Barcode Manager Software

Instruction Manual

Original Instructions
207635 Rev. E
5 May 2021
© Banner Engineering Corp. All rights reserved

207635

ПРОМЭЛЕКТРОНИКА/ sales@prom-elec.com

Contents

1 Product Description 5
1.1 Configuration and Monitoring Sessions 5
1.2 Emulators 5
1.3 Device Configuration 6
1.4 Barcode Manager Distribution Contents 6
2 Specifications and Requirements 7
2.1 PC Requirements 7
3 Install Barcode Manager 8
3.1 Uninstall Barcode Manager 8
4 Quick Start 9
4.1 Ethernet Device Discovery 9
4.2 Serial Device Discovery 10
4.3 USB Device Discovery—3000 Models 10
4.4 Decoding Configuration Parameters 11
4.4.1 Automatic Setup 11
4.4.2 Advanced Setup for Manual Adjustable Focus Models 13
4.4.3 Advanced Setup for Liquid Lens Autofocus Models 16
4.5 Operating Mode Configuration Parameters 19
4.5.1 Reading Phase 19
4.5.2 Good Read Setup 20
4.6 Output Configuration Parameters 21
4.6.1 Communications 21
4.6.2 Output Setup 22
5 User Interface 23
5.1 Barcode Manager Device Discovery Window 23
5.1.1 Device List Area 23
5.1.2 Control/Help Panel 24
5.2 Barcode Manager Main Menu and Toolbar 25
5.2.1 User Levels 28
5.3 Multi Image Acquisition Settings 28
5.3.1 Automatic Image Settings Selection 29
5.3.2 External Image Settings Selection 29
5.4 Image Cropping 31
5.5 Code Autolearn Feature 32
5.6 Image Filtering 33
5.6.1 Direct Part Marking (DPM) Applications 34
5.6.2 Image Filters 34
5.6.3 Set an Image Filter: Example 1 38
5.6.4 Set an Image Filter: Example 2 40
5.7 ABR Input/Output Control 42
5.8 Backup and Restore through Barcode Manager 43
5.8.1 Perform a Backup 43
5.8.2 Perform a Restore 44
5.8.3 Replace a Device 44
5.9 Reset to Defaults 44
5.9.1 Restore Default Startup Configuration 44
5.9.2 Restore Default Environment 45
5.9.3 Restore Factory Defaults 45
5.10 Software Reset 46
6 Reading Phase (Trigger) Modes 47
6.1 Introduction 47
6.2 Continuous Mode 47
6.2.1 Acquisition Trigger: Trigger Type = Continuous 47
6.2.2 Acquisition Trigger: Trigger Type = Periodic (ms) 48
6.3 One Shot Mode 48
6.4 Phase Mode 49
6.4.1 Acquisition Trigger: Trigger Type = Continuous 49
6.4.2 Acquisition Trigger: Trigger Type = Periodic (ms) 49
6.4.3 Acquisition Trigger: Trigger Type = Delay (ms) 50
6.4.4 Acquisition Trigger: Trigger Type = External 51
7 Data Collection Methods 52
7.1 Code Collection 52
7.2 Code Combination 53

7.3 Code Presentation 54
7.4 Match Code 55
7.4.1 User Defined 55
7.4.2 Input Based 56
7.5 Presentation Mode 58
8 Protocol Index 60
8.1 Overview 60
8.2 Including Protocol Index in the Output Message 61
8.3 Parsing a Complex Protocol Index Message 62
9 Message Formatting 64
9.1 Field Length Management 64
9.2 String Entry 65
9.3 Separator 66
9.4 Independent Diagnostic Messages 66
9.5 Code Combination Message Formatting Example 67
9.6 Script Formatter 69
9.6.1 Scripting Basics 70
9.6.2 Data Objects 74
9.6.3 Example Scripts 78
10 Host Mode Programming 79
10.1 Conventions 79
10.2 Host Mode Programming 79
10.2.1 Rockwell Automation PLC Support for HMP Commands by Ethernet 79
10.2.2 Collecting HMP Strings from Barcode Manager 80
10.3 Programming Commands 81
10.3.1 Connection to Device 81
10.4 Programming Strings 81
10.4.1 Definitions 81
10.4.2 Programming Strings 82
10.4.3 Examples 91
10.5 ABR Family Parameters List 92
10.5.1 Image Setup 92
10.5.2 Code Setup 93
11 Web Monitor 108
11.1 Access Reader Settings Though Web Monitor 110
11.1.1 Code Grading Monitoring Page 112
11.1.2 Configurator Page 113
12 Monitor 114
12.1 Accessing the Monitor 114
12.2 Monitoring Statistics 114
12.2.1 Statistics Settings 115
12.3 Monitoring Diagnostic Alarms 115
12.4 Monitor Settings 116
12.4.1 Monitor Images Options 116
12.4.2 View Window 117
13 Image Saving 119
13.1 Device Image Buffer 119
13.1.1 Image Saving Using Reader TCP Server/Client 119
13.1.2 Image Saving Using Reader FTP Client 120
13.1.3 Image Export to FileZilla FTP Server 121
13.1.4 Image Saving On Demand Using Send Images HPM Command 126
13.1.5 Image Saving Using Internal Buffer 126
13.2 UI Image Buffer 127
14 Multi Device Configuration Options 128
14.1 Pass-Through Configurations 128
14.2 Internal Network Configurations 130
14.2.1 Master Configuration 130
14.2.2 Internal Network Icons 130
14.2.3 Multidata ID-NET Network Configurations 131
14.2.4 Synchronized ID-NET Network Configurations 137
14.2.5 Verify Master/Slave Synchronization Configuration 140
14.2.6 Alternative Device Role Selection 142
15 Code Grading 144
15.1 Symbol Verification Versus Code Grading 144
15.2 International Standards Applied to Code Grading 144
15.3 Parameters Overview 145
15.4 ISO/IEC 16022 and ISO/IEC 18004 Standards 146
15.4.1 Code Quality Scan Grade Parameters 146
15.5 ISO/IEC TR 29158 (AIM DPM 2006) Quality Guideline 147

15.5.1 Code Quality Scan Grade Parameters 147
15.6 ISO/IEC 15415 Standard 150
15.6.1 Code Quality Scan Grade Parameters 150
15.7 ISO/IEC 15416 1D Standard 152
15.7.1 Code Quality Scan Grade Parameters 152
15.8 Code Grading Example Using ISO/IEC 16022 and ISO/IEC 18004 Standards 153
16 Device Configuration Settings 155
17 Device Environment Settings 156
17.1 Accessing Device Environment Settings 156
17.2 Ethernet Settings 156
17.3 Internal Network Settings 157
17.4 Smart Teach Configuration 157
17.5 LED Configuration 158
17.6 Maintenance Settings 160
17.7 Web Monitor Settings 160
18 PPI (Pixels Per Inch) Setup Chart 161
19 Troubleshooting 163
20 Product Support and Maintenance 164
20.1 Update the Software and Firmware 164
20.1.1 Update the Firmware 164
20.2 Contact Us 165
20.3 Banner Engineering Corp Limited Warranty 165

1 Product Description

Barcode Manager software interfaces with ABR Series Barcode Readers from Banner Engineering. Use the Barcode Manager software to configure and monitor devices in a few simple steps. Powerful 1D and 2D code reading is available in a simple to use software interface that offers advanced features for solving even the most difficult barcode reading applications.

