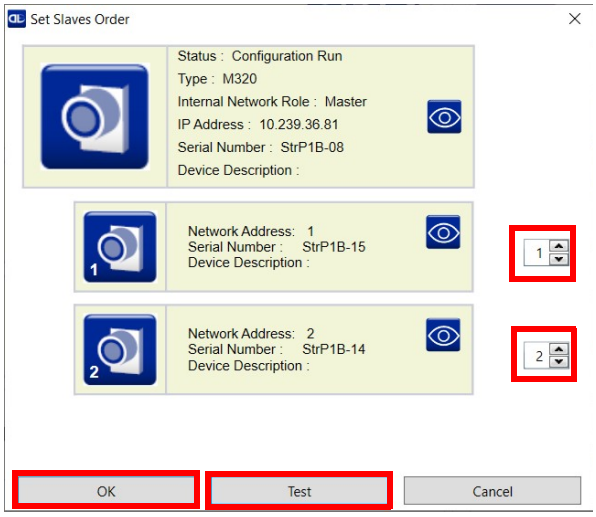
The following window opens:

Click on the up and down arrows next to each Slave device to set their order within the cluster (network address).

Clicking on the Test button all devices will wink in the new set order, starting from the Master device.

Click OK to confirm the new network addresses of the Slave devices.

The OK and Test buttons will be disabled if the network addresses of each Slave device are not unique (range 1-31).



Internal Network Icons

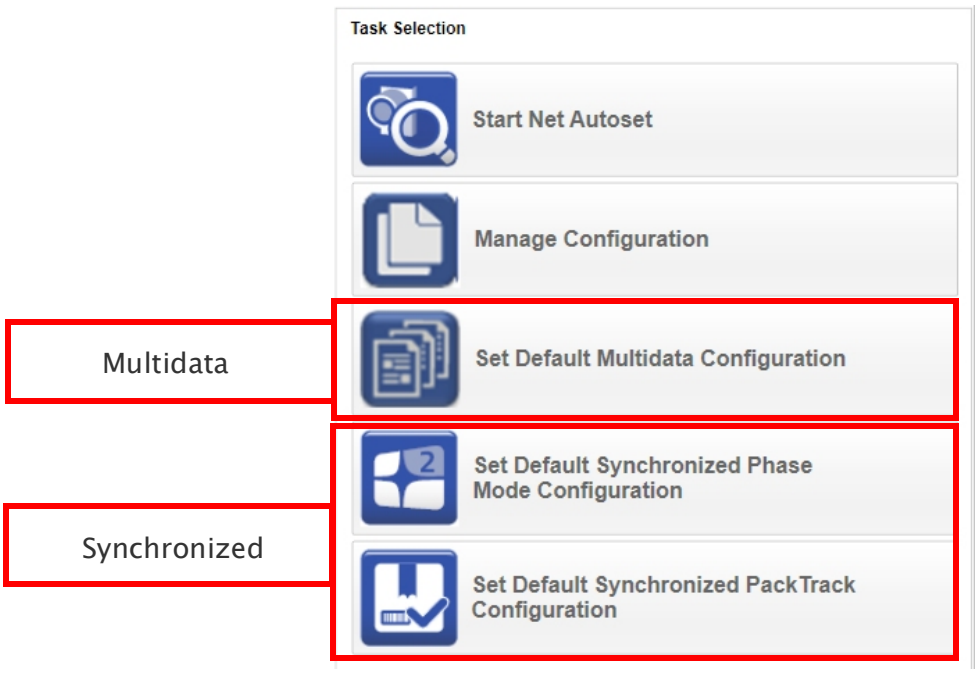
	Open Device Configuration	Go directly to the Master device configuration without setting one of the default network configurations.
	Wink	Wink device (see "Wink and Web Monitor icons" on page 31).
	Edit Device Description	Set a descriptive name for the device in the **internal** network, for example a position name: Top, Left, Right, etc. It will be shown in the Internal Network panel. This is not the same as the Device Name in the Settings menu.
	Edit Environment	Go directly to the Slave device environment parameters window.
	Reset Device	Reset the Slave device.
	Delete Device	Delete the Slave device from the list.

NOTE

The internal network can be pre-configured by right-clicking (and waiting), on the "add a Device" button under the last slave. Placeholder slave device(s) will be added to the list and a new slave network address for each will be assigned to them. When a device is physically revealed, through the Start Net Autoset button, it will be associated with the first placeholder in the list.

While the entire internal network can be pre-configured, each device must be physically added to the network one-at-a-time in order to be correctly recognized by the network.

Depending on the application, select one of the Default Internal Network Configurations: **Multidata**, **Synchronized Phase Mode** or **Synchronized PackTrack**.



This selection will open a pre-configured job for the Master reader according to the selection. Follow the specific application instructions in the following paragraphs.

Multidata ID-NET Network Configurations

The Multidata ID-NET network communications between Master and Slave are managed by the application job (configuration) using the pass-through feature. A pre-configured job is loaded with the correct pass-through settings for both the Master and Slaves when the Default Multidata Configuration is selected from the Internal Network Setting feature.

1. Complete the configuration of all the application parameters (including Image Settings) and save them to the Master with an application specific name. Optionally, checking the **Save on Slave Device** box can be helpful to save all the current individual Slave configurations with the new configuration *name*. This does not clone any parameters. If not checked, Slave configurations will remain as *Temp* configurations and you will be warned that changes to the Master have not been saved to the Cluster. For Multidata configurations, the option to **Clone Master configuration on Slaves must not be checked**.

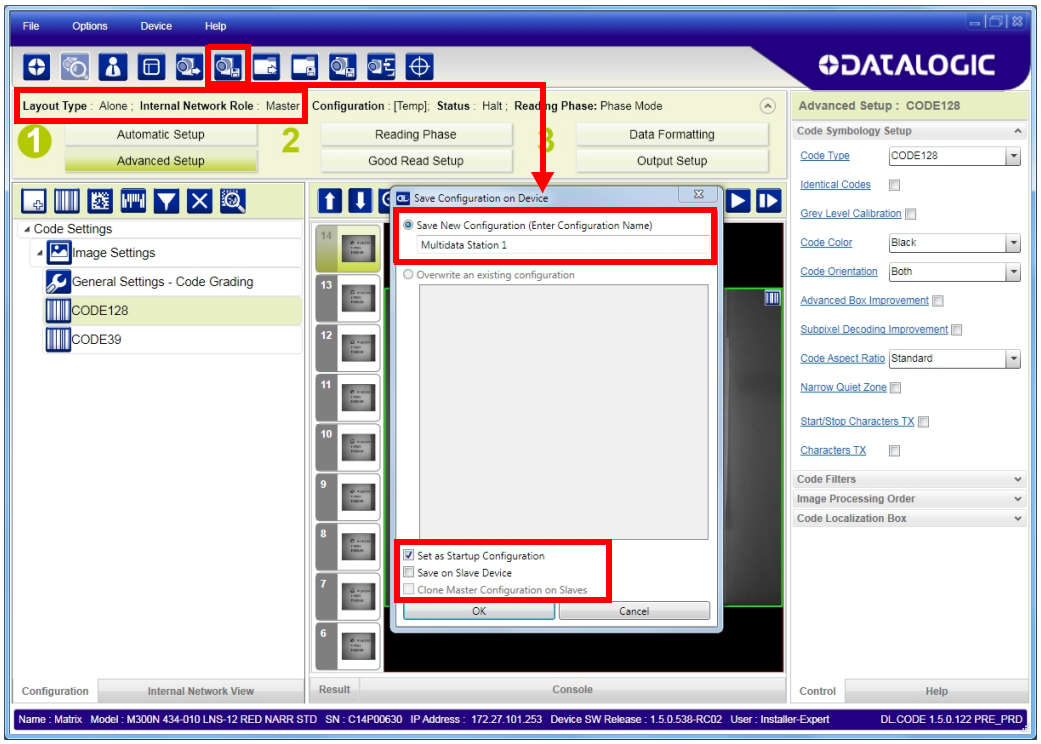


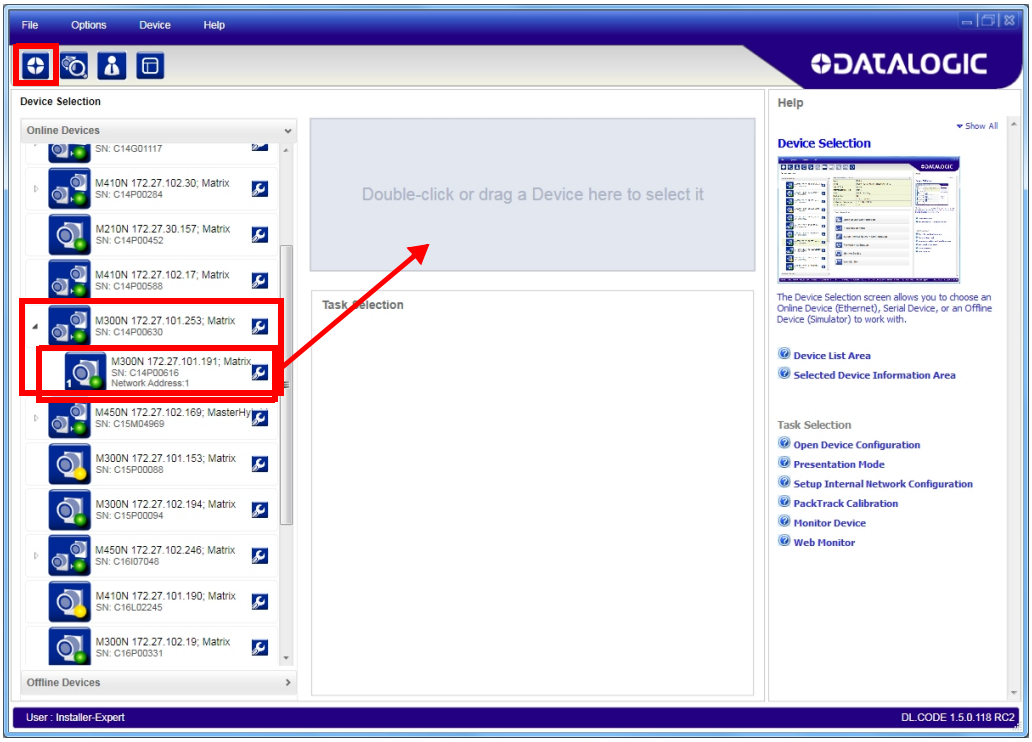
Figure 9 - Saving Multidata Configuration to Master

The jobs must not be cloned because the Master and Slaves have different input/output communication channels. The readers are also working independently from each other, often on separate stations with different code reading requirements, different operating modes, etc.

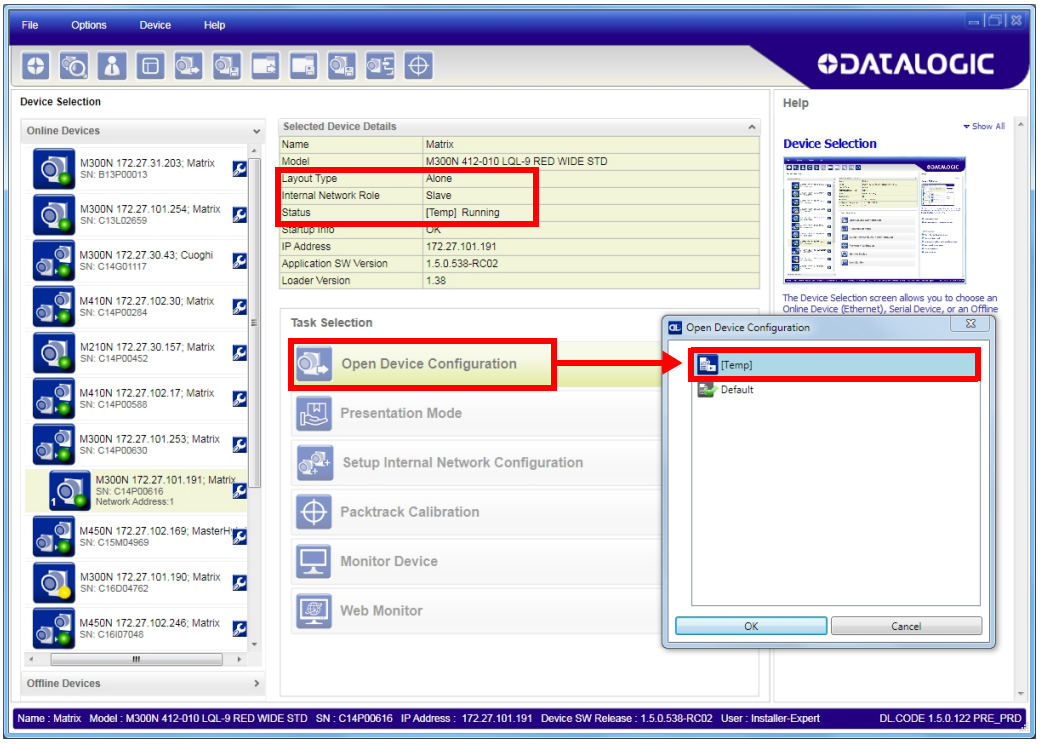
2. Connect to each Slave reader via Ethernet (see note below), and set all the configuration parameters of each Slave device.

NOTE

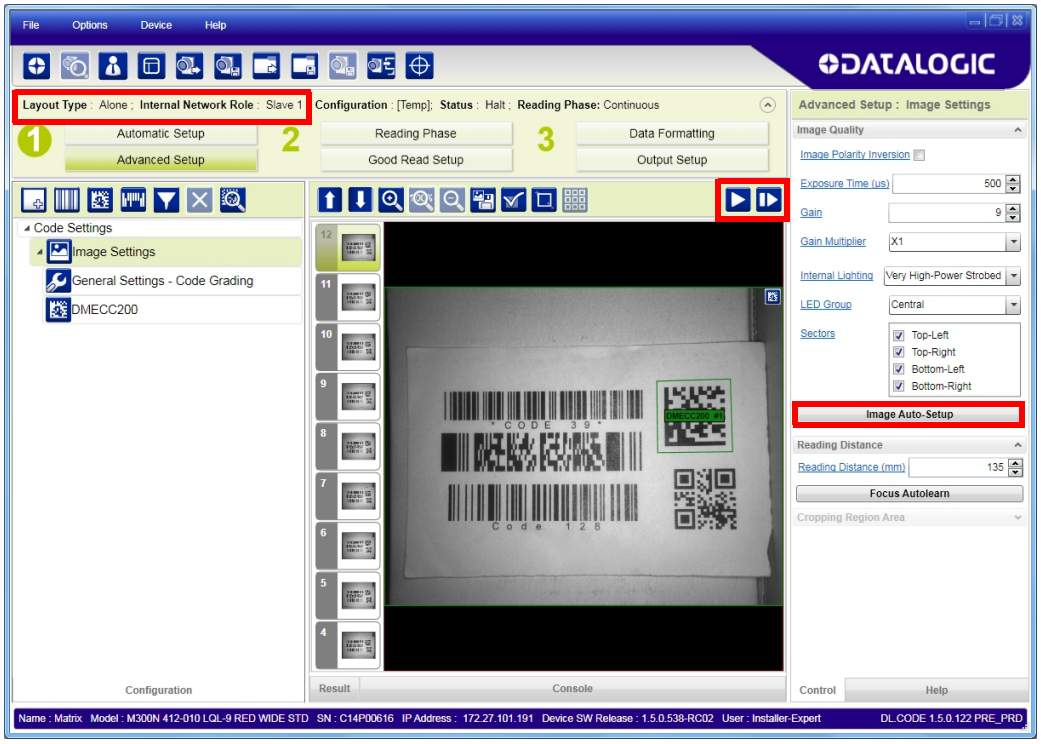
If necessary, Slave device photometric (Image Settings) parameters must be configured separately through DL.CODE. This is preferably done through each device's Ethernet TCP/IP channel. If Slave devices are not connected to Ethernet you must temporarily (manually) connect them one by one to perform Image Settings.



Open the Slave specific application job, (it will either have the new name saved from the Master or Temp depending on the **Save on Slave Device** selection).



When the configuration opens, pause run mode and set all the application specific configuration parameters (including Image Settings).



Verify the focus and decoding with the capture image function.

3. Now save them to a new Slave specific application job name[1].

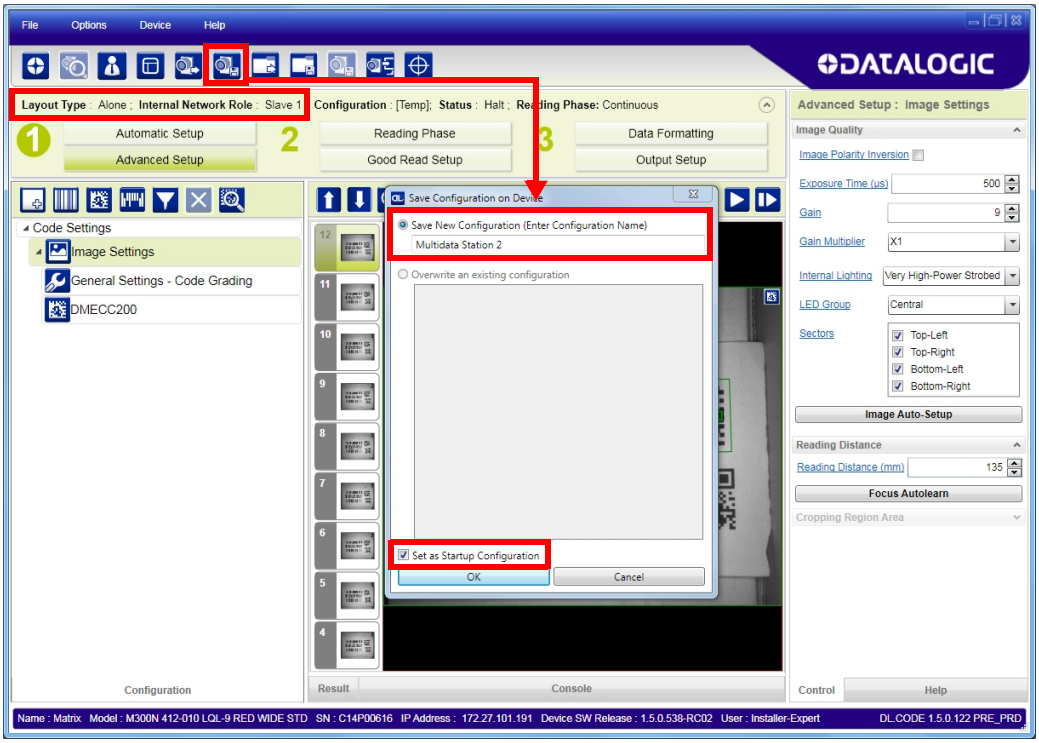


Figure 10 - Saving Multidata Configuration to Slave 1

Repeat this procedure for each Slave device until the entire network is configured.

1. If **Save on Slave Device** was selected when saving the Master configuration, an application job with the same name (but with all Slave specific configuration parameters), has been saved to the Slaves. No parameters have been cloned from the Master. There are no common parameters managed by the Master for Multidata configurations.

Example Multidata ID-NET Configuration

The Multidata ID-NET network takes advantage of the pass-through configuration to allow all the connected readers to work independently from each other (Layout Type = Alone).

In this way data is collected over the ID-NET network and passed–through the Master to the Host system on a different communication channel. See the figure below as an example.

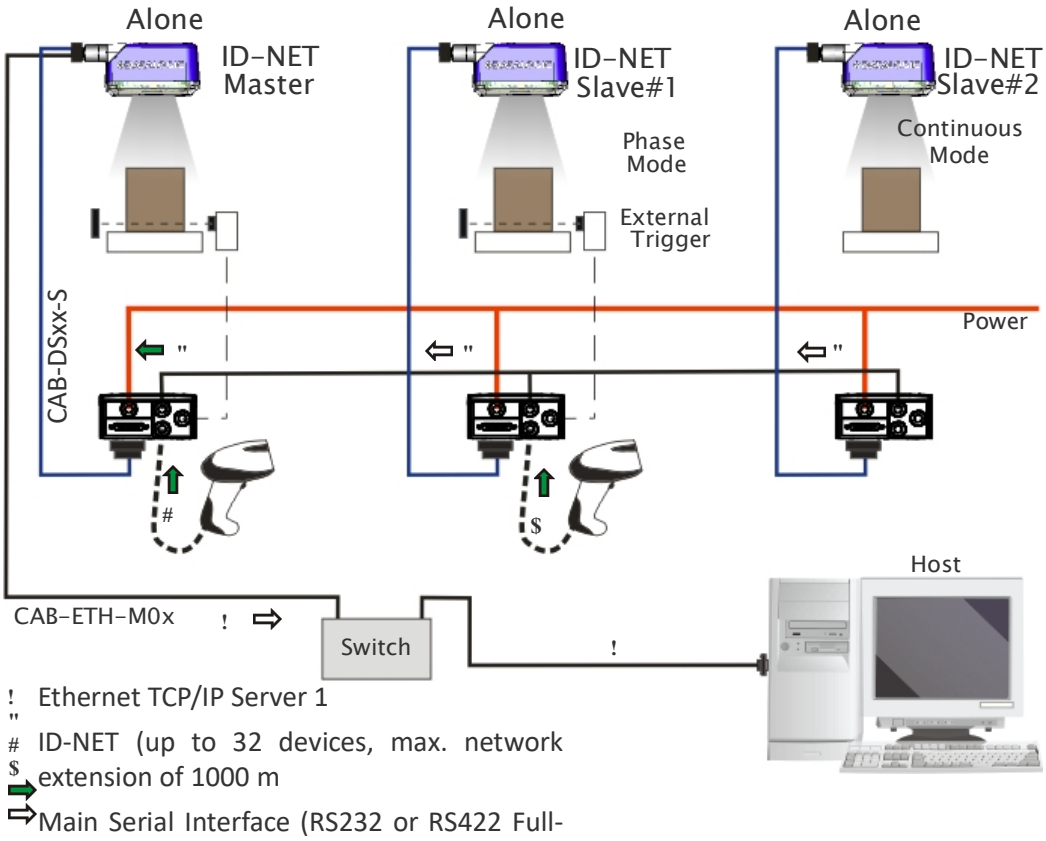
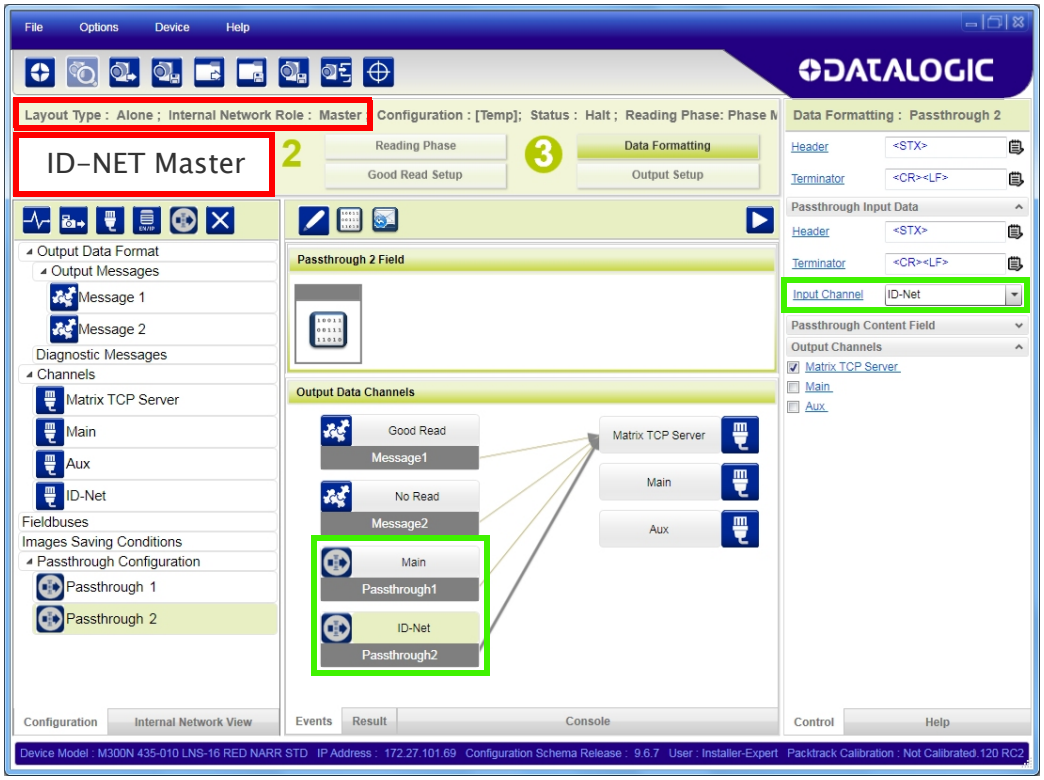
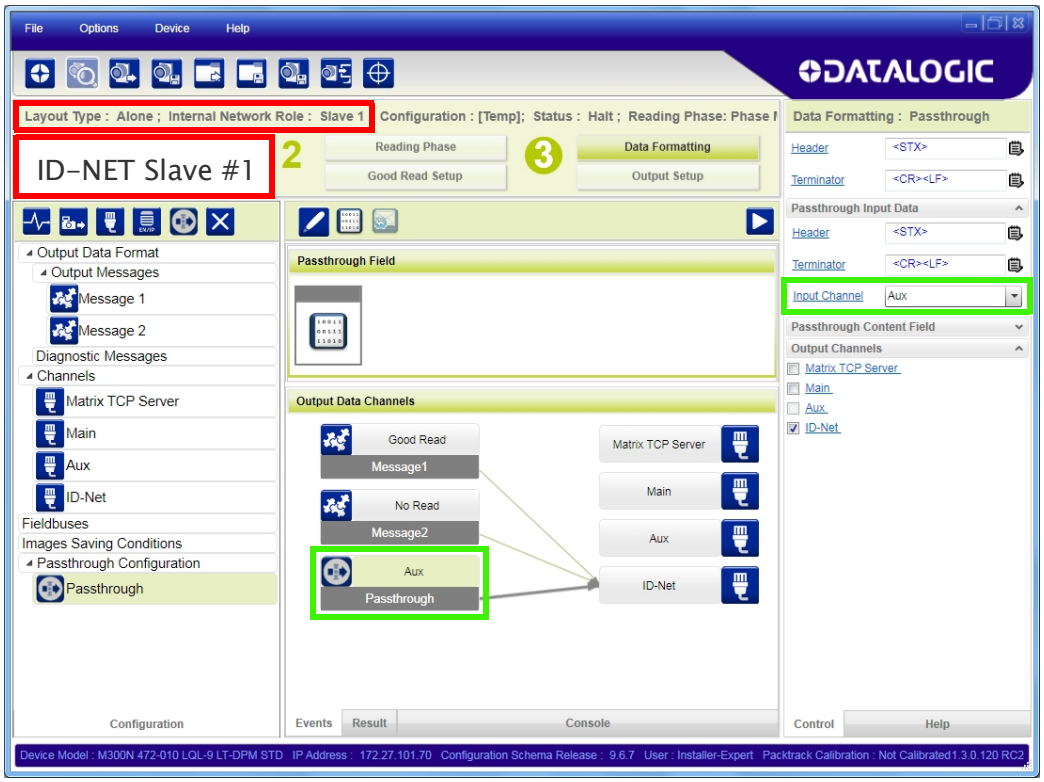
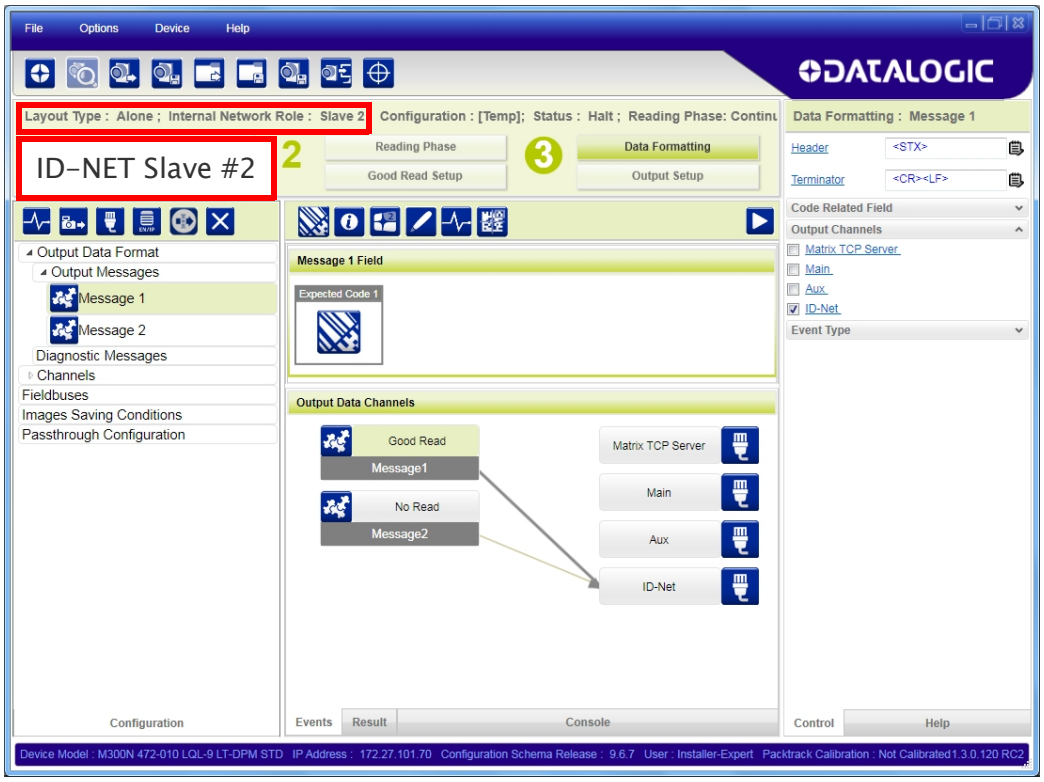


Figure 11 - ID-NET Multidata (Pass-through)

The following screenshots show the pass-through configuration settings for the three devices in the example above.


File
Options
Device
Help
DATALOGIC
Layout Type : Alone ; Internal Network Role : Master
Configuration : [Temp]; Status : Halt ; Reading Phase: Phase N
ID-NET Master
2
Reading Phase
Good Read Setup
3
Data Formatting
Output Setup
Data Formatting : Passthrough 2
Header
<STX>
Terminator
<CR><LF>
Passthrough Input Data
Header
<STX>
Terminator
<CR><LF>
Input Channel
ID-Net
Passthrough Content Field
Output Channels
Matrix TCP Server
Main
Aux
Output Data Format
Output Messages
Message 1
Message 2
Diagnostic Messages
Channels
Matrix TCP Server
Main
Aux
ID-Net
Fieldbuses
Images Saving Conditions
Passthrough Configuration
Passthrough 1
Passthrough 2
Passthrough 2 Field
Output Data Channels
Good Read
Message1
No Read
Message2
Main
Passthrough1
ID-Net
Passthrough2
Matrix TCP Server
Main
Aux
Configuration
Internal Network View
Events
Result
Console
Control
Help
Device Model : M300N 435-010 LNS-16 RED NARR STD IP Address : 172.27.101.69 Configuration Schema Release : 9.6.7 User : Installer-Expert Packtrack Calibration : Not Calibrated.120 RC2


File
Options
Device
Help
DATALOGIC
Layout Type : Alone ; Internal Network Role : Slave 1
Configuration : [Temp]; Status : Halt ; Reading Phase: Phase I
ID-NET Slave #1
2
Reading Phase
Good Read Setup
3
Data Formatting
Output Setup
Data Formatting : Passthrough
Header
<STX>
Terminator
<CR><LF>
Passthrough Input Data
Header
<STX>
Terminator
<CR><LF>
Input Channel
Aux
Passthrough Content Field
Output Channels
Matrix TCP Server
Main
Aux
ID-Net
Output Data Format
Output Messages
Message 1
Message 2
Diagnostic Messages
Channels
Matrix TCP Server
Main
Aux
ID-Net
Fieldbuses
Images Saving Conditions
Passthrough Configuration
Passthrough
Passthrough Field
Output Data Channels
Good Read
Message1
No Read
Message2
Aux
Passthrough
Matrix TCP Server
Main
Aux
ID-Net
Configuration
Events
Result
Console
Control
Help
Device Model : M300N 472-010 LQL-9 LT-DPM STD IP Address : 172.27.101.70 Configuration Schema Release : 9.6.7 User : Installer-Expert Packtrack Calibration : Not Calibrated1.3.0.120 RC2


File
Options
Device
Help
DATALOGIC
Layout Type : Alone ; Internal Network Role : Slave 2
Configuration : [Temp]; Status : Halt ; Reading Phase: Continu
ID-NET Slave #2
2
Reading Phase
Good Read Setup
3
Data Formatting
Output Setup
Data Formatting : Message 1
Header
<STX>
Terminator
<CR><LF>
Code Related Field
Output Channels
Matrix TCP Server
Main
Aux
ID-Net
Event Type
Output Data Format
Output Messages
Message 1
Message 2
Diagnostic Messages
Channels
Fieldbuses
Images Saving Conditions
Passthrough Configuration
Message 1 Field
Expected Code 1
Output Data Channels
Good Read
Message1
No Read
Message2
Matrix TCP Server
Main
Aux
ID-Net
Configuration
Events
Result
Console
Control
Help
Device Model : M300N 472-010 LQL-9 LT-DPM STD IP Address : 172.27.101.70 Configuration Schema Release : 9.6.7 User : Installer-Expert Packtrack Calibration : Not Calibrated1.3.0.120 RC2

Synchronized ID-NET Network Configurations

The Synchronized ID-NET network communications between Master and Slave are internally managed by the application software. A pre-configured job is loaded with the Synchronized Layout Type and the correct Operating Mode for both the Master and Slaves when either the Phase Mode or PackTrack Configuration is selected from the Internal Network Setting feature.

1. Complete the configuration of all the application parameters (including Image Settings) and save them to the Master with an application relative name and **with** the option to **Clone Master configuration on Slaves**.

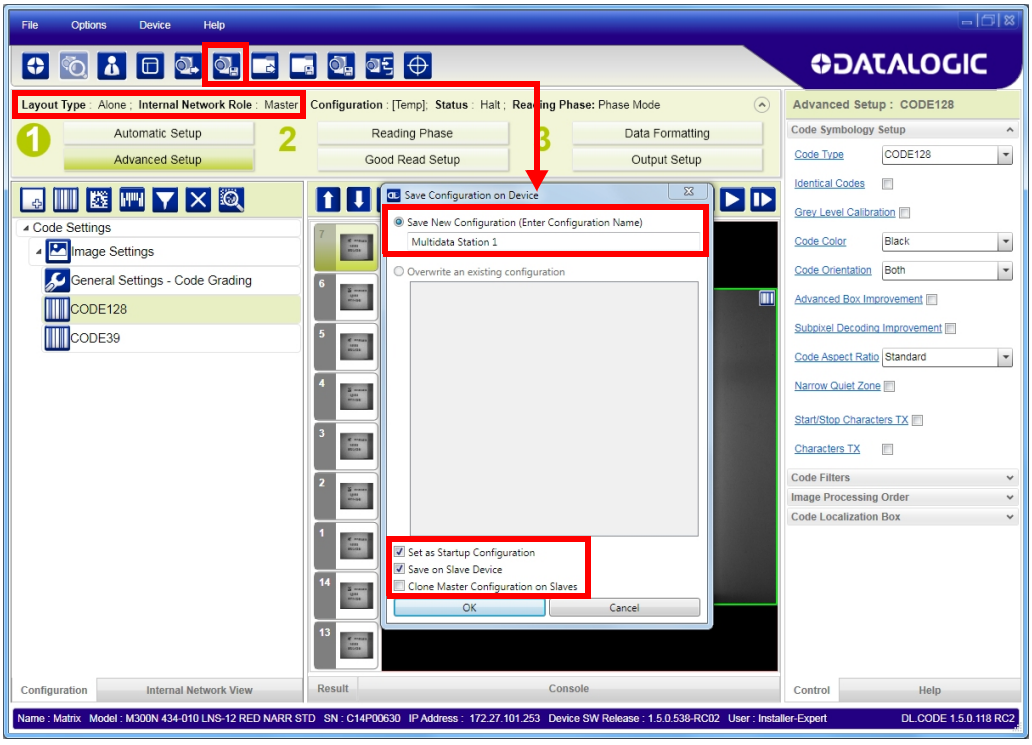
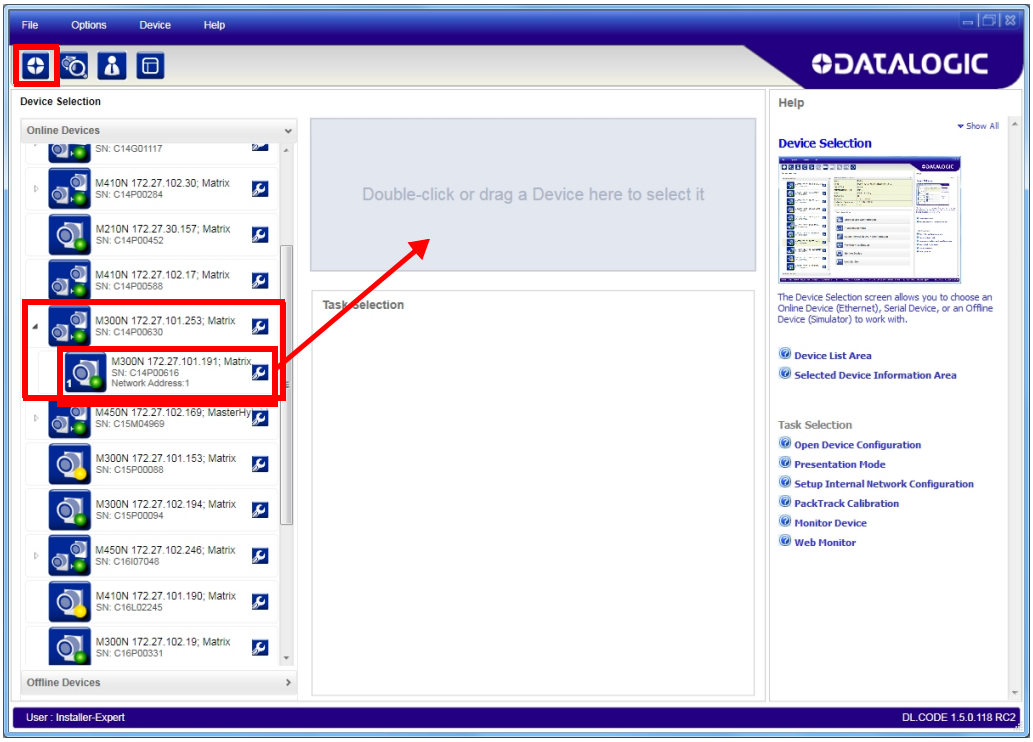


Figure 12 - Saving Synchronized Phase Mode Configuration to Master

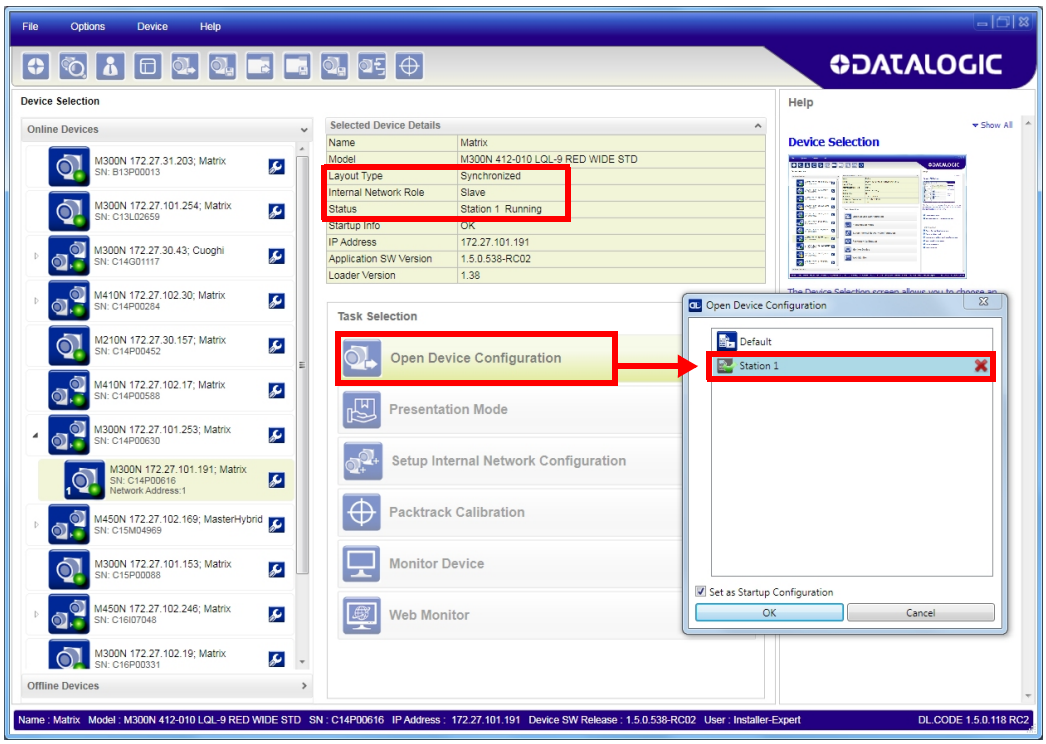
2. Connect to each Slave reader via Ethernet (see note below), and set the Slave specific parameters.

NOTE

If necessary, Slave device photometric (Image Settings) parameters must be configured separately through DL.CODE. This is preferably done through each device's Ethernet TCP/IP channel. If Slave devices are not connected to Ethernet you must temporarily (manually) connect them one by one to configure Image Settings.