Connect your device to the software over Ethernet, serial or USB, or use the offline Emulator to create inspections while not connected to a device. Use Automatic Setup for a one-step process for automatically setting the image acquisition settings , learning the code symbology, and optimizing decoding settings. Users have full access to change and configure settings in Advanced Setup. Once you have optimized your barcode reader to read your specific code, configure inputs and outputs, good read pass criteria and format the output messages.

Main features include:

- Connect and monitor multiple ABR devices
- Three different user access levels to have simplicity or advanced control
- Easy auto-detecting Master/slave configuration available
- Multi-Lingual Interface
- Monitor configured devices with PC software, in a web browser or on an external device
- Optional scripting tool for complex logic and data formatting
- Configurable Code Quality Grading Options
- Image filters for improving readability of low quality codes

1.1 Configuration and Monitoring Sessions

Perform device configuration using Barcode Manager running on a remote PC through a single session. Multiple instances of Barcode Manager cannot run on one PC. After a device is connected for configuration it cannot be accessed by another PC running Barcode Manager.

However, the monitoring feature can be accessed simultaneously by several PCs running Barcode Manager.

1.2 Emulators

Barcode Manager has different device prototypes which can be loaded as offline devices (emulators). This allows offline configurations to be prepared and loaded to a device at a later time.

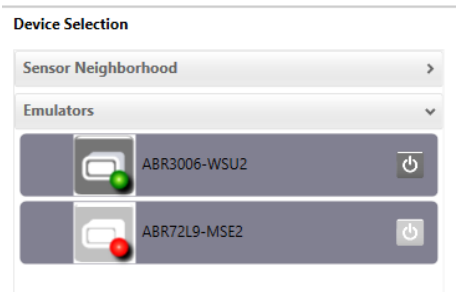
Note: Creating and saving offline configuration (.bncfg) files and then loading them onto physical devices can be done, but the following precautions should be followed:

- 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 Setup, Reading Phase, and Output Setup
- The following parameter groups will be maintained: Advanced Setup Code Setup, Good Read, and Communications

To load an Emulator, click on the **Emulators** tab at the bottom of the Device List area to open the list of available emulators.

All of the emulators are offline by default. To select a reader, click its **Emulator Power** icon.

Figure 1. Device Selection—Emulators

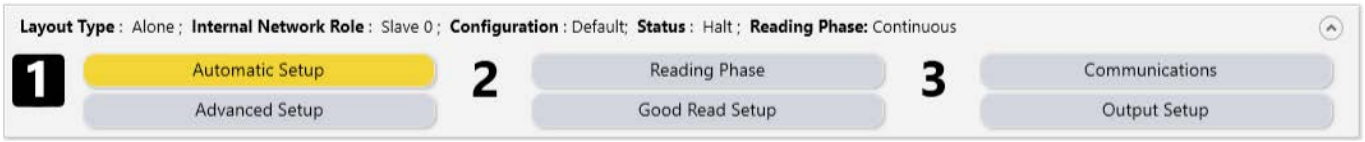


Double-click or drag the Emulator into the Selected Device Information Area and begin a new configuration. See Quick Start on p. 9.

1.3 Device Configuration

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

Figure 2. Barcode Manager Configuration Groups



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

1 Decoding

Automatic Setup

This feature provides a one-step process for automatically setting the image acquisition settings and the code symbology selection (for a single code type). Click **Start Automatic Setup** to start the procedure. It is particularly useful for direct part marking (DPM) applications and requires no calibration.

The Automatic Setup procedure optimizes a number of decoding settings for whichever code type it finds, and this procedure is the only way to automatically perform this optimization.

Advanced Setup

This group completely manages the following:

- Image Setting parameters, including **Image Auto Setup** for photometry, multiple Image Settings, **Focus Autolearn** for Liquid Lens Autofocus 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 acquisition.

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

Communications

This group manages the output message to the Host.

Output Setup

This group manages the digital outputs as well as the Green/Red Spots.

1.4 Barcode Manager Distribution Contents

The Barcode Manager program distribution contains the following:

- Complete Installation of Barcode Manager
- This manual

2 Specifications and Requirements

2.1 PC Requirements

Administrative rights are required to install the Barcode Manager software.

Operating System
Microsoft® Windows® operating system version 7, 8, or 10 [2] [3]
Barcode Manager does not currently support Windows Embedded (often used in industrial PCs and/or PLCs)

System Type
32-bit or 64-bit

Hard Drive Space
2 GB hard disk for 64-bit machines; 1 GB hard disk for 32-bit machines

Memory (RAM)
1 GB RAM

Processor
2.00 GHz or faster microprocessor

Screen Resolution
One 19-inch or larger monitor, optimized for 1280×1024 resolution

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

Connection
100 Base-T Ethernet

[2] Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries.
[3] Barcode Manager no longer supports Windows XP.

3 Install Barcode Manager

Administrative rights are required to install the Barcode Manager software.

Important: Install Barcode Manager on a Windows® 7, 8, or 10 [4] computer. Barcode Manager does not currently support Windows Embedded (often used in industrial PCs and/or PLCs).

1. Download the latest version of Barcode Manager from www.bannerengineering.com.
2. Navigate to and open the downloaded file.
3. Run the downloaded file to access the installation screen.
4. Follow the onscreen installation procedure.
   After the installation is complete, the Barcode Manager entry is created under **Start** > **Programs** > **Banner Engineering**. A desktop icon is also created.

3.1 Uninstall Barcode Manager

To completely uninstall Barcode Manager, including software drivers, from your PC you must use the following sequence:

1. Uninstall all versions of Barcode Manager from your PC from the Control Panel.
2. In Windows, click the **Start** button, and then, on the **Start** menu, click **All Programs**.
3. Click the **com0com** folder, then click **Uninstall** to uninstall the **com0com driver** driver.
4. Uninstall the **USBCOMInstaller** from the Control Panel.
5. Uninstall the **cdcecmInstaller** from the Control Panel.

Note: Trying to uninstall software drivers before all versions of Barcode Manager are uninstalled will cause an error.

[4] Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries.

4 Quick Start

The following is an example configuration demonstrating the basic steps of Barcode Manager configuration.

To configure your device for your application using Barcode Manager, the following preliminary steps are assumed:

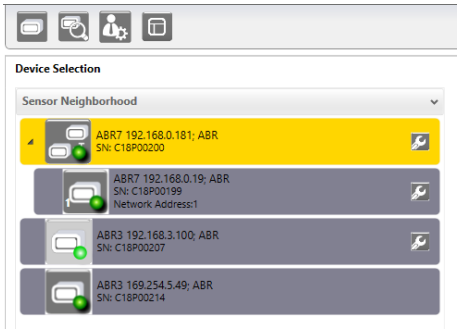
- The reading device(s) are installed and running
- Barcode Manager is installed and running

4.1 Ethernet Device Discovery

The following configuration procedure assumes that a laptop computer running Barcode Manager is connected to a factory default reader through the Ethernet port.

The Barcode Manager user interface opens and displays a list of all the devices belonging to the Local Area Network (LAN).

Figure 3. Device Discovery

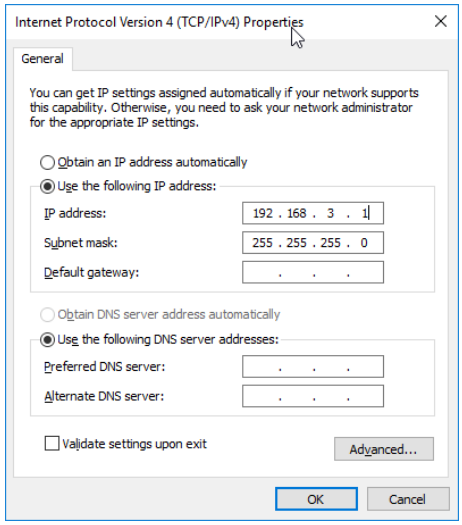


The Barcode Manager discovery feature also shows devices not belonging to the LAN and displays them in light gray (see Figure 3 on p. 9).

The following is an example configuration for Windows® operating system version 7, 8, or 10.

1. Confirm the network connections. Changing the Local Area Connection (LAN) properties of the programming computer to be compatible with the ABR device on the network may be required for connection.
   a) Click the **Start** button, then on the **Start** menu, click **Control Panel** or search for **Control Panel**.
   b) In **Control Panel**, click **Network and Internet**, then click **Network and Sharing Center**, and then click **Change adapter settings**.
   c) Right-click on the connection that you want to change, then click **Properties**.
      If you are prompted for an administrator password or confirmation, enter the password or provide confirmation.
   d) In the connection properties, click **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**.

Figure 4. Local Area Connection Properties



   e) In the **Internet Protocol (TCP/IPv4) Properties**, select **Use the following IP address**.
   f) Make sure that the IP address is 192.168.3.1, and the subnet mask is 255.255.255.0.
      The IP address must be compatible with the default device address 192.168.3.100.

2. As an alternate method, change the IP address of the device.
   a) Add the device to the LAN by aligning its IP Address to the network. The network administrator should provide valid LAN address(es).
   b) Click the **device wrench** icon to open the **Device Environment Configuration** window.
   c) Change the Ethernet Settings (IP Address, Subnet Mask, Gateway Address, etc.) according to the network requirements.
   d) Click **OK**.
3. In Barcode Manager, click **Find Devices**.
   The device displays in **Sensor Neighborhood** with a dark gray icon, meaning it is now part of the LAN and can be configured. The new IP address also displays.
4. Double-click or drag the **device** icon into the **Selected Device Information Area**.
   Details about the device display in this area.

Figure 5. Device Selection—Selected Device Details

Selected Device Details	
Name	ABR
Model	ABR3006-WSU2
Layout Type	Alone
Internal Network Role	Slave
Status	Default Running
Startup Info	OK
Application SW Version	1.5.4.645-f273f1
Loader Version	N/D

After device discovery, configure your device through Barcode Manager.

4.2 Serial Device Discovery

Note: Although this feature allows all devices to be configured through their Serial Interface, be aware that transmission speeds and some Barcode Manager features are limited when using this interface. It is advised to use the Ethernet or USB interface whenever possible.

Serial Device Discovery is not enabled by default.

1. In Barcode Manager, from the main menu go to **Options** > **UI Settings** window.
2. Click on the **Global Settings** menu and scroll down to the **Find Devices** section.
3. Select **Enable Serial Device Discovery**.
   Additional options become available, including **Serial Parity**, **Serial Databits**, **Serial Stop Bits**, and **Baud Rates**.
4. Scroll down to see the options.
5. Select the Serial communication parameters according to your application.
   The default is 115200.

   Note: If you're not sure of the Serial baud rate, select **Enable Automatic Device Discovery** 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. In general it is better to disable it to increase discovery efficiency.
6. Click **OK** to return to Barcode Manager.
7. Click the **Getting Started** icon.
8. Open the **Serial Devices** tab.
9. Drag the **device** icon into the **Selected Device Information** area.
   The device is now connected to the Barcode Manager Configuration environment. Configure your device through Barcode Manager.

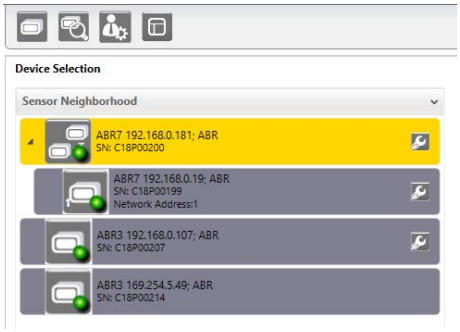
4.3 USB Device Discovery—3000 Models

The following configuration procedure assumes that a laptop computer running Barcode Manager is connected to an ABR 3000 USB model reader through the USB port using cable MQDEC-1703SS-USB or CSB-M121701USB02M121702.

ABR 3000 Ethernet models can also connect by USB. However, they do not show up with this device discovery method. For more information see *ABR 3000 Ethernet Model Connection via USB* available at www.bannerengineering.com.

1. After the reader is connected to the USB port and successfully starts, from Barcode Manager, click **Getting Started** to discover the reader.
   The reader is shown in the **Sensor Neighborhood** list.

*Figure 6. **Sensor Neighborhood** List*



2. Find your reader in the list by matching its serial number (SN). The USB driver creates a virtual Ethernet connection with an IP address that cannot be modified.

Tip: The USB unit is the one in the **Sensor Neighborhood** list without a **wrench** icon (used for modifying the IP parameters).

3. Double-click on or drag the **device** icon into the **Selected Device Information Area**. Details about the device display in this area.