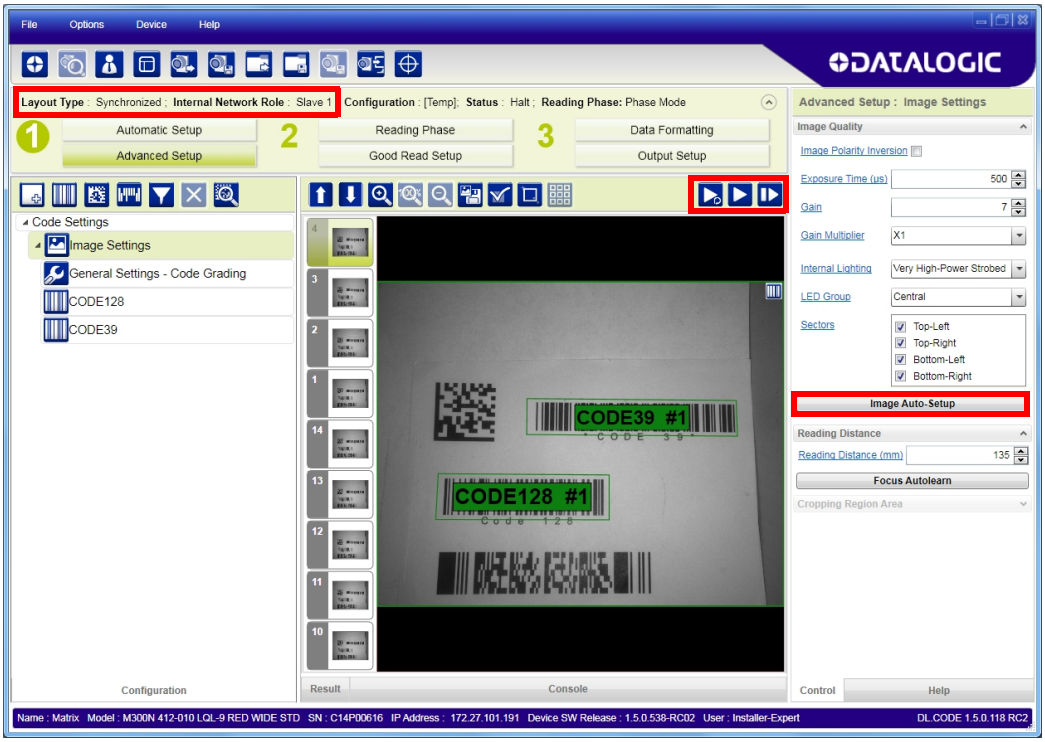
Open the cloned application job.



When the job opens, pause run mode and configure the Slave specific parameters. These depend on the application and include the following:

- photometric parameters (Image Auto-Setup feature in the Advanced Setup - Image Settings step)
- Acquisition Trigger Delays necessary to avoid lighting interference between adjacent or oppositely positioned readers (Reading Phase step)
- Images Saving if used (Data Formatting step)

- Encoder Sensor: if used, (for all Slaves, the Encoder Type must be set to **Internal**)



Verify the focus and decoding with the capture image function.

3. Now save them overwriting the cloned application job[1].

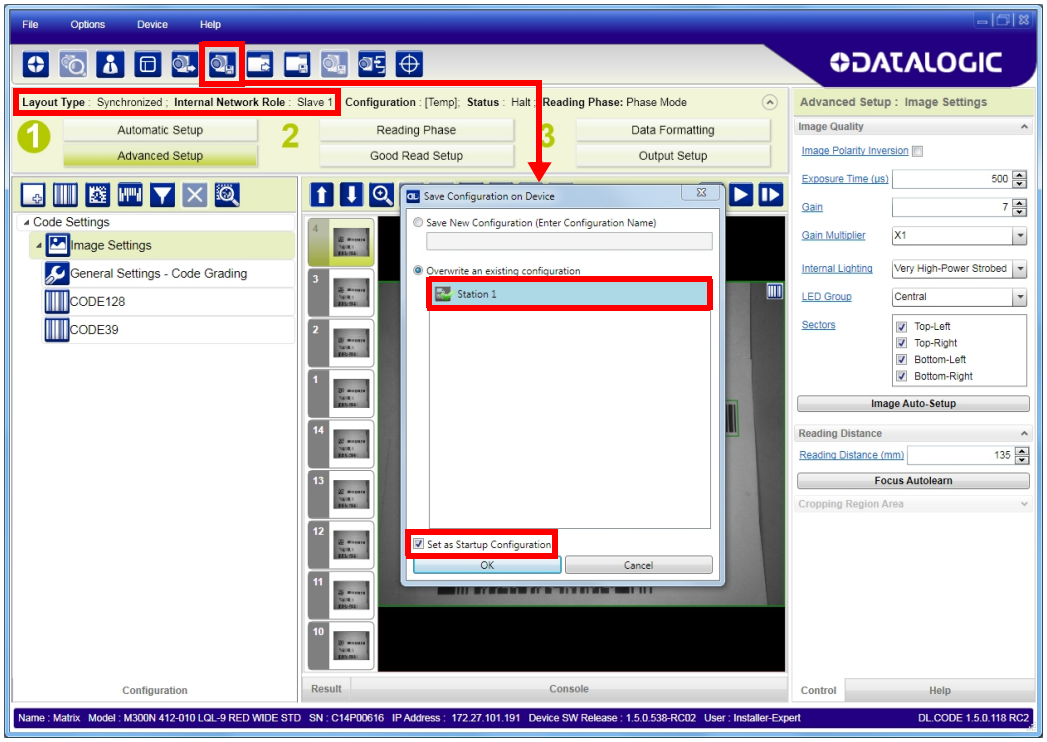


Figure 13 - Saving Synchronized Phase Mode Configuration to Slave

Repeat this procedure for each Slave device until the entire network is configured.

1. An application job with the same name as the Master's has been cloned to the Slaves. Each Slave can have its own Image Settings parameters saved in its own copy of the application job. Common parameters managed by the Master such as Operating Mode cannot be modified in the Slave jobs and are shown in dark gray.

Example Synchronized ID-NET Configuration

When the device is working in a **Synchronized** Layout Type, the ID-NET connection is used to collect data from several readers to build a multi-point or a multi-sided reading system; there can be one Master and up to 31 Slaves connected together.

The Slave readers are connected together using the ID-NET interface. Every slave reader must have an ID-NET address in the range 1-31.

The Master reader is also connected to the Host on one of its communication channels. In the following example the TCP/IP on-board Ethernet interface is used.

For a Master/Slave Synchronized layout the External Trigger signal is unique to the system; there is a single reading phase and a single message from the Master reader to the Host computer. **It is not necessary to bring the External Trigger signal to all the readers.**

In the Master/Slave Synchronized layout the Master operating mode can only be set to PackTrack or Phase Mode.

The TCP/IP and ID-NET interfaces are connected as shown in the following figures.

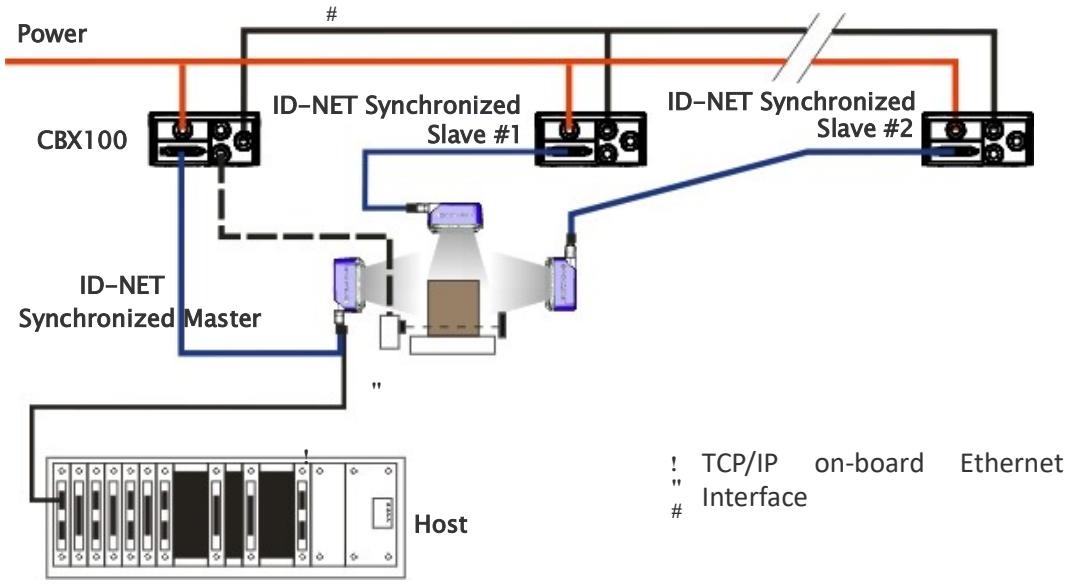


Figure 14 - Example ID-NET Synchronized Layout with Master on-board TCP/IP Ethernet Interface to Host

The Master reader can be connected to the CBX series connection box with the advantage of the Backup and Restore configuration function (CBX + BM100 module).

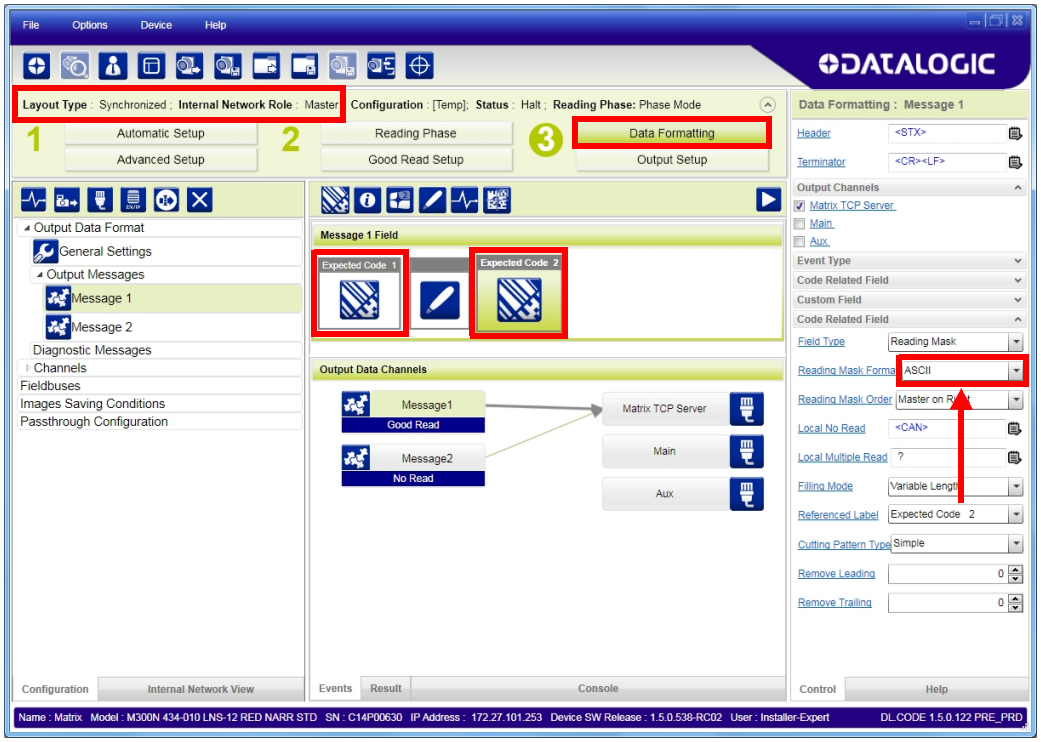
All devices always support multiple output channels (i.e. for data monitoring).

Verify Master/Slave Synchronized Configuration

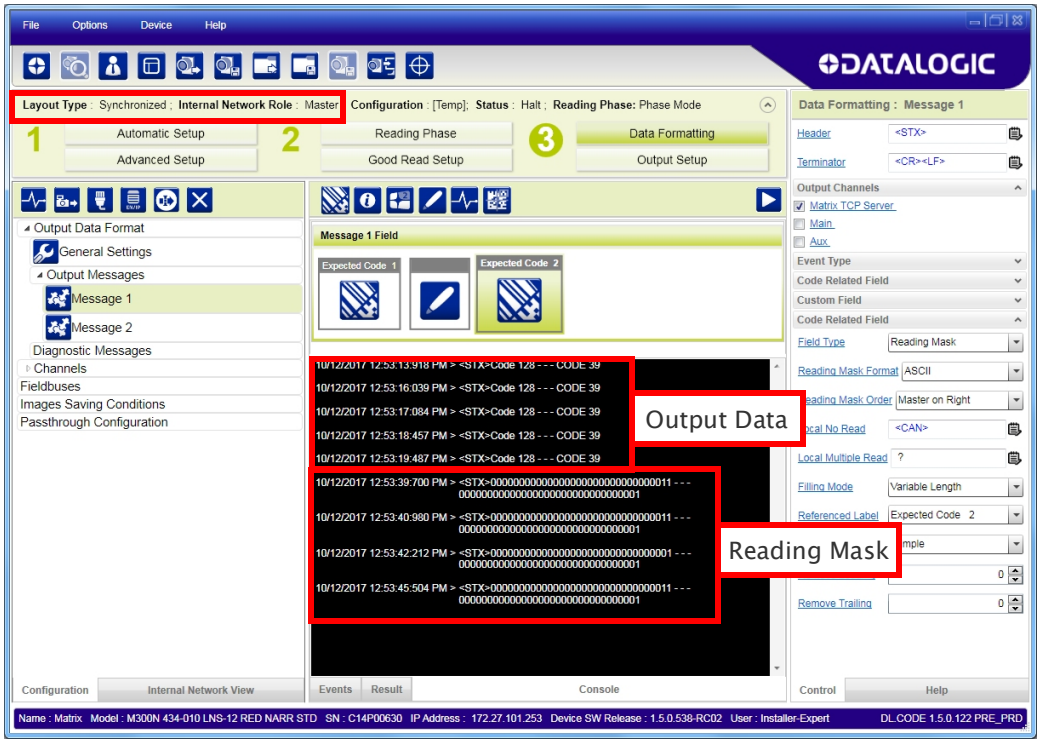
From the Master configuration, run the application and monitor the output data from the DL.CODE Console or a configured channel terminal.

If necessary, as a troubleshooting tip, you can temporarily apply the **Reading Mask** field in place of each Code Content field to verify if all devices are reading. To do this:

1. Connect to the Master device via Ethernet and from the Data Formatting step, change each Expected Code Field Type from Code Content to Reading Mask.



2. Run the application and monitor the output data from the DL.CODE Console or a configured channel terminal.



The Reading Mask shows which device reads which Expected Code. The mask is composed of a fixed 32-character string (0=No Read or 1=Read) representing the 32 possible readers in an ID-NET network. By default the Master is the last character in the string (**Master on Right**) but this can be changed. The Slaves are shown adjacent to the Master in order (1 to 31), by default from right to left.

Example: Master + Slave 1 = 00000000000000000000000000000011

The figure above shows that both the Master and Slave 1 are reading Code 128 while only the Master is reading Code 39.

3. After verifying correct functioning of the reading devices, return the Expected Code fields from Reading Mask to Code Content.

4. If you haven't made any other changes you can exit without saving. Otherwise, save the Master device configuration overwriting its previous one, making sure to save **without** **Clone Master Configuration on Slaves**, otherwise the Slave configurations will be overwritten.

To view the connected Slave configurations:

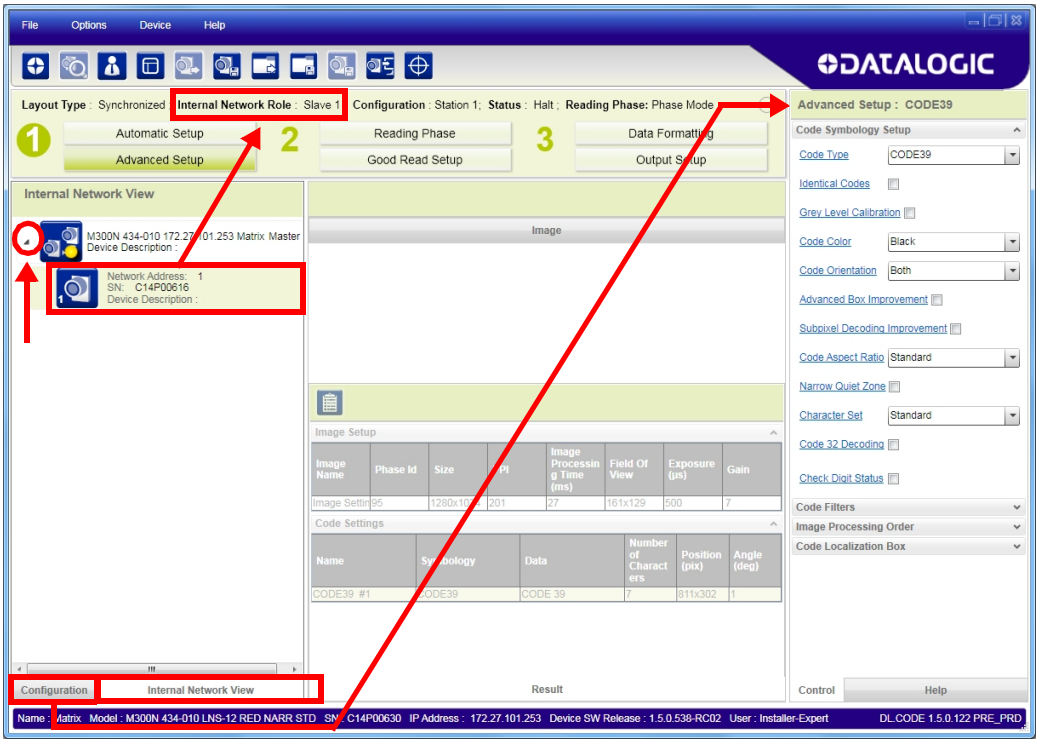
1. Click on the Internal Network View tab at the bottom of the screen

2. Open the Master branch by clicking on the arrow to the left of the Master icon.

3. Select any slave. Wait for the configuration to load.

4. Click The Configuration tab at the bottom of the screen.

By selecting the various configuration steps above you can visualize the slave configuration.



NOTE

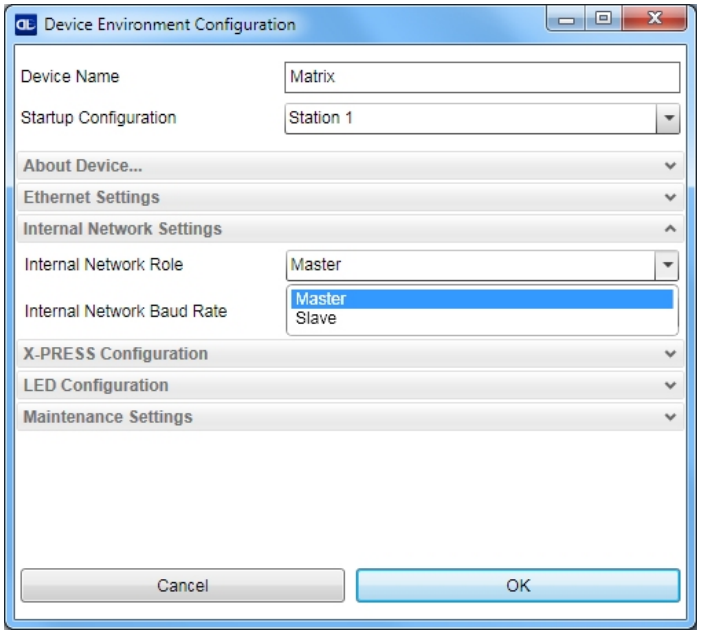
You can modify some Slave Synchronized parameters from this view but you cannot save them here.

To save changed slave parameters, you must click on the Master and Save the configuration overwriting it, making sure to select Save on Slave Device but without Clone Master Configuration on Slave, otherwise all the Slave configuration parameters will be overwritten by the Master configuration.

Alternative Device Role Selection

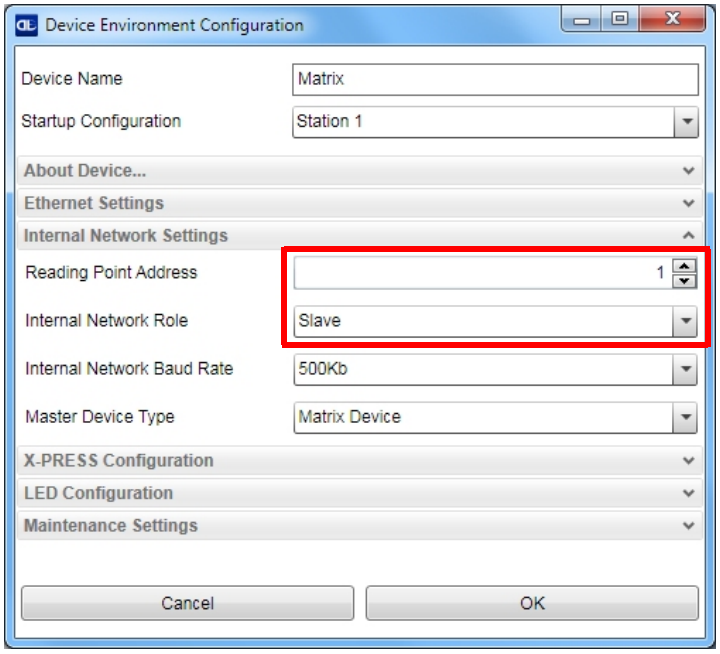
To set up a Master/Slave Internal Network Role you can also enter the Device Environment settings from the Device menu>Settings>Settings and open the Internal Network Settings group.

Change the Internal Network Role to Master or Slave accordingly.



Wait for the device to reset. It may be necessary to wait several seconds and then perform a Discovery to refresh the device list area. You should now see the Master device with its relative icon.

Likewise, you can set the device **Internal Network Role** to Slave and the **Reading Point Address** (ID-NET address) according to the network configuration.



Complete the Slave configuration and save it to the device.

Complete all the ID-NET Slaves in the same way.

The Master must be set to expect the same number of slaves as foreseen by the application. Now by simply making the electrical connections the network will be automatically recognized upon power up.

CHAPTER 11
PACKTRACK CALIBRATION

PackTrack Calibration can be performed on all Standalone or Master Matrix N family readers that support this feature.

OVERVIEW

PackTrack Operating Mode is a method used to correctly assign codes read to their corresponding parcel or pack in systems where multiple packs are simultaneously present in the reading area.

The figure below illustrates the main concepts defining a PackTrack system. The main hardware parts are: conveyor, Matrix N reader(s), encoder (tach), and presence sensor (photocell).

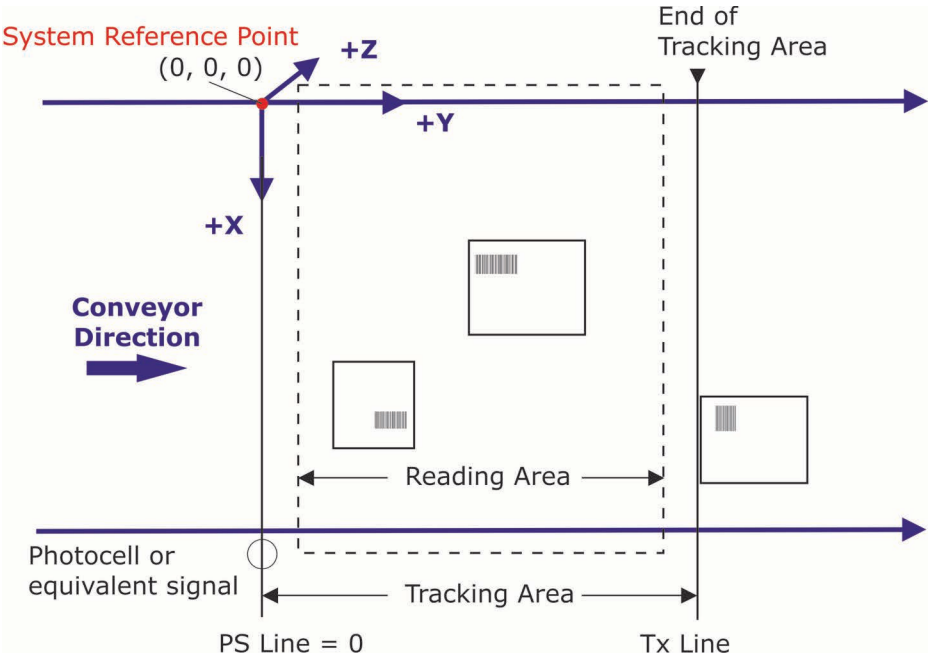
The Reading Area corresponds to the Field Of View of the Matrix N reader.

An area called the Tracking Area is defined between the System Reference Point and the Transmission Line (Tx Line) and obviously includes the Reading Area. All packs passing through the system will have their position tracked.

The System Reference Point is defined as the point where the coordinates (X, Y, Z = 0). The Presence Sensor is normally aligned at the Y = 0 coordinate. If necessary, (after first-time calibration), it can be offset using the PS Line parameter.

The Encoder signal (Encoder Step), together with the Presence sensor is used to track the length of the pack as it passes through the system. The physical encoder can be replaced by an internal signal representing a constant speed conveyor, depending on the application.

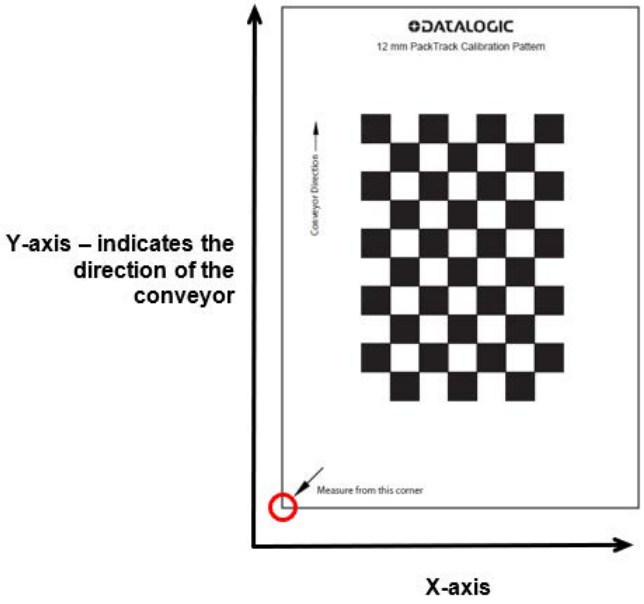
When the pack transits the Tx Line its message is sent to the Host.



CALIBRATION PATTERN

Top/Bottom Orientation

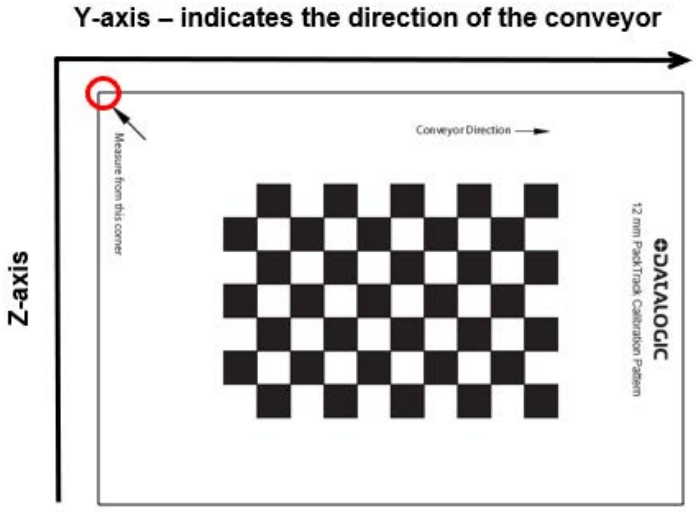
The Calibration Pattern has a precise orientation as shown in the image. The top has 4 black squares and the bottom has 3 black squares.



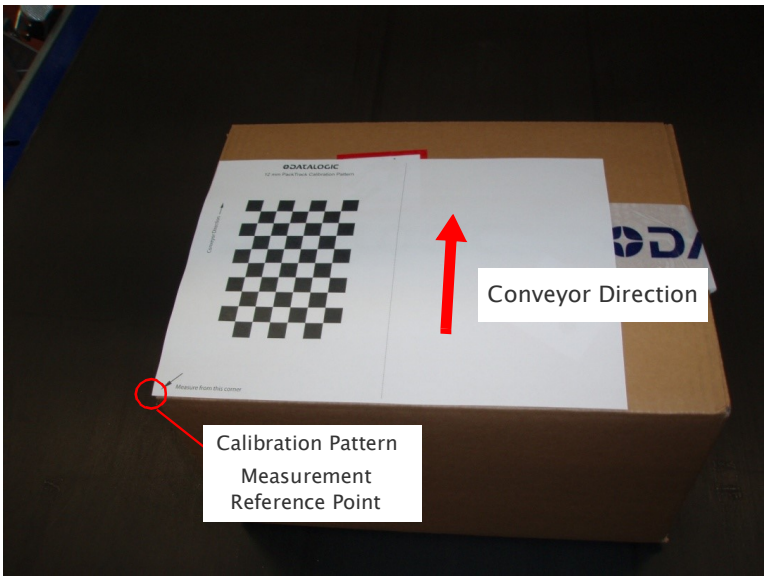
During the calibration process the user has to input the coordinates measured from the System Reference Point to the **lower left corner of the Calibration Pattern** indicated by the red circle.

Left/Right Orientation

When calibrating left/right side readers, the Calibration Pattern is aligned to match the Y-axis (rotated 90° or 270°). This means the short side of the pattern is now aligned with the Z-axis.



Top/Bottom Calibration Chart Positioning



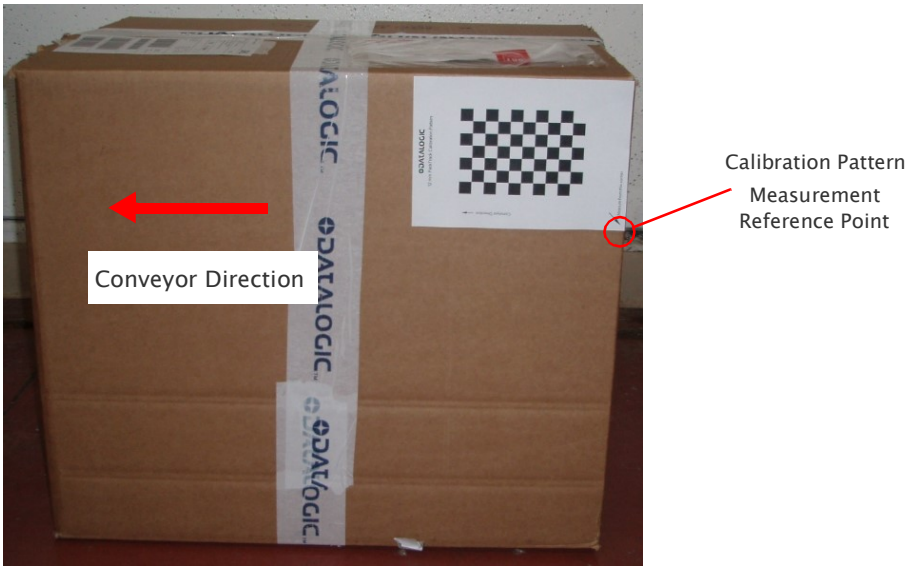
Right Side Calibration Chart Positioning

For Right side readers, the pattern should be placed as shown below so that measurements can easily be taken from the pack itself. **The Conveyor Direction Arrow must always be aligned with the conveyor direction.**


Calibration Pattern
Measurement
Reference Point
Conveyor Direction

Left Side Calibration Chart Positioning

For Left side readers, the pattern should be placed as shown below so that measurements can easily be taken from the pack itself. **The Conveyor Direction Arrow must always be aligned with the conveyor direction.**



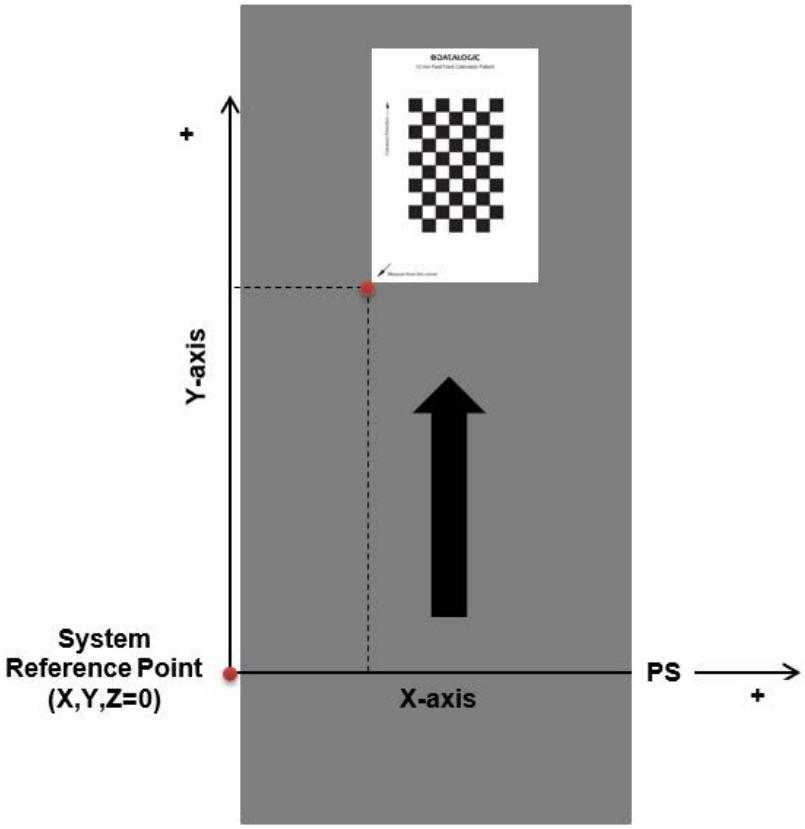
REFERENCE SYSTEM

The image below shows the coordinate reference system on the conveyor: the origin of the coordinate reference system is the System Reference Point.

The Y-axis runs parallel to the conveyor movement direction. So the Y coordinate of a point is measured as the distance between that point the and the System Reference Point along the conveyor direction.

The X-axis runs perpendicular to the conveyor movement direction. The X coordinate is measured as the distance between the point and the System Reference Point across the conveyor.

The Z-axis runs vertically through the conveyor plane with the positive direction above the conveyor.



REQUIREMENTS

The following is a list of required hardware/software that supports PackTrack for Matrix and is necessary for performing the PackTrack Calibration.

- DL.CODE release: 1.00 or later
- Matrix N Standard Application Program Software: 1.00
- or later
- Products: Matrix 300N™ Matrix 410N™, Matrix 450N™, XRF410N™[1]
- PackTrack Calibration Pattern

CAUTION

Print either the A4 or Letter size pdf file according to the paper size you are using. Printing on the wrong size paper or rescaling the Calibration Pattern will cause PackTrack calibration errors.

- Tape Measure
- These instructions

TOP CALIBRATION USING DL.CODE

CAUTION

The conveyor must be STOPPED while performing this procedure!

NOTE

Standard Setup including optical Calibration must be completed before performing PackTrack Calibration.

Calibration is performed using the Calibration Pattern positioned on the plane corresponding to the tallest pack, (Near Plane, i.e. closest to the Matrix reader) and on the plane corresponding to the shortest pack, (Far Plane, i.e. farthest from the Matrix reader).

The PackTrack Calibration is completed only after both planes have been calibrated and saved in Flash.

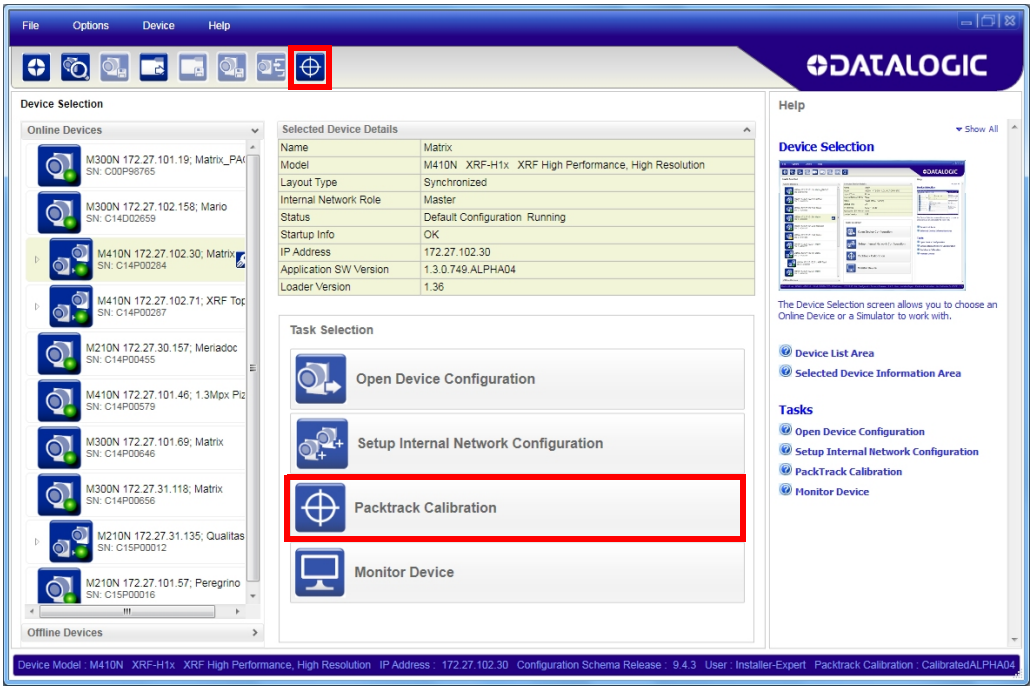
NOTE

Once a completed calibration is performed, it is not possible to perform calibration on a single plane, for example modifying one plane while maintaining the previous parameters of the other plane. The PackTrack Calibration always requires both steps to be completed.

1. For XRF410N products, PackTrack Calibration must be performed separately for all the readers, first the Master and then the Slave (or Slaves in the case of Extended models).

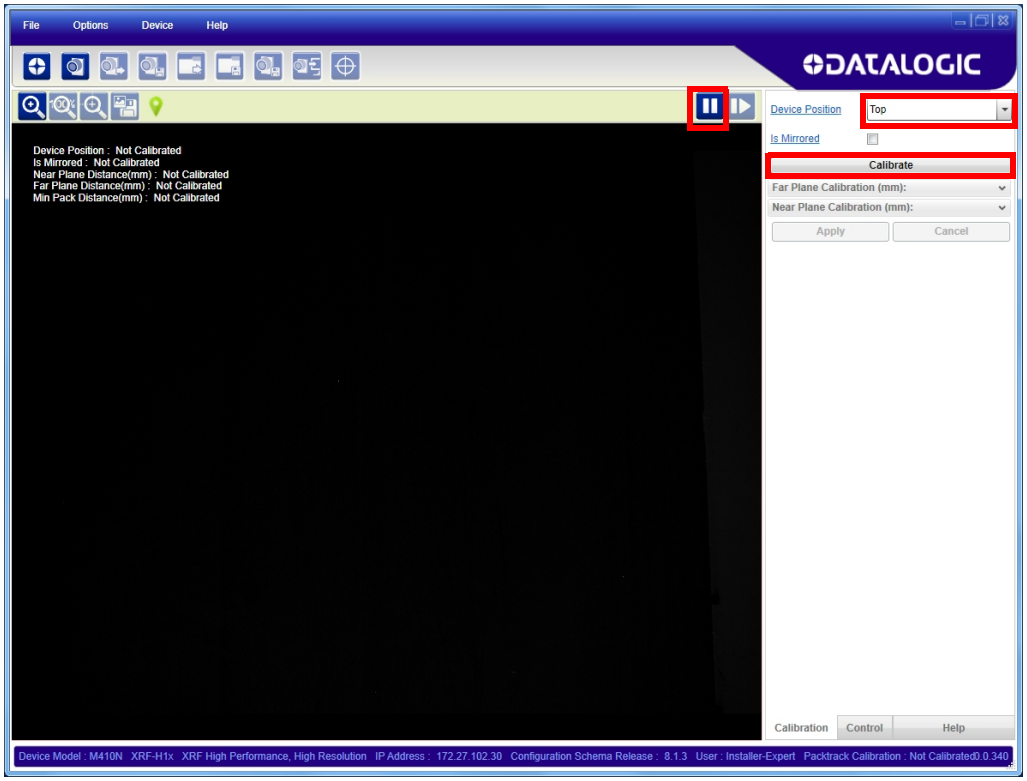
STEP 1 - Run PackTrack Calibration:

1. Select **PackTrack Calibration** from the toolbar icon or item in the DL.CODE Task Area.



2. Set the Device Position to Top from the dropdown list.

3. Click on the Calibrate button to open the Calibration Planes panel for co-ordinate input.



STEP 2 - Determine the PackTrack System Reference Point:

1. Determine the PackTrack System Reference Point, if possible on the conveyor frame surface, where the X, Y, Z co-ordinates = (0,0,0). Visibly mark this point on a piece of tape or other surface, so that it can be used to make the measurements necessary for calibration. The Y = 0 value **normally** corresponds to the PS Line position.

STEP 3 - Far Plane Calibration:

NOTE

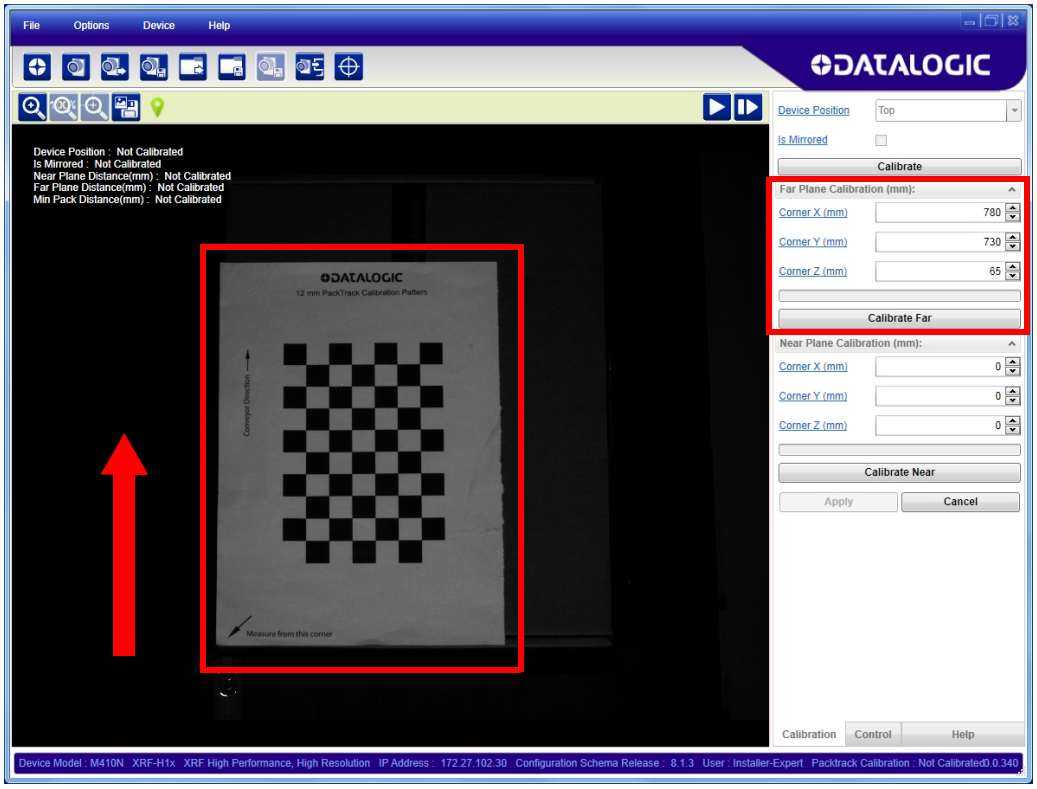
Step 3 and Step 4 can be inverted.