Figure 7. Device Selection—Selected Device Details

Selected Device Details	
Name	ABR
Model	ABR3006-WSU2
Layout Type	Alone
Internal Network Role	Slave
Status	Default Running
Startup Info	OK
Application SW Version	1.5.4.645-f273f1
Loader Version	N/D

4.4 Decoding Configuration Parameters

The Decoding Configuration parameters are divided into two groups:

- Automatic Setp
- Advanced Setup

4.4.1 Automatic Setup

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

Note: For Manual Adjustable Focus models go to Advanced Setup for Manual Adjustable Focus Models on p. 13.

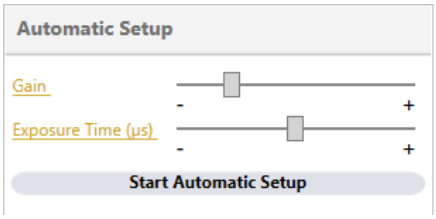
Automatic Setup provides an automatic procedure for setting optical/illumination and code definition parameters to obtain the most stable decoding conditions for a single code symbology based on the images presented to the reader. It can be set to include Image Filters if necessary. See the table below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

Enabled 1D Codes		Enabled 2D Codes	Enabled Filters
Code 128	GS1 DataBar Stacked	Data Matrix ECC 200	Erode 3×3, 5×5 and 7×7
EAN 128	GS1 DataBar Limited	QR	Dilate 3×3, 5×5 and 7×7
Code 39	GS1 DataBar Expanded	Micro QR	Smoothing
Code 93	GS1 DataBar Expanded Stacked	Aztec	
Codabar	UPCEAN Family EAN13	MAXICODE	
PDF417	UPCEAN Family EAN8	DOTCODE	
MICRO PDF417	UPCEAN Family UPCA		
GS1 DataBar	UPCEAN Family UPCE		

1. Click **Open Device Configuration**. The **Open Device Configuration** window opens showing the list of configurations (jobs) currently saved on the device. For new devices, the only saved configuration is the Default configuration.
2. Click **OK**. The device enters Live Image Capture mode and begins acquiring images.
3. Place the application code in front of the reader at the correct application reading distance.
4. If needed, focus the reader on the code.
5. After the code is positioned, click **Pause** to stop image acquisition.

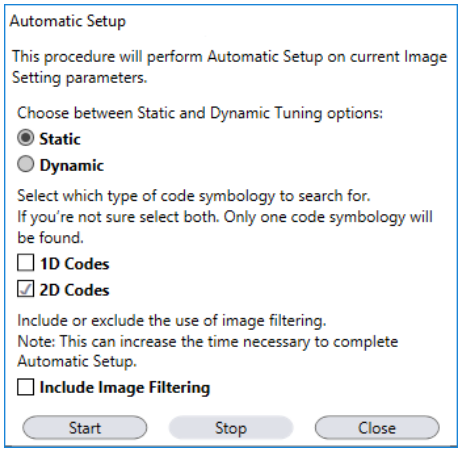
Note: If the image display area is too dark to see the images being captured, drag the Gain and Exposure Time sliders to the right to increase visibility. This will not affect Automatic Setup.

Figure 8. Gain and Exposure Time



6. Click **Start Automatic Setup**. The **Automatic Setup** window opens.

Figure 9. Automatic Setup



7. Select the correct reading conditions.
   - **Static** Tuning—No maximum limit on exposure time
   - **Dynamic** Tuning—Maximum allowable image exposure is automatically calculated using the parameters
   - **1D** code
   - **2D** code
   - **Include Image Filtering**—Select to find the best decoding condition.
8. Click **Start**.

   The reader begins acquiring images, adjusting the brightness and focus (for liquid lens autofocus models), and adjusting the decoding settings to find a barcode and optimize reading for the first code it finds. At the end of the procedure the Status: Completed message displays.

9. Close the **Automatic Setup** window.

   Your reader is now optimized for decoding. Continue setting up the reader for your application as desired. Typically, **Reading Phase** is configured next. See Reading Phase on p. 19.

4.4.2 Advanced Setup for Manual Adjustable Focus Models

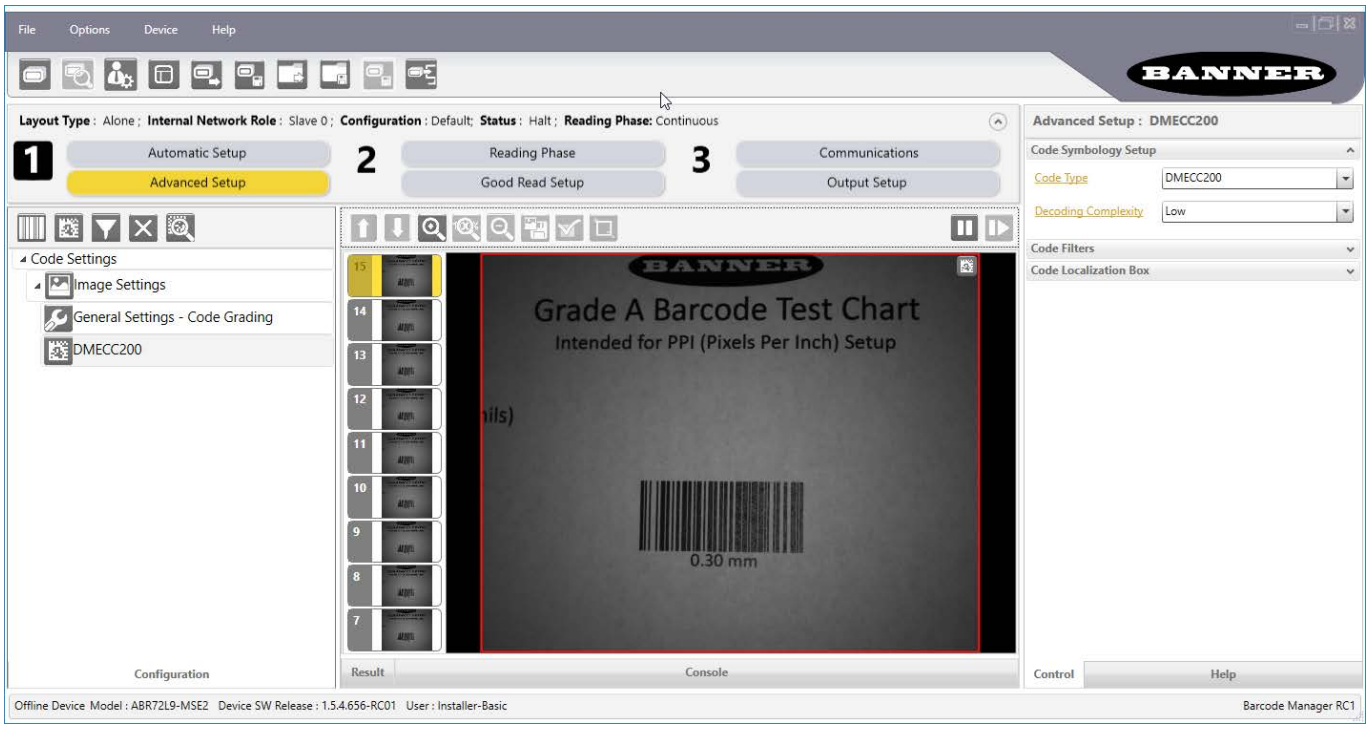
Advanced Setup provides access to the complete array of optical/illumination and code definition parameters that can be fine-tuned semi-automatically and manually to obtain the best results for applications of any complexity. If your application requires multiple code symbologies, multiple image settings, Code Grading, or other parameter settings for decoding, use the Advanced Setup.

To begin configuration, correctly mount the reader so that its Field of View (FOV) covers the application reading area.

Note: For Liquid Lens Autofocus models go to Advanced Setup for Liquid Lens Autofocus Models on p. 16.

1. From the **Task Area** select **Open Device Configuration**.
   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.
2. Click **OK**.
   The device enters run mode and begins acquiring images.
3. Click **Advanced Setup**.
4. Click the **Play** icon.
5. Place the PPI (Pixels per Inch) Setup Chart in the reading area. See PPI (Pixels Per Inch) Setup Chart on p. 161.
6. After the chart is positioned, click the **Pause** icon to stop image acquisition.

Figure 10. Chart Positioned

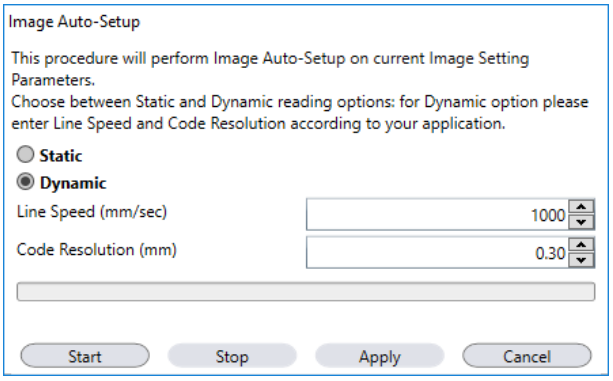


Note: If the image display area is too dark to see the images being captured, drag the Gain and Exposure Time sliders to the right to increase visibility. This will not affect Automatic Setup.

7. Click **Image Settings**.
8. Click Image Auto-Setup **Image Auto-Steup** to automatically acquire the best exposure time and gain values.
9. Select the reading option.
   - **Static** reading —No maximum limit on exposure time

- **Dynamic** reading—Maximum allowable image exposure is automatically calculated using the parameters **Line Speed** and **Code Resolution** to avoid image blurring

*Figure 11. **Image Auto-Setup** Window*

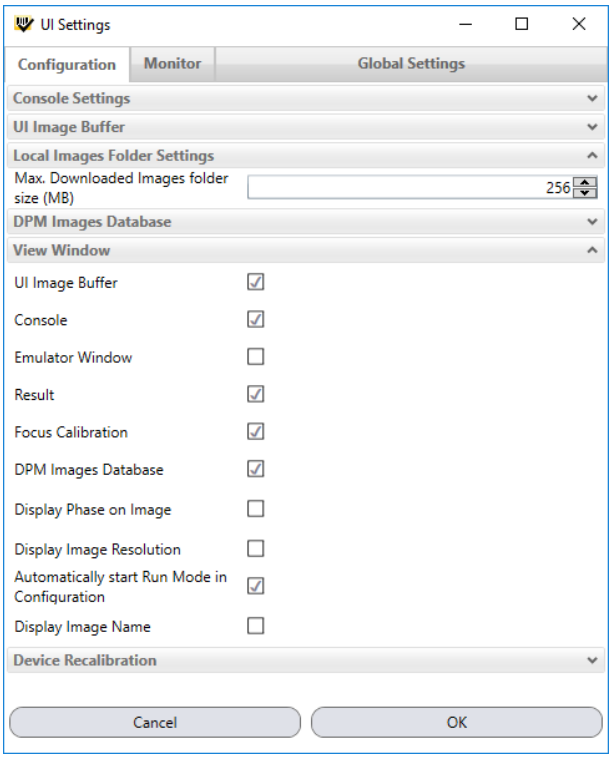


10. Click **Start**.
11. Click **Apply**.

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. See Multi Image Acquisition Settings on p. 28 for details.

12. From the main menu open **Options** > **UI Settings** > **Configuration** tab.
13. Select **Focus Calibration** under **View Window** if it is not already selected.

Figure 12. UI Settings

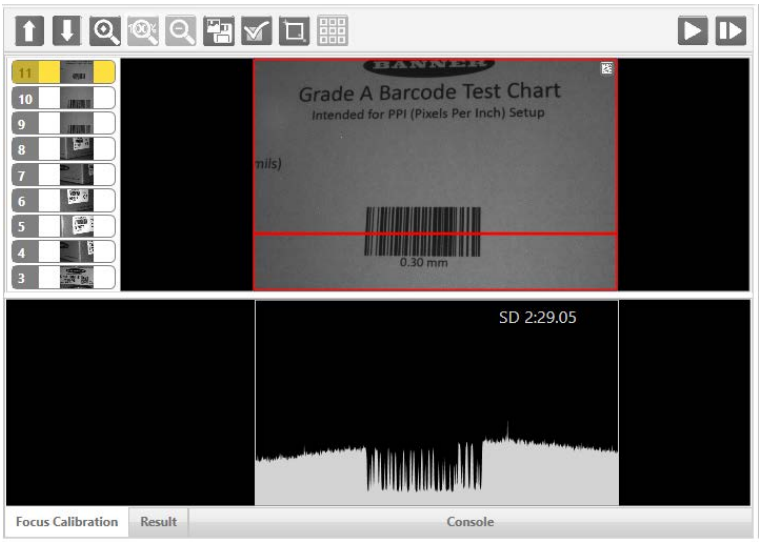


14. Click the **Focus Calibration** tab at the bottom of the window.

Note: This feature is only available for manual focus models.

The oscilloscope view is shown in the bottom panel and can be used for manual focus adjustment.