1. The reader should already be running (illuminator flashing and acquiring images) so that positioning can be seen on the monitor. The pause button should be shown indicating the reader is running.

2. Place the Calibration Pattern so that it is completely visible in the monitor window and it corresponds to the plane representing the lowest pack allowed to pass through the system on the conveyor. This is the Far Plane which can also be on the conveyor surface.

NOTE

The Calibration Pattern must be aligned so that the y-axis is parallel to the conveyor movement direction.

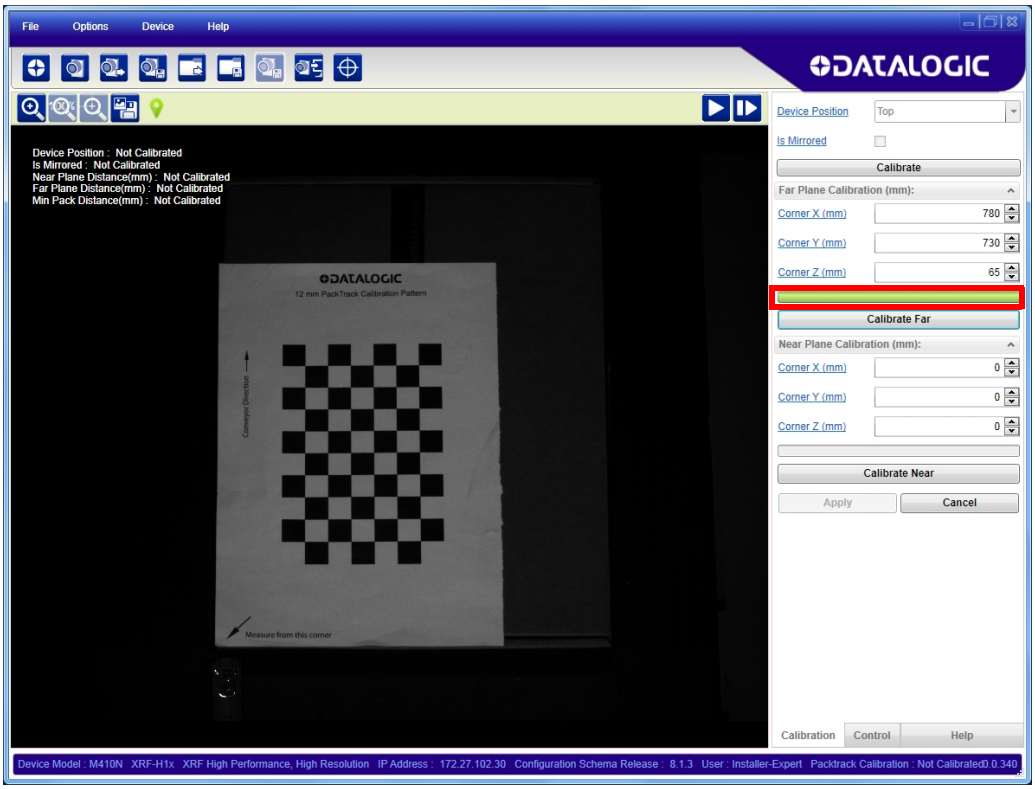


3. Press the **Pause** button.

4. Using the tape measurement, physically measure the X, Y and Z offsets from the System Reference Point to the lower left corner of the Calibration Pattern and input this data **(mm)** into the Far Plane Calibration boxes.

5. Press the **Calibrate Far** button for start Far Calibration.

6. Wait until the operation finishes. An orange progression bar runs above the Calibrate Far button and should end in a solid green bar indicating successful calibration of the far plane.



Possible error causes:

- Calibration Pattern is not completely contained in the Field of View.
- Calibration Pattern is partially obscured by objects covering it.

STEP 4 - Near Plane Calibration:

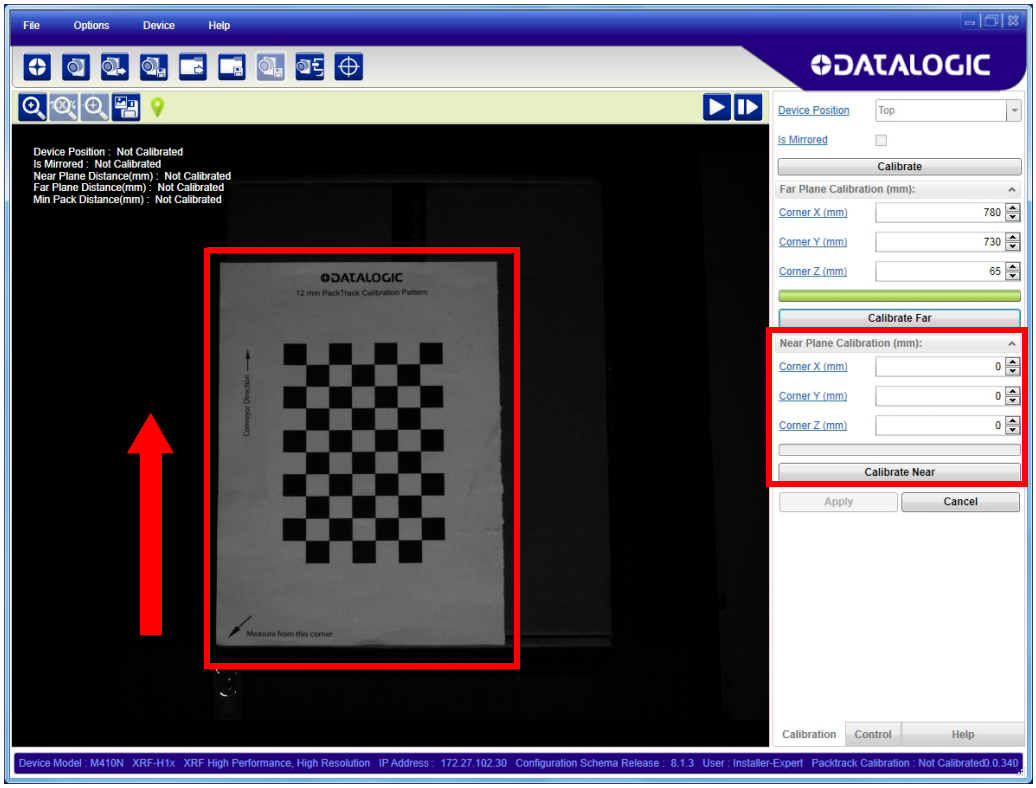
NOTE

Step 3 and Step 4 can be inverted.

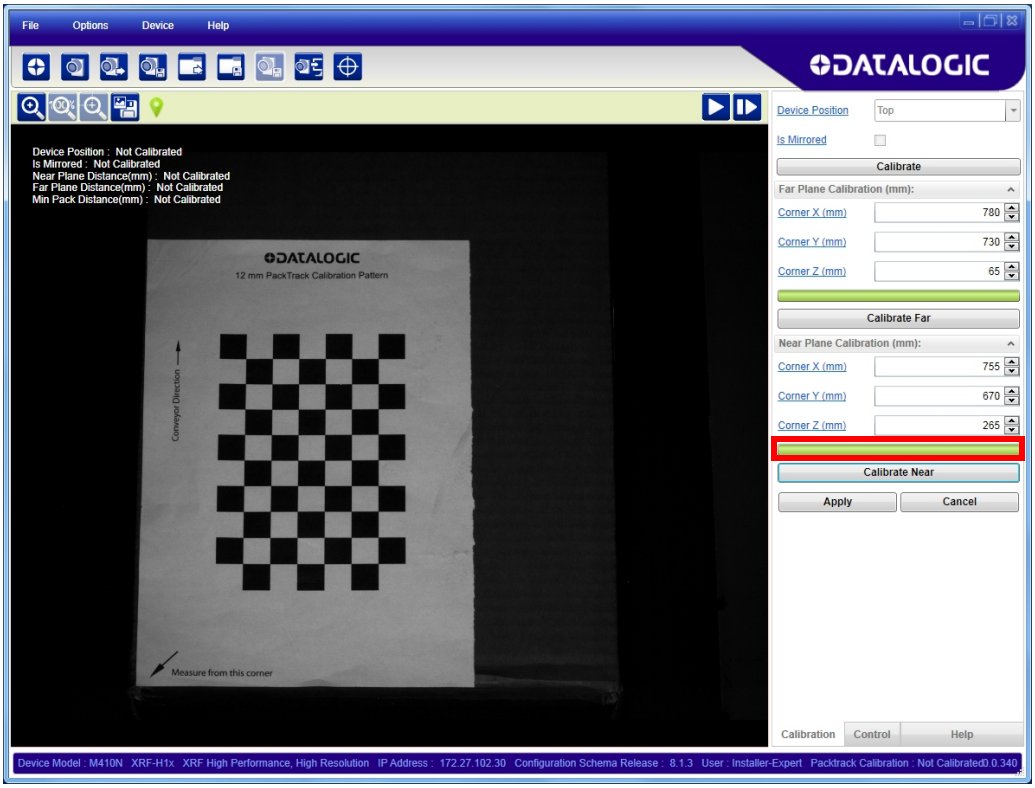
1. Press the **Play** button. The reader begins to acquire images again.
2. Place the Calibration Pattern on a pack so that it is completely visible in the monitor window and it corresponds to the plane representing the tallest pack allowed to pass through the system on the conveyor. This is the Near Plane.

NOTE

The Calibration Pattern must be aligned so that the y-axis is parallel to the conveyor movement direction.



3. Press the **Pause** button.
4. Using the tape measure, physically measure the X, Y and Z offsets from the System Reference Point to the lower left corner of the pack (aligned with the Calibration Pattern) and input this data **(mm)** into the Near Plane Calibration boxes.
5. Press the **Start** button for Near Calibration.
6. Wait until the operation finishes. An orange progression bar runs above the **Calibrate Near** button and should end in a solid green bar indicating successful calibration of the near plane.



Possible Error Causes:

- Calibration Pattern is not completely contained in the Field of View.
- Calibration Pattern is partially obscured by objects covering it.

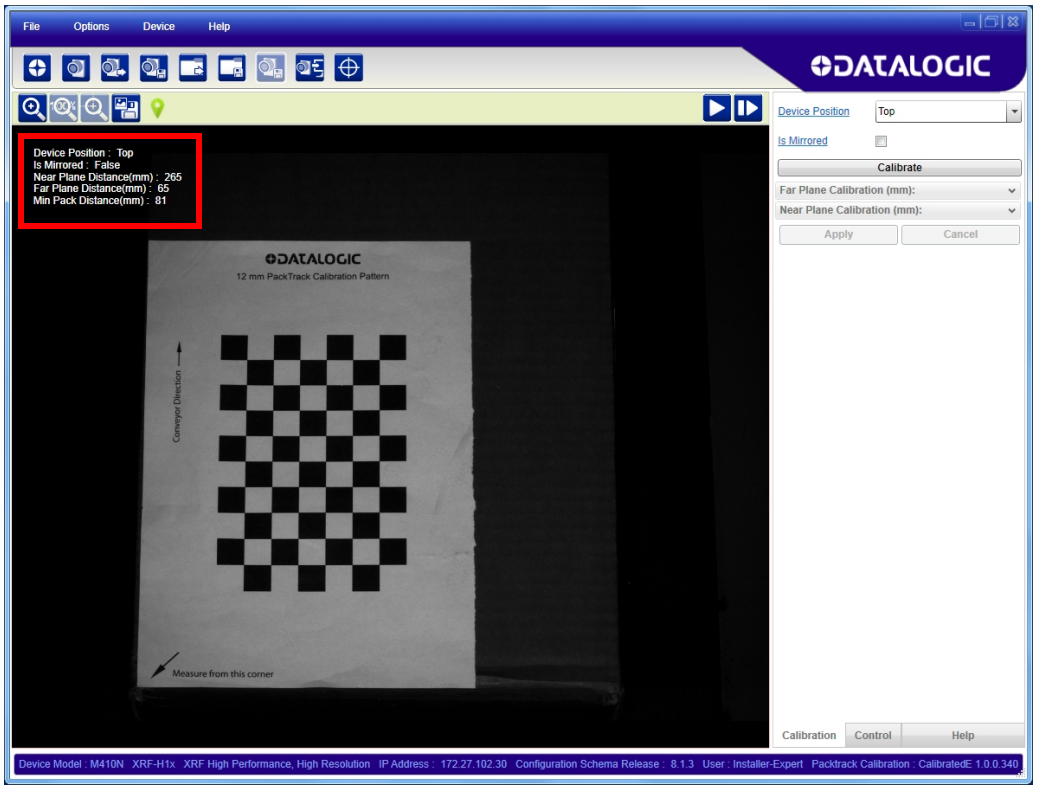
In this case (first time calibration), it is possible to repeat the Near Calibration without losing the previously completed Far Calibration.

NOTE

The items in the monitor window are still shown as "Not Calibrated" because the calibration has not been saved yet.

STEP 5 - Saving Calibration:

7. Click on the **Apply** button to save the calibration values in the reader flash memory.



The calibration also advises the minimum distance between packs (Min Pack Distance), for which correct code to pack assignment can be guaranteed.

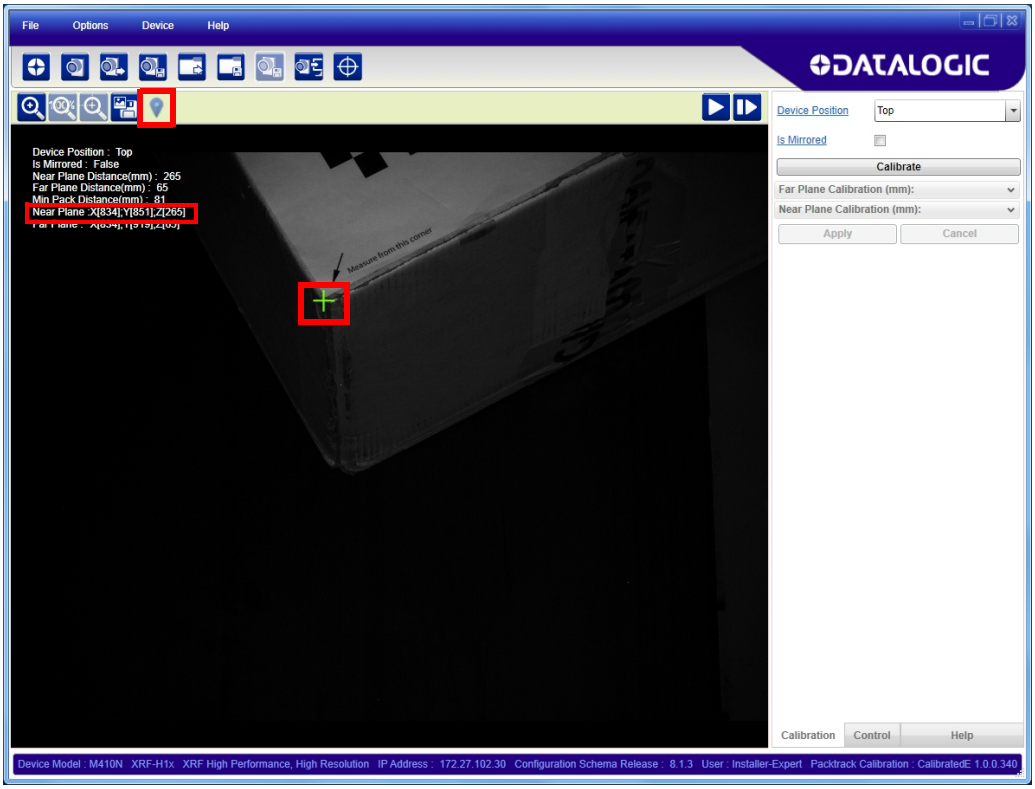
At this point, PackTrack Calibration has been successfully completed.

STEP 6 - Verify Calibration Results:

1. Place a pack, code or other object onto either the Near or Far plane at a different coordinate from the calibration, however it must be visible in the monitor window.

2. Click on the **Show Real World Coordinates** icon. A green cross will appear in the monitor window. Drag this cross with the mouse to an easy-to-measure reference point (i.e. pack edge).

3. Using the tape measure, physically measure the X and Y coordinates from the System Reference Point and compare them to the data shown in the monitor window for the reference plane you are measuring.



CHAPTER 12
DPM

DPM AUTOLEARN

In DL.CODE 1.5.0 the DPM Autolearn feature has been replaced by the Automatic Setup, see “Automatic Setup” on page 15.

If you load devices with software application programs having previous versions and you don’t want to upgrade, then you will need to consult the specific software version documentation for the DPM Autolearn feature.

PRE-CONFIGURATION

Before performing Automatic Setup the following factors require attention in order to produce the best decoding results for DPM code reading applications.

1. The best reading results occur in static applications (no code movement during image acquisition).
2. When using internal illumination, reduce skew angle to minimum to allow uniform lighting on the code surface.
3. The code should be placed as close as possible to the center of the FoV.
4. Reduce reading distance to reduce ambient lighting interference.
5. The best internal illumination chain combination in part depends on the reading distance. More light is applied to the surface at closer distances.

CHAPTER 13
CODE GRADING

SYMBOL VERIFICATION VERSUS CODE GRADING

Symbol Verification involves completely testing the adherence of 2D and 1D codes to the parameters defined in specific International Standards in order to guarantee their reliability and therefore ability to be correctly decoded. In Symbol Verification the (Overall) Symbol Grade is only meaningful if it is expressed in conjunction with the measurement wavelength and aperture used. It should be shown in the format:

Grade / Aperture / Wavelength [/ Angle]

Where:

“Grade” is the overall symbol grade (i.e. the arithmetic mean of the individual Scan Grades for a number of tested images of the symbol).

“Aperture” is the aperture reference number or the diameter in thousandths of an inch (to the nearest thousandth) of the synthetized aperture.

“Wavelength” is the peak light wavelength in nanometers.

“Angle” is the angle of incidence of the illumination relative to the plane of the symbol of the illumination (if 45° it is omitted).

Code Grading for the Matrix N family reader is a feature used to evaluate the quality of a code within a specific application based only on the Scan Grade parameters defined in certain International Standards. **It does not take into consideration the external environmental lighting parameters such as Aperture, Wavelength and Illumination Angle which can in any case affect the Scan Grade.**

The Overall Code Grade is determined by the lowest resulting Scan Grade within the evaluated set of individual Scan Grade parameters.

Through DL.CODE you can also configure the Matrix N reader to perform Code Grading on a specific sub-set of parameters for evaluation. For example, it may be that you are only interested in grading the Print Growth and Symbol Contrast parameters as a function of symbol print quality and therefore only want to monitor these two parameters. Only these two parameters then will contribute to the Overall Code Grade. See also “Code Grading Example Using ISO/IEC 16022 and ISO/IEC 18004 Standards” on page 184.

NOTE

Overall Code Grading cannot be equated with and should not be confused with Symbol Verification.

INTERNATIONAL STANDARDS APPLIED TO CODE GRADING

Matrix N family readers can be used to evaluate printed or marked symbols according to the ISO/IEC 16022, 18004, AIM DPM, and ISO/IEC 15416 standards.

ISO-IEC 16022

(Data Matrix - International Symbology Specification)

The ISO-IEC 16022 Standard specifies general requirements (data character encoding, error correction rules, decoding algorithm, etc.) for Data Matrix symbology.

ISO-IEC 18004

(QR Code - International Symbology Specification)

The ISO-IEC 18004 Standard specifies general requirements (data character encoding, error correction rules, decoding algorithm, etc.) for QR Code symbology.

ISO-IEC TR 29158 (AIM DPM 2006)

(Direct Part Mark Quality Guideline)

The AIM DPM Quality Guideline is applicable to the symbol quality assessment of direct parts marking performed in using two-dimensional bar code symbols. It defines modifications to the measurement and grading of several symbol quality parameters.

The marking processes covered by this guideline are as follows: Dot Peening, Ink Jet, Laser Etching and Electro-Chemical Etching.

ISO-IEC 15415

(Two-Dimensional Symbols - Print Quality Test Specification)

The ISO-IEC 15415 Standard specifies the methodologies for the measurement of specific attributes of two-dimensional bar code symbols, and methods for evaluating and grading these measurements and deriving an overall assessment of symbol quality.

ISO-IEC 15416

(Linear Symbols - Print Quality Test Specification)

The ISO-IEC 15416 Standard specifies the methodologies for the measurement of specific attributes of linear bar code symbols, and methods for evaluating and grading these measurements and deriving an overall assessment of symbol quality.

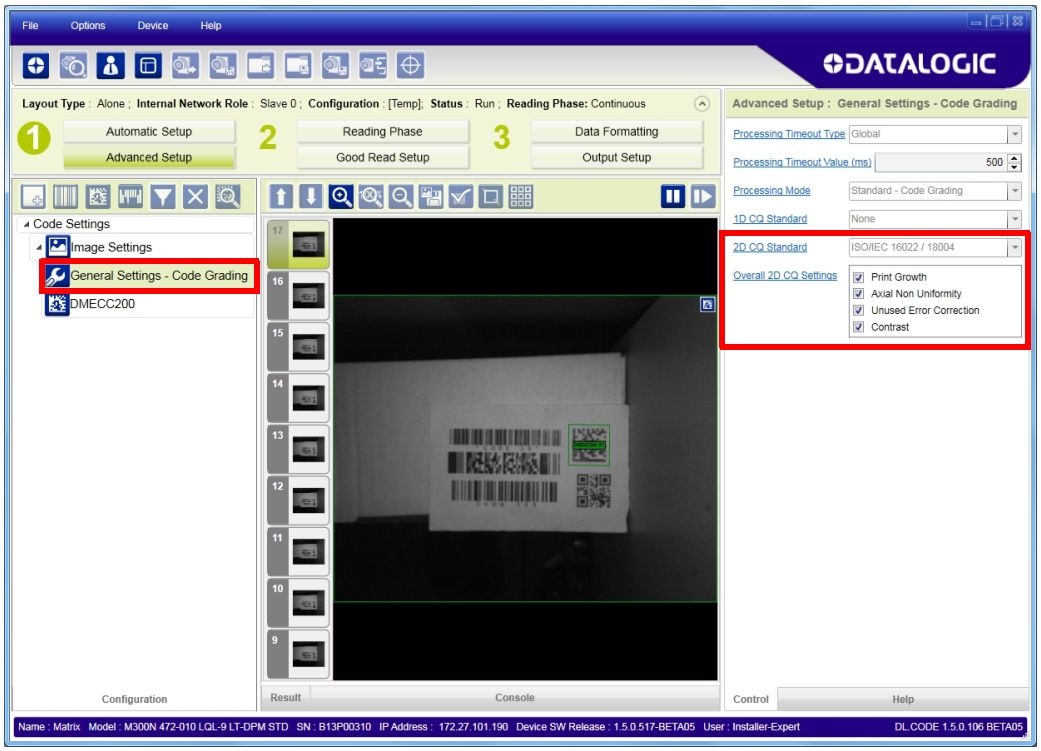
PARAMETERS OVERVIEW

Standards / Parameters	ISO/IEC 16022	ISO/IEC 18004	ISO/IEC TR 29158	ISO/IEC 15415	ISO/IEC 15416
	Data Matrix	QR	2D DPM	2D	1D
Print Growth	P	P	Non Graded	P	Non Graded
Axial Non Uniformity	P	P	P	P	
Unused Error Correction	P	P	P	P	
Symbol Contrast					
Cell Contrast	P	P	P	P	P
Cell Modulation			P	P	P
Decode			P	P	P
Fixed Pattern Damage			P	P	
Grid Non Uniformity			P	P	
Minimum Reflectance			P		P
Minimum Edge Contrast					P
Decodability					P
Modulation					
Defects					P
Reflectance Margin					P

You can enable Code Grading by selecting the International Code Quality (CQ) Standard from the Advanced Setup General Settings menu.

1. Set the Processing Mode parameter to Standard – Code Grading.
2. Select the 1D or 2D Code Quality Standard from the drop down lists:
   1D: None, ISO/IEC 15416
   2D: None, ISO/IEC 16022 / 18004, ISO/IEC 29158 (AIM DPM), ISO/IEC 15415

Depending on the selection, the relative Overall Code Quality Settings box is displayed that allows you to select which parameters to use to determine the overall grading.



ISO/IEC 16022 AND ISO/EIC 18004 STANDARDS

The ISO-IEC 16022 and ISO-IEC 18004 Standards specify the methodologies for the measurement of specific attributes respectively for **Data Matrix** and **QR** code symbols, and methods for evaluating and grading these measurements and deriving an overall assessment of symbol quality.

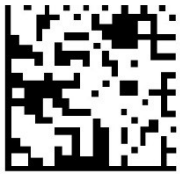
Each quality parameter shall be measured and a grade on a descending scale of integers from 4 to 0 shall be allocated to it. The grade 4 represents the highest quality, while the grade 0 represents failure.

Code Quality Scan Grade Parameters

The following scan grade parameters can be evaluated for the ISO-IEC 16022 and ISO-IEC 18004 Standards:

Print Growth

Measures the deviation of actual element dimension from the expected element dimension due to the printing problems (i.e. overprint or underprint).

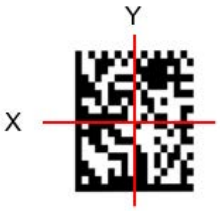


Underprinting

Overprinting

Axial Non-Uniformity (ANU)

Measures and grades the squareness of all modules in the direction of each of the symbol's major axes (X-axis and Y-axis) by applying the decode algorithm to the binarized image.

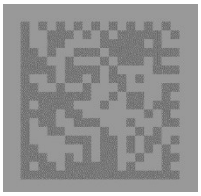


Unused Error Correction (UEC)

This parameter tests and grade the extent to which regional or spot damage in the symbol has eroded the information redundancy margin that error correction provides. 100% Unused Error Correction Capacity is the ideal condition.

Symbol Contrast (SC)

Symbol Contrast tests that the two reflective states in the symbol, namely Light and Dark, are sufficiently distinct within the symbol.



The Overall Code Grade is determined by the lowest resulting Scan Grade within the evaluated set of individual Scan Grade parameters.

ISO/IEC TR 29158 (AIM DPM 2006) QUALITY GUIDELINE

The AIM DPM Quality Guideline is applicable to the symbol quality assessment of direct parts marking performed in using two-dimensional bar code symbols. It defines modifications to the measurement and grading of several symbol quality parameters.

The marking processes covered by this guideline are as follows: Dot Peening, Ink Jet, Laser Etching and Electro-Chemical Etching.

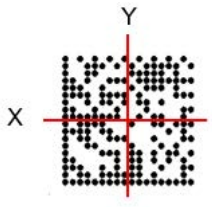
Each quality parameter shall be measured and a grade on a descending scale of integers from 4 to 0 shall be allocated to it. The grade 4 represents the highest quality, while the grade 0 represents failure.

Code Quality Scan Grade Parameters

The following scan grade parameters can be evaluated for the AIM DPM Standard:

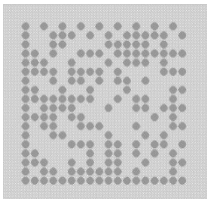
Axial Non-Uniformity (ANU)

Measures and grades the squareness of all modules in the direction of each of the symbol's major axes (X-axis and Y-axis) by applying the decode algorithm to the binarized image.



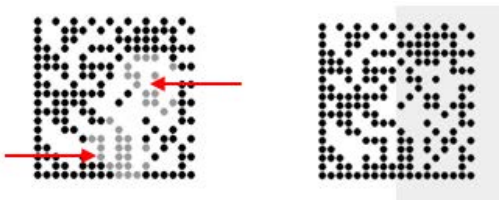
Cell Contrast (CC)

Measures and grades the difference between the means of brightest and darkest values of the symbol (instead of determining differences between the brightest and darkest values).



Cell Modulation (CM)

Cell modulation analyzes the grid center points within the data region to determine the reflectance uniformity of light and dark elements after considering the amount of error correction available in the code.



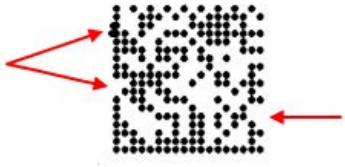
Decode

The Decode parameter tests, on a Pass/Fail basis, whether the symbol has all its features sufficiently correct to be readable. If the image cannot be decoded using the symbology reference decode algorithm, then it shall receive the failing grade 0. Otherwise, it shall receive the grade 4.

This parameter then will always produce Grade A for good reads. If the code cannot be decoded, then a No Read result will be produced by the reader so you will never have a Grade F result for this parameter.

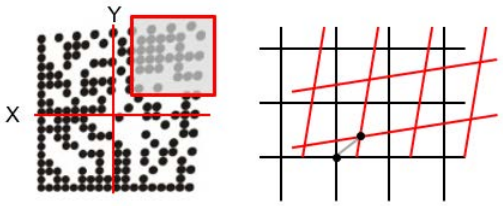
Fixed Pattern Damage (FPD)

This metric is similar to Cell Modulation, but it analyzes the finder pattern and clock pattern as well as the quiet zone around the code instead of the data region.



Grid Non-Uniformity (GNU)

Measures and grades the largest vector deviation of the grid intersections, determined by the reference decode algorithm from the binarized image of a given symbol, from their “ideal” theoretical position. Assuming a grid on which the ideal angle of intersection is 90°, any angle deviation from 90° constitutes Grid Non-Uniformity.



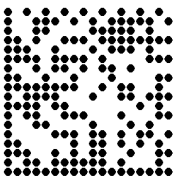
Minimum Reflectance (MR)

The image brightness is adjusted on a reference part, after which this calibrated value is compared with the reflectance of that part. Minimum Reflectance is the ratio of the parts reflectance to the calibrated reflectance

Unused Error Correction (UEC)

This parameter tests and grade the extent to which regional or spot damage in the symbol has eroded the information redundancy margin that error correction provides.

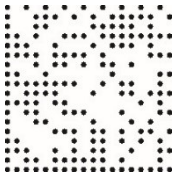
100% Unused Error Correction Capacity is the ideal condition.



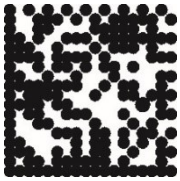
Non Graded Parameters

Print Growth

Measures the deviation of actual elements dimension from the expected element dimension due to printing problems (i.e. overprint or underprint).



Underprinting



Overprinting

ISO/IEC 15415 STANDARD

The ISO-IEC 15415 Standard specifies the methodologies for the measurement of specific attributes two-dimensional bar code symbols, and methods for evaluating and grading these measurements and deriving an overall assessment of symbol quality.

Each quality parameter shall be measured and a grade on a descending scale of integers from 4 to 0 shall be allocated to it. The grade 4 represents the highest quality, while the grade 0 represents failure.

Code Quality Scan Grade Parameters

The following scan grade parameters can be evaluated for the ISO-IEC 15415 Standard:

Print Growth

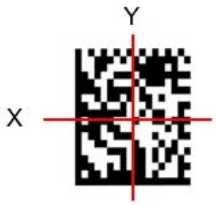
Measures the deviation of actual elements dimension from the expected element dimension due to printing problems (i.e. overprint or underprint).

Underprinting

Overprinting

Axial Non-Uniformity (ANU)

Measures and grades the squareness of all modules in the direction of each of the symbol's major axes (X-axis and Y-axis) by applying the decode algorithm to the binarized image.

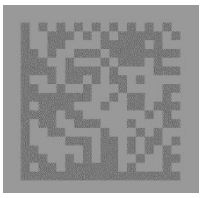


Unused Error Correction (UEC)

This parameter tests and grade the extent to which regional or spot damage in the symbol has eroded the information redundancy margin that error correction provides. 100% Unused Error Correction Capacity is the ideal condition.

Symbol Contrast (SC)

Symbol Contrast tests that the two reflective states in the symbol, namely Light and Dark, are sufficiently distinct within the symbol.



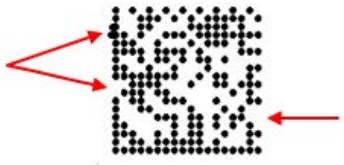
Decode

The Decode parameter tests, on a Pass/Fail basis, whether the symbol has all its features sufficiently correct to be readable. If the image cannot be decoded using the symbology reference decode algorithm, then it shall receive the failing grade 0. Otherwise, it shall receive the grade 4.

This parameter then will always produce Grade A for good reads. If the code cannot be decoded, then a No Read result will be produced by the reader so you will never have a Grade F result for this parameter.

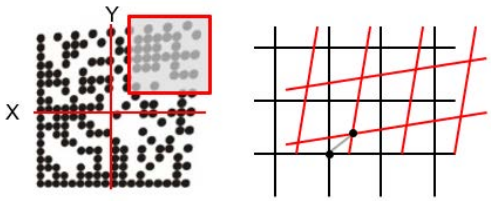
Fixed Pattern Damage (FPD)

This metric is similar to Cell Modulation, but it analyzes the finder pattern and clock pattern as well as the quiet zone around the code instead of the data region.



Grid Non-Uniformity (GNU)

Measures and grades the largest vector deviation of the grid intersections, determined by the reference decode algorithm from the binarized image of a given symbol, from their “ideal” theoretical position. Assuming a grid on which the ideal angle of intersection is 90°, any angle deviation from 90° constitutes Grid Non-Uniformity.



Modulation (MOD)

Modulation is the ratio of the minimum edge contrast to Symbol Contrast. It can be considered as the quality of the Analog signal related to the printing contrast.

The Overall Code Grade is determined by the lowest resulting Scan Grade within the evaluated set of individual Scan Grade parameters.

ISO/IEC 15416 1D STANDARD

The ISO/IEC 15416 Standard specifies the methodologies for the measurement of specific attributes of linear bar code symbols, and methods for evaluating and grading these measurements and deriving an overall assessment of symbol quality.

Bar code symbol quality assessment shall be based on an analysis of the Scan Reflectance profiles. The scan reflectance profile is a record of the Reflectance values measured on a single line across the entire width of the barcode.

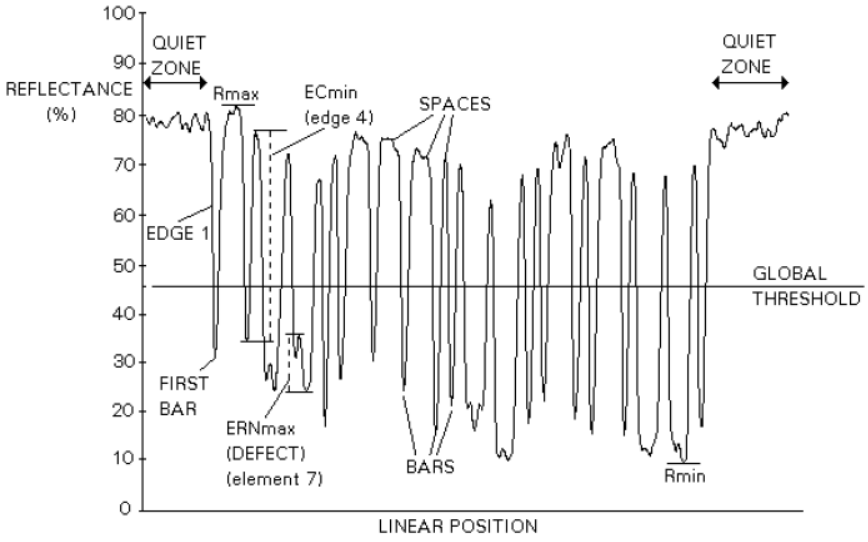


Figure 15 - Scan Reflectance Profile

Symbol Quality grading shall be used to derive a relative measure of symbol quality under the measurement conditions used. Each scan reflectance profile shall be analyzed and a grade on a descending scale of decimals from 4.0 to 0.0 shall be allocated to each of the parameters evaluated.

Code Quality Scan Grade Parameters

The following scan grade parameters can be evaluated for the ISO-IEC 15416 Standard:

Decode

The symbology reference decode algorithm shall be used to decode the symbol using the element edges determined on the Scan Reflectance profile. This algorithm may be found in the symbology specification.

Decodability

The decodability of a bar code symbol is a measure of the accuracy of its production in relation to the appropriate reference decode algorithm.

Defects

Defects are irregularities found within elements and quiet zones, and are measured in terms of element reflectance non-uniformity.

Element reflectance non-uniformity within an individual element or quiet zone is the difference between the reflectance of the highest peak and the reflectance of the lowest valley.

Defect measurement is expressed as the ratio of the maximum element Reflectance Non-Uniformity (ERNmax) to Symbol Contrast.

Minimum Edge Contrast (EC)

Edge contrast is the difference between the Rs (Space Reflectance) and Rb (Bar Reflectance) of adjoining elements including quiet zones.

The lowest value of edge contrast found in the scan reflectance profile is the minimum edge contrast, ECmin.

Minimum Reflectance (Rmin)

Rmin is the lowest reflectance value in the scan reflectance profile. Rmin shall not be higher than 0.5 x Rmax. This parameter is intended to ensure that Rmin shall not be too high, especially when the value of Rmax is high.

Modulation (MOD)

Modulation is the ratio of the minimum edge contrast to Symbol Contrast. It can be considered as the quality of the Analog signal related to the printing contrast.

Symbol Contrast (SC)

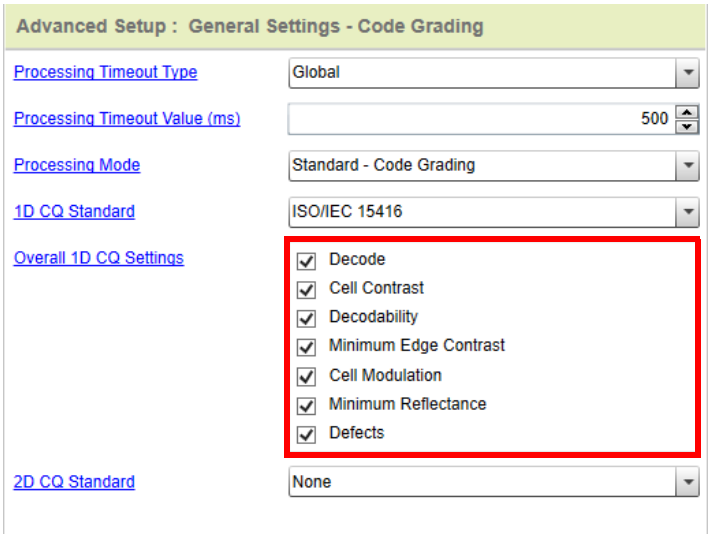
Symbol contrast is the difference between the highest and lowest reflectance values in a scan reflectance profile.

Reflectance Margin

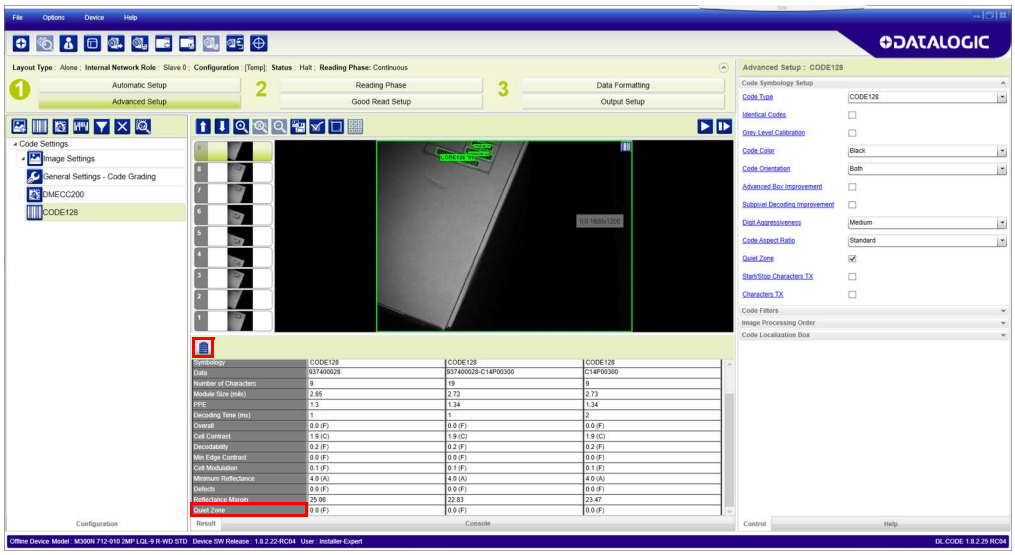
Reflectance margin measures how close the reflectance value of the darkest space or palest bar is to the global threshold, expressed in terms of the symbol contrast. When this value is less than 5% it is likely that the barcode is close to failing on decode.