Figure 13. Oscilloscope View



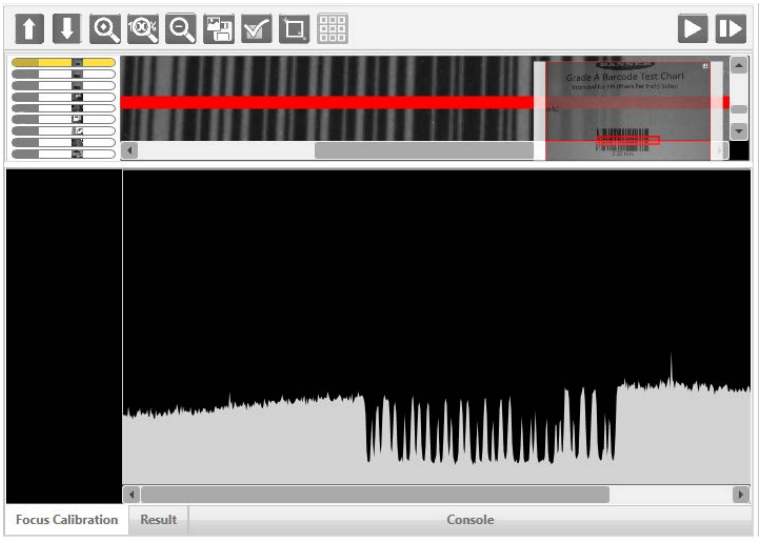
The red line in the image panel above the oscilloscope must pass through the code. Click and drag the red line vertically to reposition it over the code.

Note: To enlarge the visual image of the code and the oscilloscope views, drag the Focus Calibration window up and click 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.

Figure 14. Code View and Oscilliscope View Zoomed In



When focused, click **Pause** to stop image acquisition.

15. Click **Acquire PPI** to automatically set the Image Density so that the ABR functions correctly and to the fullest extent of its capabilities.

    This procedure is necessary to enable transmitting accurate barcode size estimates for barcodes at the same reading distance as the test card.

    Note: At this point it is good practice to save the configuration from temporary memory to permanent memory, giving it a specific name.

16. Place an application-specific code in front of the reader and only click **Image Auto-Setup** to register any changes in lighting or code surface contrast.

Do not repeat Focus Calibration or PPI.

17. Click 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 small green box around it indicating it is decoded.

Figure 15. Decoded Image



Note: The large green box for each symbol indicates the code localization area which by default is equal to the maximum FOV. Resize and move the box by dragging its borders with the mouse. The code must be found within this area in order to be decoded.

18. Add your application-specific codes to the Code Settings by selecting them from the icons over the Configuration Parameters tree area.
19. If the Data Matrix symbology is not used, delete it from the Code Settings with the icon.
20. If you don't know the code type, use the Code Autolearn feature by clicking on the icon [5].
21. For each code symbology set the relative parameters according to your application.

Continue the configuration using Reading Phase on p. 19.

4.4.3 Advanced Setup for Liquid Lens Autofocus Models

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

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

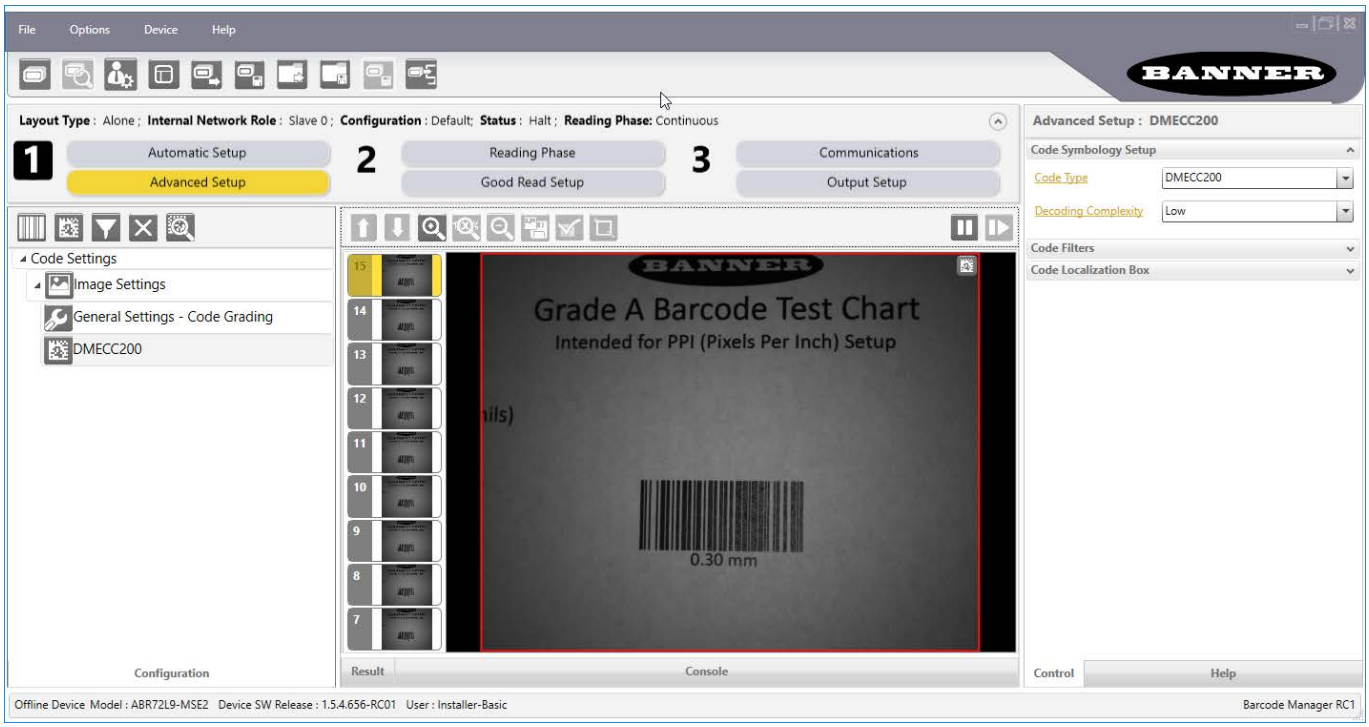
Note: For manual adjustable focus models go to Advanced Setup for Manual Adjustable Focus Models on p. 13.

1. From the **Task Area** select **Open Device Configuration**.
   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.

[5] The Code Autolearn procedure does not recognize the following symbologies: Pharmacode, MSI, Standard 2 of 5, Matrix 2 of 5.

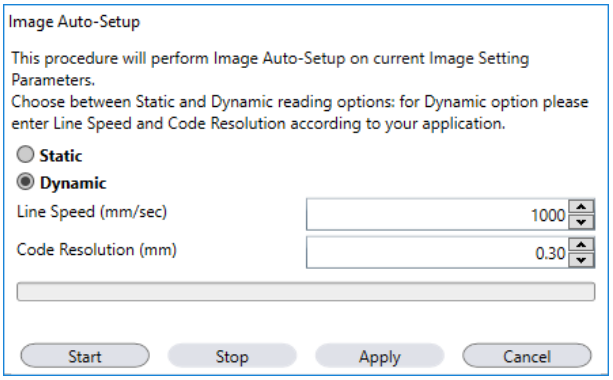
2. Click **OK**.
   The device enters run mode and begins acquiring images.
3. Click **Advanced Setup**.
4. Click the **Play** icon.
5. Place the PPI (Pixels per Inch) Setup Chart in the reading area. (See PPI (Pixels Per Inch) Setup Chart on p. 161.)
6. After the chart is positioned, click **Pause** to stop image acquisition.

Figure 16. Chart Positioned



7. Click **Image Settings**.
8. Click **Image Auto-Setup** **Image Auto Setup** to automatically acquire the best exposure time and gain values.
9. Select the reading option.
   - **Static** reading—No maximum limit on exposure time
   - **Dynamic** reading—Maximum allowable image exposure is automatically calculated using the parameters **Line Speed** and **Code Resolution** to avoid image blurring

*Figure 17. **Image Auto-Setup** Window*



Note: For applications having multiple lighting or code reading conditions, up to 10 different Image Setups can be configured by adding them with the icon. See Multi Image Acquisition Settings on p. 28 for details.

10. Click **Start**.

11. Click **Apply**.
12. Click Focus Autolearn **Focus Autolearn**.

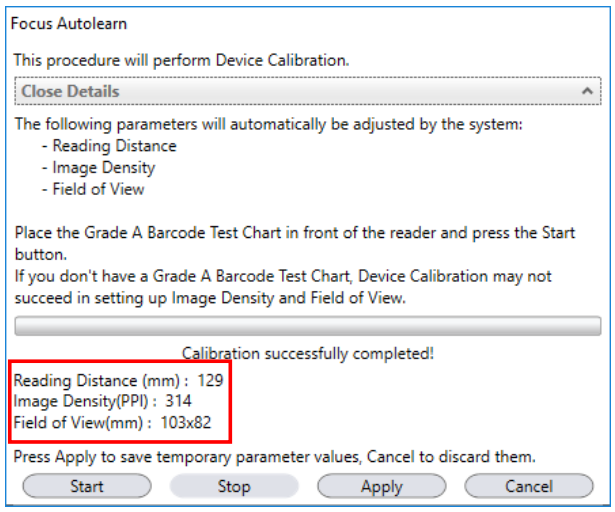
Tip: You may have to click **Image Settings** again before you can click **Focus Autolearn.**

Note: The **Reading Distance** value is not significant until the Focus Autolearn procedure ends successfully.

The Calibrate dialog box opens allowing you to start the procedure.

13. Click **Start**.
    At the end of the calibration you can see the new Reading Distance and Image Density (PPI) values as well as the FOV dimensions.

Figure 18. Reading Distance, Image Density, and FOV Dimensions



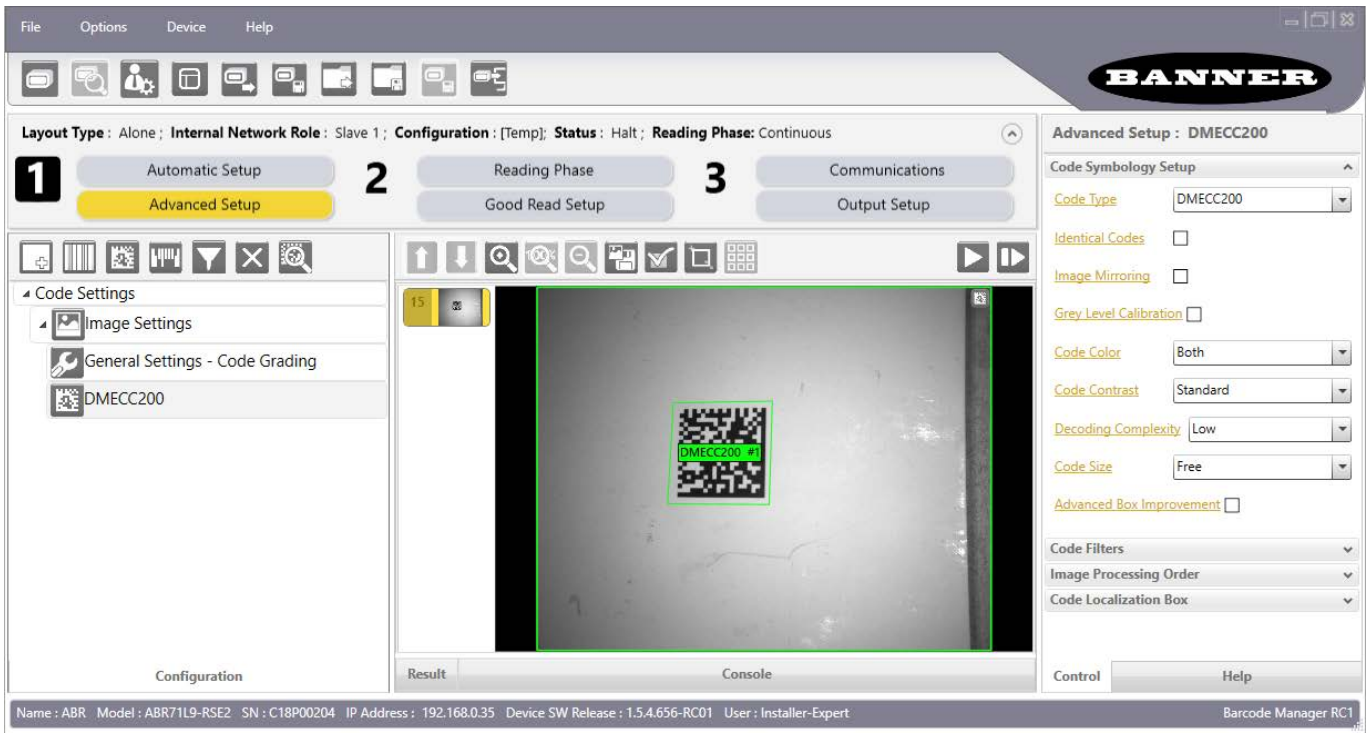
14. Click **Apply**.

Note: To enlarge the visual image of the code view, click the zoom image icon, repositioning it on the code.

Note: At this point it is good practice to save the configuration from temporary memory to permanent memory, giving it a specific name.

15. Place an application-specific code in front of the reader and only click **Image Auto-Setup** to register any changes in lighting or code surface contrast.
16. Click 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 small green box around it indicating it is decoded.

Figure 19. Decoded Data Matrix ECC 200



Note: The large green box for each symbol indicates the code localization area which by default is equal to the maximum FOV. Resize and move the box by dragging its borders with the mouse. The code must be found within this area in order to be decoded.