NOTE

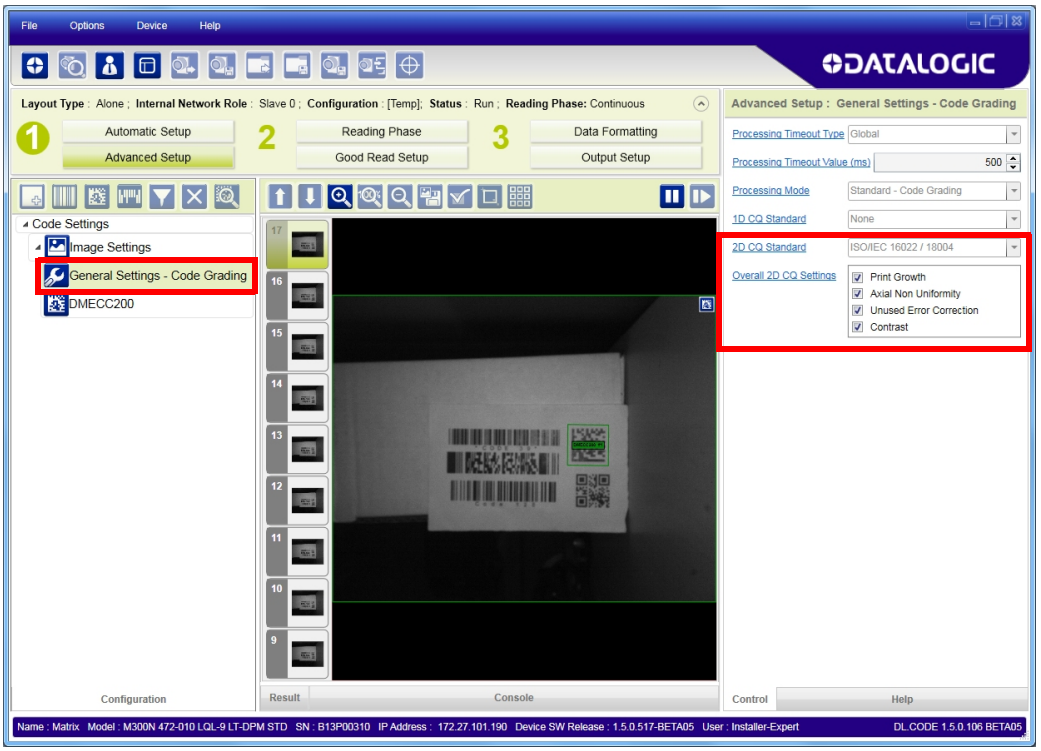
To successfully perform Code Grading according to ISO/IEC 15416 Standard, all Scan Grade parameters must be enabled, as shown in the figure below.

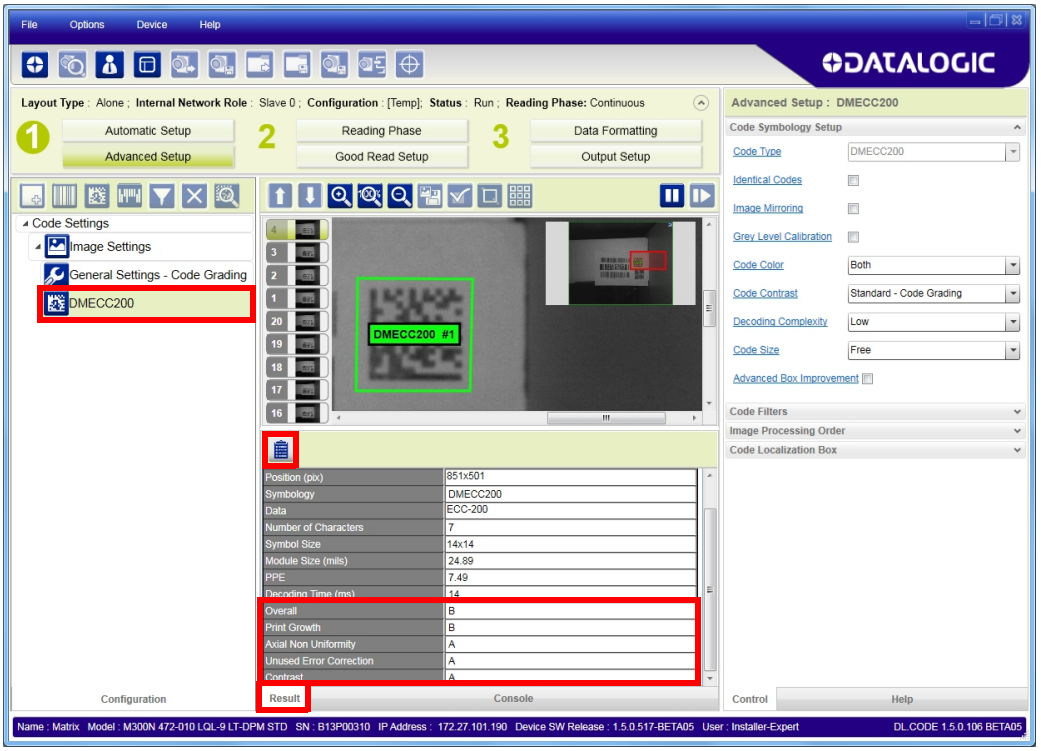


The **Multiple Result View** shows additional information on the decoded symbologies, such as the Quiet Zone, which indicates a trend in the quiet zone quality.

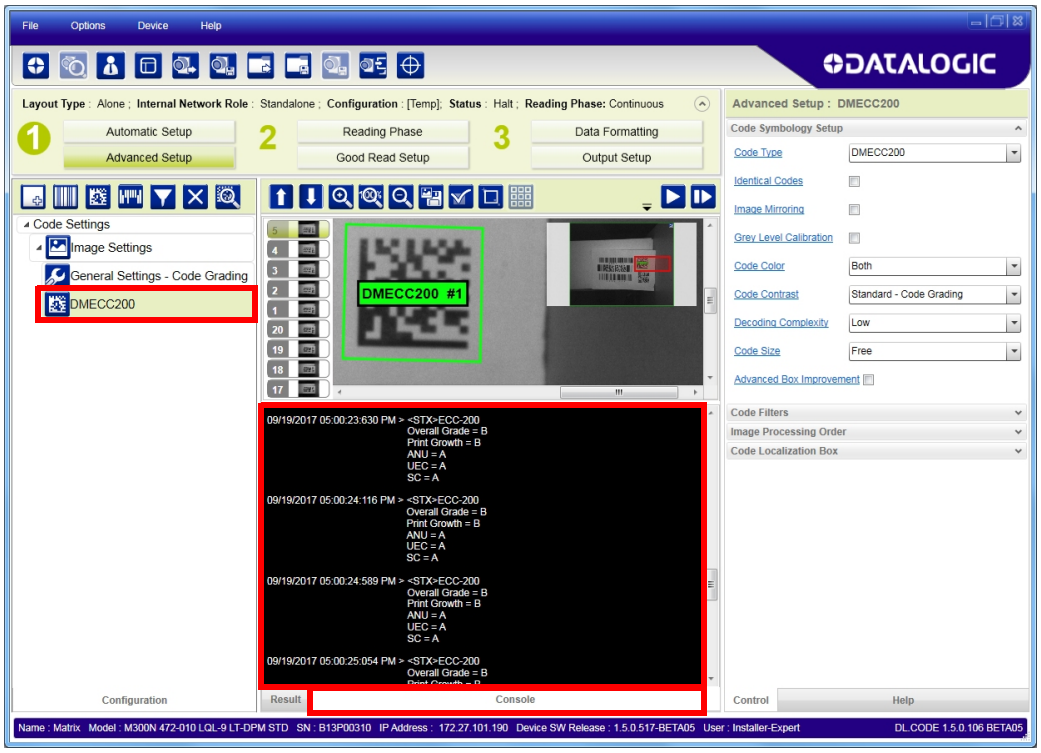


CODE GRADING EXAMPLE USING ISO/IEC 16022 AND ISO/IEC 18004 STANDARDS





The output message can also be defined to include the scan grade parameters.



CHAPTER 14
DEVICE CONFIGURATION SETTINGS

The Statistics behavior for the DL.CODE Monitor is managed through the Configuration Settings window.

The other configuration groups manage some special application parameters for hybrid systems where Matrix readers are used in combination with other devices and monitoring software.

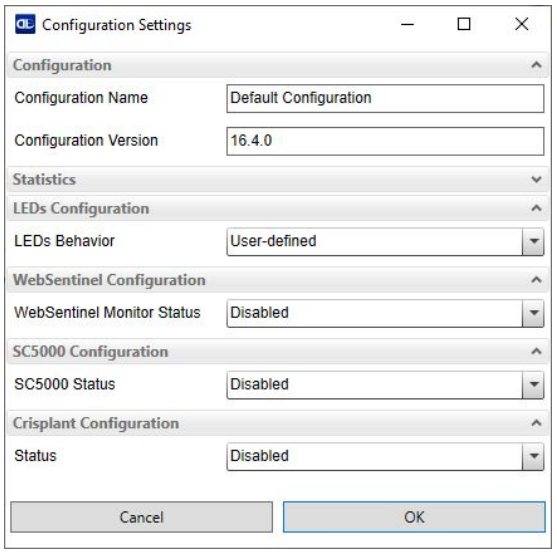
ACCESSING DEVICE CONFIGURATION SETTINGS

To access the device configuration settings the device must be connected to DL.CODE. From the Device>Settings>Configuration Settings menu open the Configuration Settings window.

This window features the following fields:

The configuration **Name** and **Version** of the Default Configuration are reported in the first group (read-only).

This is independent from the actual configuration currently running on the connected device.

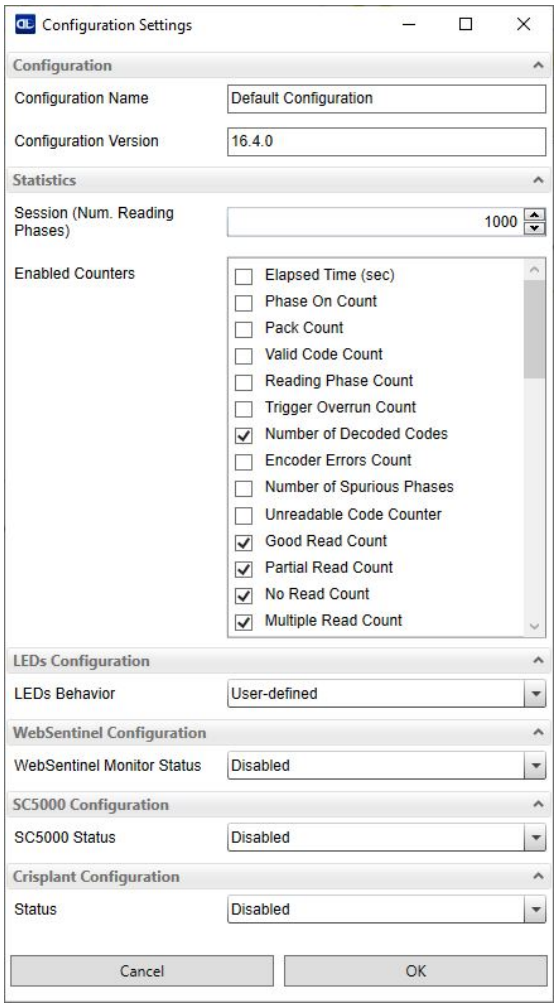


STATISTICS

This group manages the statistics reporting behavior in the Monitor Statistics page. See also “Monitoring Statistics” on page 113.

Session lets you set the number of reading phases to monitor for a Session (from 10 to 1000).

Enabled Counters lets you choose which counter fields to visualize in the Monitor Statistics page.

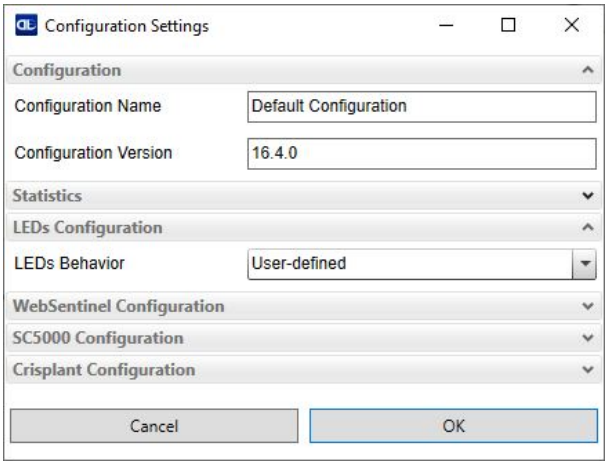


LEDs CONFIGURATION

This group manages the LEDs behavior of your reader.

If **User-defined** is selected, the Green/Red Spots and the 360° Feedback can be configured by the user.

If **STS Mode** is selected, the default LEDs behavior of your reader is used.



In particular, the STS Mode is default for Matrix 320 ATS readers, but it can be enabled for any product working in Phase Mode. The STS Mode behaves as follows:

- the Green/Red Spots and the 360° Feedback cannot be controlled by the user;
- when a Master or Slave reader reads a readable code, the Green Spot and the Green 360° Feedback turn on for 500 ms;
- on Good Read, Success, and Match analysis result, no feedback is provided;

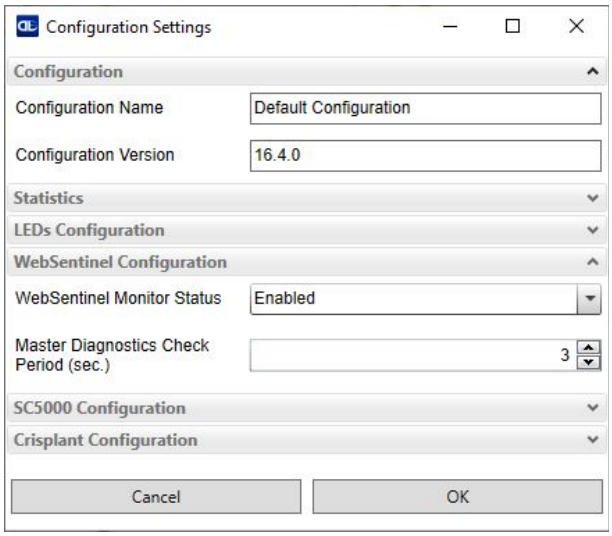
- on No Read, Failure, No Match, and Multiple Read analysis result, the Red Spot and Red 360° Feedback of the Master reader turn on;
- on Partial Read analysis result, both Green and Red 360° Feedbacks of the Master reader turn on, resulting in a Yellow 360° Feedback, however the unreadable codes are not shown.

WEBSENTINEL CONFIGURATION

This group manages configuration to the WebSentinel PLUS Monitoring software program.

WebSentinel Monitor Status enables or disables connection to the WebSentinel PLUS Monitor program.

Master Diagnostics Check Period sets the polling frequency for Diagnostic messages to be reported to WebSentinel PLUS.



SC5000 CONFIGURATION

NOTE

This configuration can only be used in PackTrack operating mode and Code Combination data collection method.

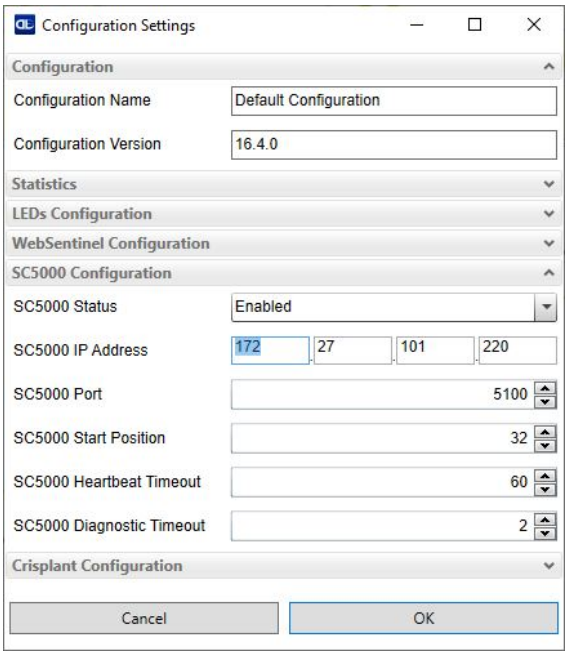
This group manages the configuration of the Matrix device to the SC5000 Controller in a Hybrid configuration.

SC5000 Status enables or disables connection to the SC5000 Controller.

SC5000 IP Address must match the SC5000 Controller.

SC5000 Port must match the SC5000 Controller.

Start Position is the address number (1 to 31) of the Matrix Master or Stand Alone reader which acts as a Slave to the SC5000.



The Matrix Master reader has its own slaves (ID-NET network) whose addresses will automatically be assigned by the SC5000 in consecutive order to the Matrix Start Position address.

SC5000 Heartbeat Timeout must match the SC5000 Controller. A value of 1 to 180 seconds can be configured for the Heartbeat.

SC5000 Diagnostic Timeout sets the polling frequency (in seconds) for Diagnostic messages to be reported to the SC5000 Controller.

NOTE

Since messages sent to the SC5000 on the selected communication channel will be formatted by the SC5000, DL.CODE message Data Formatting for this channel is ignored. This is not graphically indicated in DL.CODE, however do not disable Message Output Channels.

If necessary, Image Saving can be applied, see "Image Saving" starting on page 121.

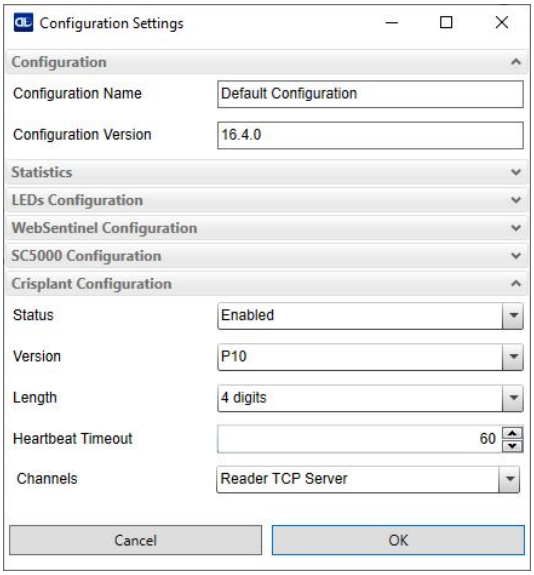
See the Matrix-SC5000 Hybrid System Application Note for complete setup and configuration.

CRISPLANT PROTOCOL CONFIGURATION

NOTE

This protocol can only be used in Phase Mode or PackTrack operating modes.

This group manages configuration for the Crisplant Protocol.



Status enables or disables Crisplant protocol configuration.

Version (Type) selects the Crisplant protocol type. Currently only **P10** is supported. The <CR><LF> characters appear at the end of transmitted/received telegrams.

Length (**Index**) selects between a 4-bit or a 6-bit index applied to the beginning of the telegram. This must match the incoming Crisplant index type.

Heartbeat Timeout must match the Crisplant protocol. A value of 1 to 180 seconds can be configured for the Heartbeat. A value of 0 means Heartbeat is disabled.

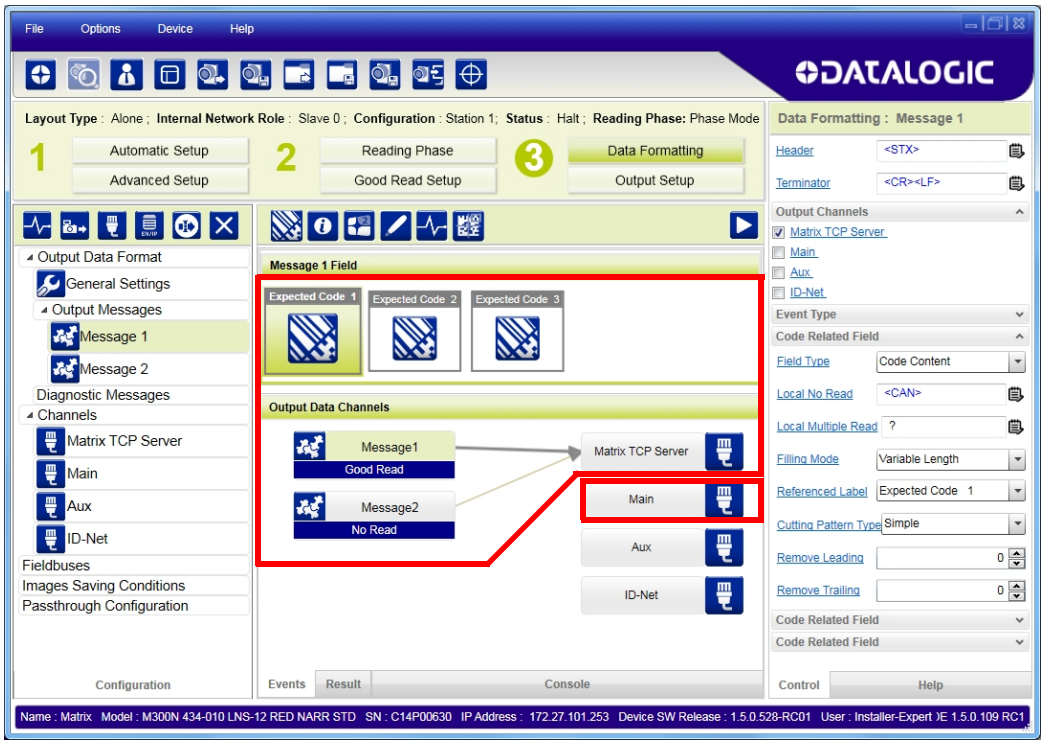
Channels selects which Matrix channel the Crisplant protocol is sent over. This should be a dedicated channel whose communication parameters match the Crisplant Host channel. Matrix Channel parameters are set in the Channels branch of the Configuration Parameters tree area.

NOTE

Since messages sent to the Crisplant Host on the selected communication channel will be formatted with the Crisplant Protocol, DL.CODE message Data Formatting for this channel is ignored. This is not graphically indicated in DL.CODE, however do not disable Message Output Channels. See following example.

Example: Crisplant Protocol Communication on Matrix Main Serial port

For this port the messages will be sent according to the Crisplant protocol and ignore the Data Formatting applied to the TCP Server Port. Even if not used, do not disable the Matrix TCP Server Message Output Channel.



CHAPTER 15
DEVICE ENVIRONMENT SETTINGS

ACCESSING DEVICE ENVIRONMENT SETTINGS

To access the device environment settings obviously the device must be connected to DL.CODE. From the Device>Settings>Settings menu open the Device Environment Configuration window.

The following fields are presented:

Device Name (default “Matrix”) can be personalized by typing a new name in this field.

This can be used to distinguish this device from others in the network (i.e. a name to indicate the device position in the network).

This name also shows up in the Device List Area.

See also note below for use with embedded Profinet-IO communication.

Startup Configuration which can be changed by selecting a different configuration from the dropdown list (if any).

About Device gives details about the various software components currently loaded and running on this device. This data is important to know for troubleshooting purposes.

NOTE

When using embedded Profinet IO Fieldbus communication, Device Name coincides with the Station Name parameter and therefore must adhere to the following rules in order to be recognized by the Profinet Master (Host).

Station Name is a string (max 240 characters) which identifies the node on the Profinet IO network as an alternative to the IP address. The syntax is:

- one or more "labels" separated by the . (dot) character

- max label length is 63 characters

- valid characters are lower case letters (a..z), numbers (0-9) and - (dash) characters

- the . (dot) and - (dash) characters cannot be used as the first or last character in the name.

Example name showing four labels: device-1.machine-8.plant-234.vendor

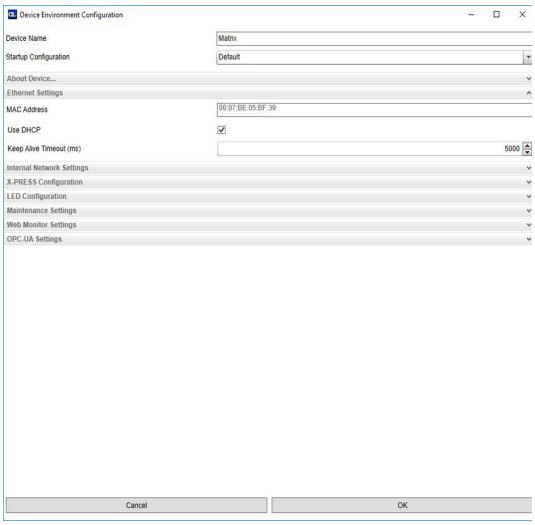
If the Station Name is changed by the host application during runtime, a reset is required in order for changes to have effect.

NOTE

A possible mismatch of the device type declaration on GSDML files and device FWs could show a wrong "Device Type" label on T.I.A. environments. This does not affect the correct Profinet-IO behavior.

ETHERNET SETTINGS

This group presents all the Ethernet Settings for correct device connection to the LAN.



It can also be accessed directly by clicking on the wrench icon in the device list area as shown in "Ethernet Device Discovery" on page 8.

Change the Ethernet Settings (IP Address, Subnet Mask, Gateway Address etc.) according to the network requirements.

The **Keep Alive Timeout** parameter selects the period for which a signal is sent from the device to maintain the Ethernet connection with the DL.CODE UI configuration environment.

NOTE

When using DHCP, if there is a communication loss (i.e. cable disconnect), the reader will take about 40 seconds to boot.

NOTE

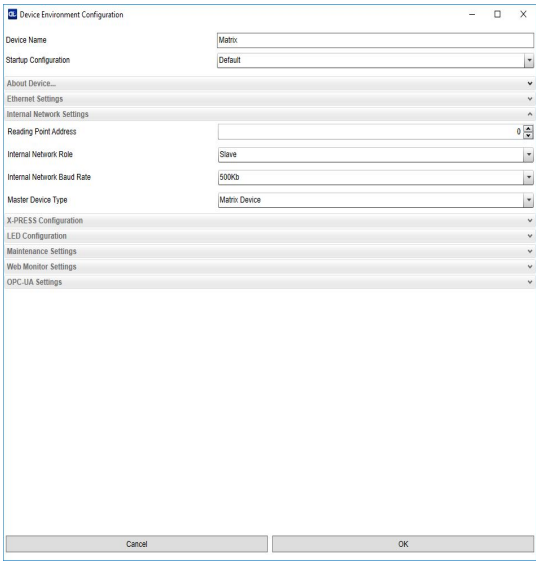
If using DHCP with the embedded Profinet IO interface, the Profinet IO Host will not be able to change the IP address. For this interface it is suggested to use Static IP addressing.

INTERNAL NETWORK SETTINGS

NOTE

All devices are pre-assigned an Internal Network Role independent from their use. If the device is not used in an ID-NET network then this setting can be ignored.

This group allows managing the device role in an ID-NET network (Master or Slave).



The **Reading Point Address** indicates the address of the slave reader. This address is set automatically through the Setup Internal Network Configuration procedure but it can also be set here manually (1-31 for ID-NET Slaves).

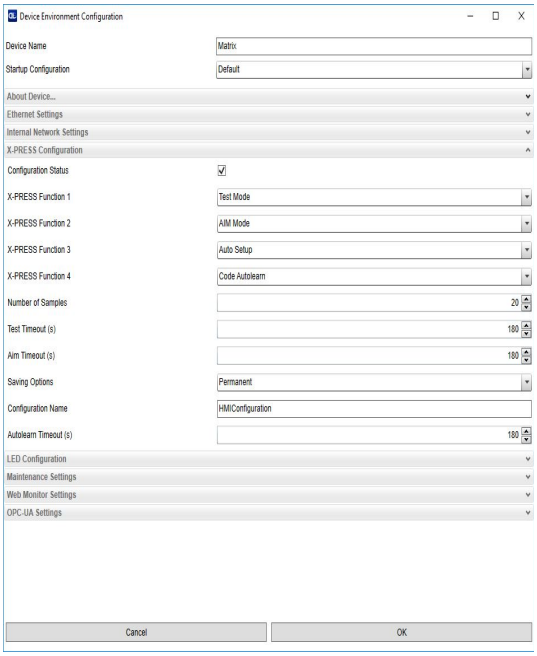
The ID-NET Baud Rate is selected here and must be common to all devices in the network.

For a Matrix Slave Reader the **Master Device Type** parameter can be selected between Matrix Device or SC4000 ID-NET Controller.

See the device Reference Manual for details on ID-NET network configuration.

X-PRESS CONFIGURATION

This group allows managing the HMI X-PRESS™ behaviour. See the device Reference Manual for more details on the X-PRESS features.



Configuration Status enables/disables the X-PRESS features available through the multifunction key on the device.

X-PRESS Function (1-4) assigns an HMI function selected from the list to each one of the X-PRESS keys on the device.

Number of Samples selects the number of samples to analyze for the Test percentage.

Test Timeout sets when the expired timeout causes the Test feature to exit.

Aim Timeout sets when the expired timeout causes the Aim/Autofocus feature to exit.

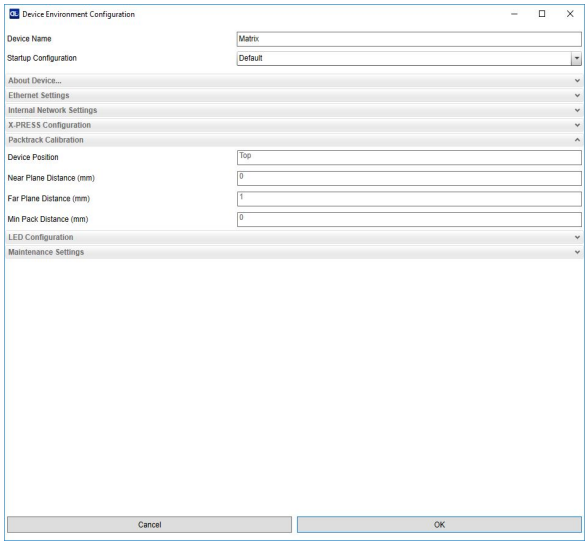
Saving Options selects whether the X-PRESS Setup and Learn features will save their results to Permanent or Temporary memory. If set to Permanent memory, the configuration will be saved as the default configuration in the job list having the Configuration Name.

Configuration Name is the name given to the configuration saved to permanent memory by the X-PRESS Setup and Learn procedures.

Autolearn Timeout sets when the expired timeout causes the Learn feature to exit.

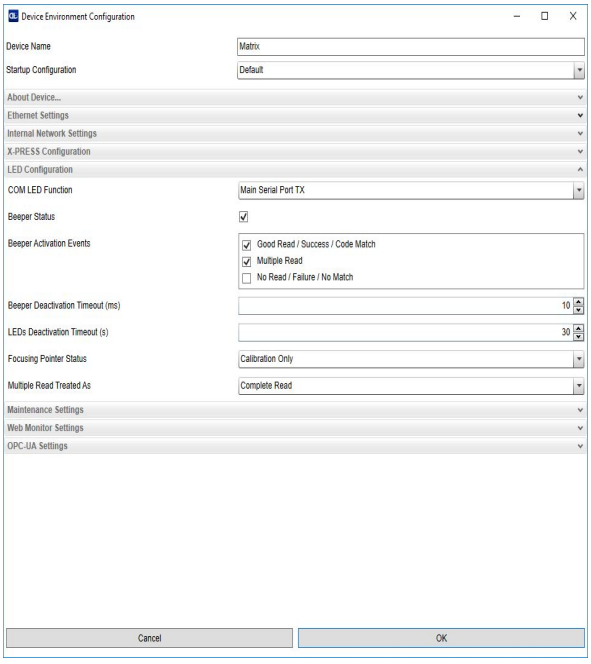
PACKTRACK CALIBRATION

This group shows the PackTrack Calibration parameter settings (read-only) for a device that has already been calibrated.



LED CONFIGURATION

This group manages the device LEDs and Beeper behavior.



COM LED Function selects whether the COM LED on the device (which signals activity on the Main Serial port) is ON when data is transmitted by the device (TX) or received from the Host (RX).

Beeper Status enables/disables the device beeper.

Beeper Activation Events selects which events will trigger the beeper.

Beeper Deactivation Timeout determines the length of the beeper signal.

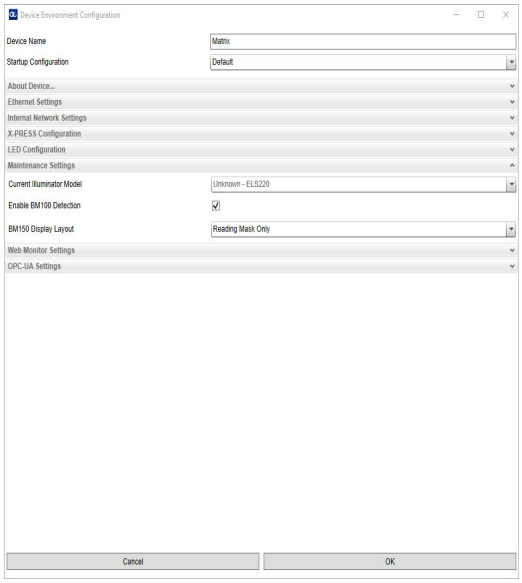
LEDs Deactivation Timeout determines the length of time the LED signals are ON.

Focusing Pointer Status (for devices with laser pointers) sets the aiming system management: **Disabled** - laser pointers always OFF; **Always On** - laser pointers always ON; **Calibration Only** - laser pointers only ON during calibration procedures.

Multiple Read Treated As determines which device LED will be activated on a Multiple Read event: **Complete Read** – the Good Read LED will be activated; **No Read** – the Status LED will be activated.

MAINTENANCE SETTINGS

Current Illuminator Model shows the internal illuminator model associated with this device.



If the device is not correctly associated with its internal illuminator incorrect functioning and/or damage can occur. For Matrix 410N devices see the following Illuminator Management procedure.

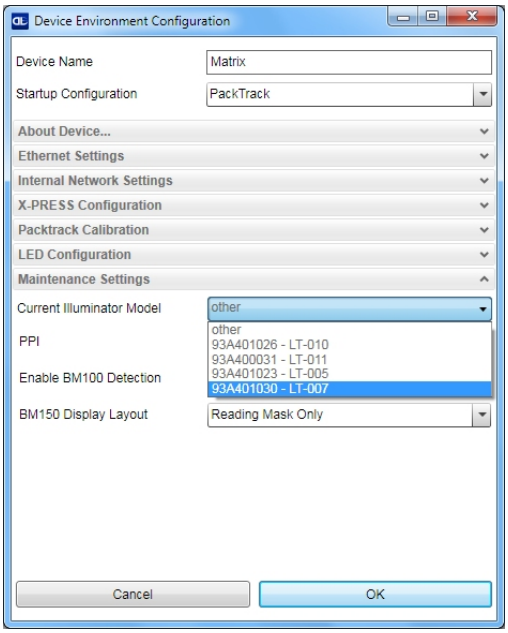
PPI is the same value saved in the **Acquire PPI** image density setting procedure located in the Advanced Setup – Image Settings branch. Here it can be set manually although it is advised to use the Acquire PPI procedure.

Enable BM100 Detection if enabled, at startup, the reader sends a message to recognize the presence of, and communicate with, the External Backup Memory (BM100 Backup Module or integrated QLM-Series accessories). If using the Backup Memory, this parameter must be enabled.

BM150 Display Layout selects which information layout to display on the BM150 accessory display for CBX500 connection boxes. See the Matrix N Reference Manual for more details.

DL.CODE Illuminator Management Procedure for Matrix 410N

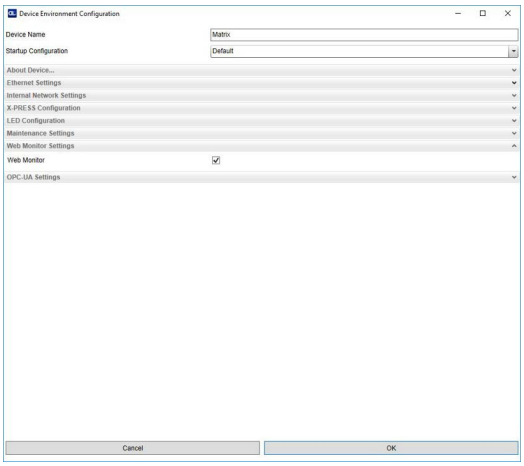
1. In the DL.CODE Maintenance Settings > Current Illuminator Model item, select the correct Illuminator being used from the dropdown list.
2. Click OK and at the device reset prompt click Yes and wait until the device resets. You can confirm by reopening this item from the same menu.



The above procedure must also be performed before any attempt to use the X-PRESS configuration on readers mounting the LT-005, LT-007, LT-010 or LT-011 illuminators.

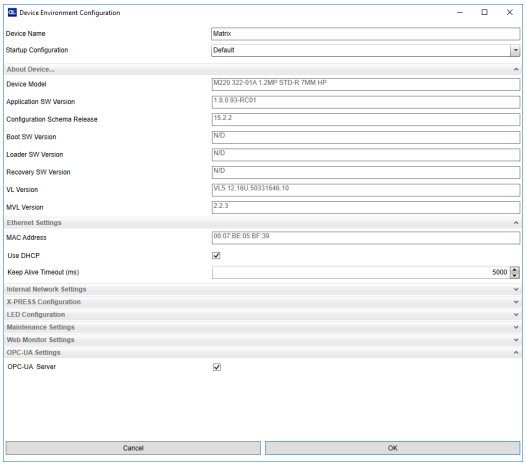
WEB MONITOR

On the **Web Monitor** field you can enable the web monitor function for your reader. For more information, refer to “Web Monitor” on page 70.



OPC UA SETTINGS

On the **OPC UA Settings** field you can enable the OPC UA protocol on your OPC UA-based Matrix reader. For more information, refer to your Product Reference Guide.



CHAPTER 16
MAINTENANCE

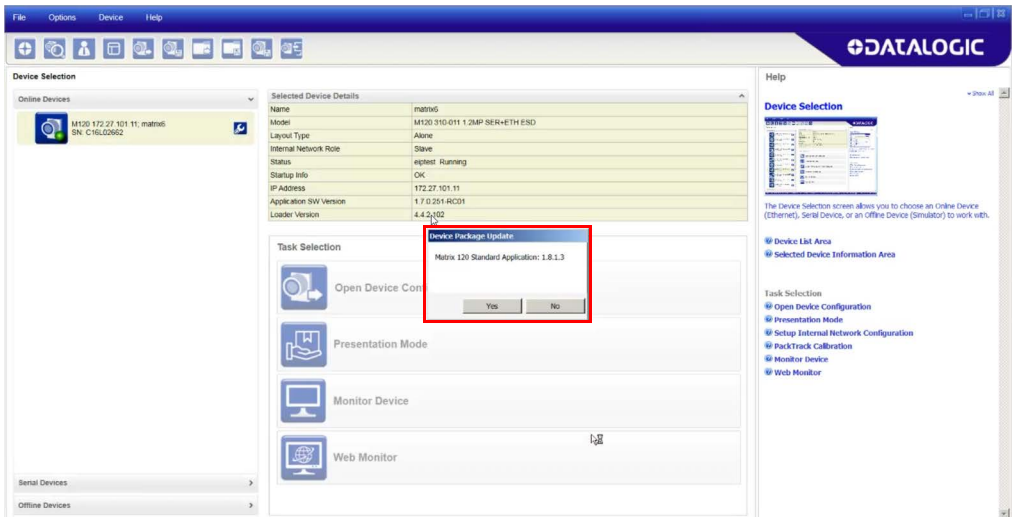
As with nearly all electronic components, performance may drift over time making it necessary to calibrate the device at periodic intervals for optimal reading performance. There is a Recalibration Tool provided in DL.CODE which performs the recalibration procedure. If it ever becomes necessary to perform this procedure, contact our Technical Support team for assistance.

This procedure is intended to be performed by trained technicians and not performed frequently (i.e. only after several years of operation and only if the focus level adversely affects decoding).

UPDATE PACKAGE

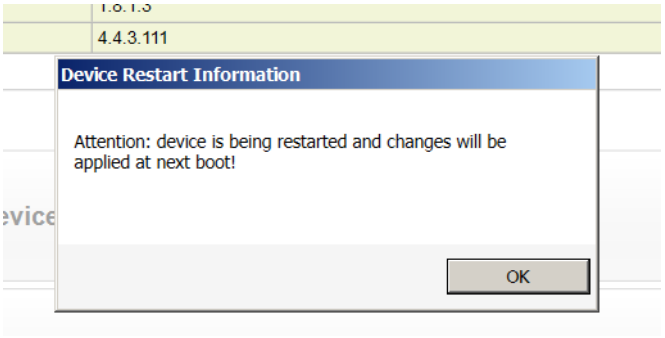
The user can upgrade or downgrade the device firmware (application program, schema, etc.) to a different version.

- First connect your Matrix device (preferably via Ethernet) and select it on DL.CODE.
- In the Main Menu, select **Device > Update Package**.
- Select the suitable package to load on your Matrix device. The following pop-up window is displayed: click *Yes* to confirm and start the upgrade procedure.



- During firmware upgrade, a progress bar is displayed reminding the user NOT to turn off or unplug the device.

- The following Device Restart Information pop-up window is displayed. Click *OK* and wait approx. 1 minute to allow the device to restart and changes to be applied.



CAUTION

Do NOT unplug the device or attempt to reconnect the device to DL.CODE during this phase.

The procedure is completed when all LEDs on the HMI turn on and off, and only the green Ready LED remains on. Now you can reconnect the device and check that the firmware has been upgraded correctly on the Selected Device Information Area.

CHAPTER 17
TROUBLESHOOTING

Problem	Solution
Online Device is not displayed in the Device Selection Area	In order to be found by DL.CODE, Online devices must be powered on and connected to the Local Area Network; if you don't see the desired device within this list, please verify its connections to the LAN and assure it is powered on; then click on the Discovery icon to run a new device search.
Serial Device is not displayed in the Device Selection Area	In order to be found by DL.CODE, Serial devices must be powered on, connected a Serial port of the configuration PC and have the serial port driver installed. if you don't see the desired device within the Serial Devices list, please verify its connections and assure it is powered on and the serial port driver is installed; then click on the Discovery icon to run a new device search.
Cannot open a device configuration DL.CODE Selected Device Information area shows a yellow background	Exit DL.CODE and cycle power to the device. Run DL.CODE and reselect the device. The background should now be green and device configurations can be opened.
Cannot Configure the Device (parameters and icons appear in grey)	The device is in run mode. Click on the Pause button to exit run mode.
Connection problem between DL.CODE and Online Device	If more than one LAN card is present on the local PC and a Simulator (offline device) is enabled in DL.CODE (disabled by default), the program can freeze up. Either disable the simulators in DL.CODE or disable the other LAN cards on the PC.
The User is not able to download images	The FTP Username and FTP Password must be the same as the ones defined in the FTP Client.

Problem	Solution
Some Images are missing	Image availability depends heavily on the number of images transferred and the rate of transfer, (system throughput and network bandwidth). Since Image downloading is usually based on specific criteria (i.e. No Read or Multiple Read conditions) the FTP Server is adequate to handle most applications. In extreme cases where a high throughput application requires all images to be downloaded it is possible that some images may not be available on the reader having been overwritten in the device's circular buffer.

DATALOGIC

www.datalogic.com

Datalogic S.r.l.
Via S. Vitalino, 13 | 40012 Calderara di Reno | Bologna - Italy
Tel. +39 051 3147011 | Fax +39 051 3147205

821005910 (Rev J) November 2021