17. Add your application specific codes to the Code Settings by selecting them from the icons over the Configuration Parameters tree area.
18. If the Data Matrix symbology is not used, delete it from the Code Settings with the icon.
19. If you don't know the code type, use the Code Autolearn feature by clicking on the icon [6].

Continue the configuration using Reading Phase on p. 19.

4.5 Operating Mode Configuration Parameters

The Operating Mode Configuration parameters are divided into two groups:

- Reading Phase parameters
- Good Read Setup parameters

4.5.1 Reading Phase

1. Click Reading Phase **Reading Phase**.
2. Select your application-specific Operating Mode from the icons over the Configuration Parameters tree area:
   - Continuous
   - One Shot
   - Phase Mode

   Continuous Mode and **Acquisition Trigger** are shown by default.
3. Configure the relative **Operating Mode** parameters from the **Reading Phase** Control panel.
   Different groups appear in the panel depending on the selected icons over the **Configuration Parameters** tree area.

[6] The Code Autolearn procedure does not recognize the following symbologies: Pharmacode, MSI, Standard 2 of 5, Matrix 2 of 5.

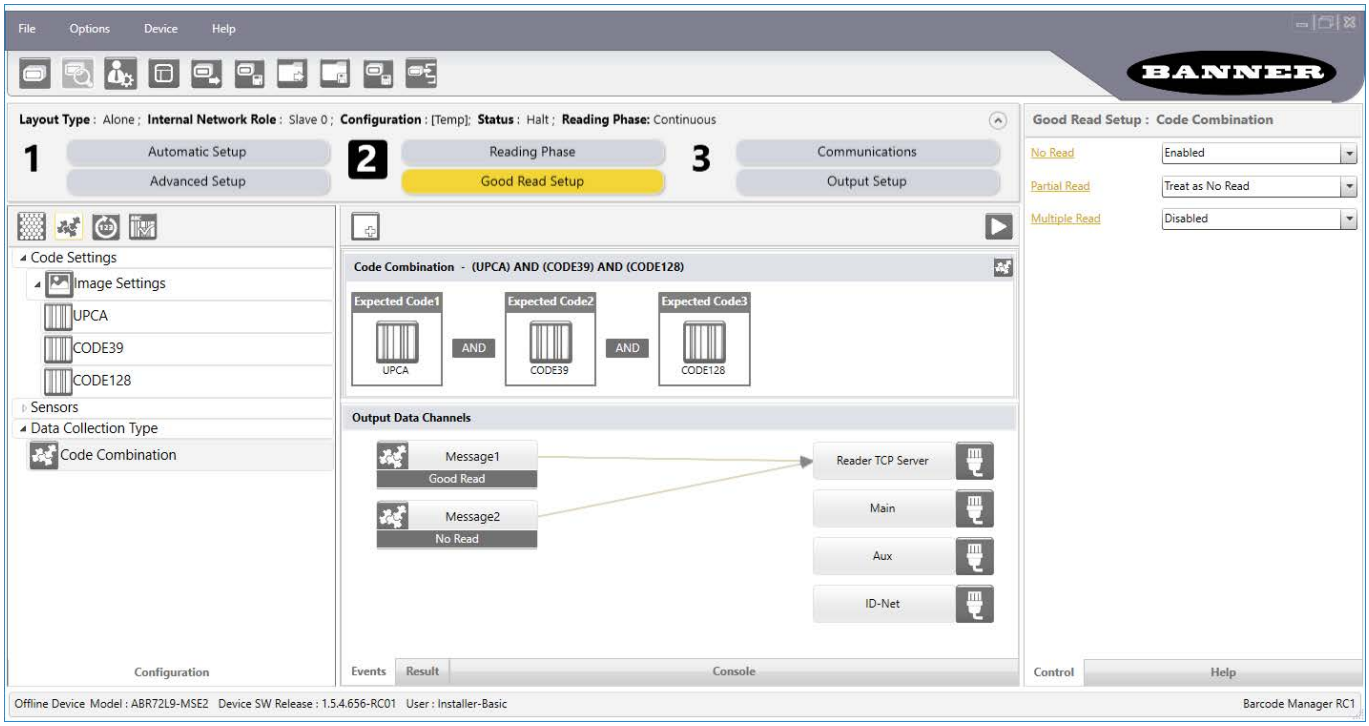
4.5.2 Good Read Setup

1. Click Good Read Setup **Good Read Setup**.
2. Select your specific data collection type from the icons over the Configuration Parameters tree area:
   - Code Collection
   - Code Combination
   - Code Presentation
   - Match Code

   Not all data collection types are available for all Operating Modes. Incompatible data collection types are shown in gray and cannot be selected.

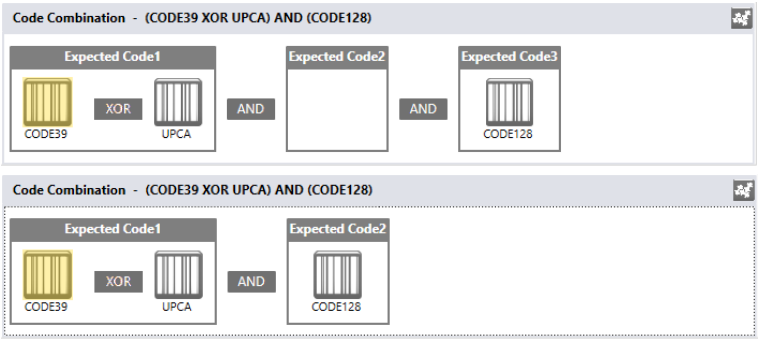
   The following example shows **Code Combination**. By default, the Expected Codes (when more than one code type is selected), are in logical AND, which means that all codes are required to be decoded to produce a Good Read condition.

Figure 20. Good Read Setup: Code Combination



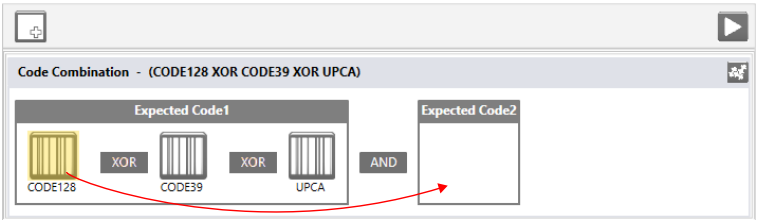
3. If a Good Read condition should be produced when any single code is decoded, independent from the others, combine them in logical XOR.
   a) Drag the code icon(s) from their relative Expected Code box into the Expected Code box of the XOR combination you wish to create.
   b) Delete the empty box by selecting it with the mouse (highlighted) and pressing **delete** on your keyboard.

Figure 21. Code Combination



c) To create a logical AND condition from a logical XOR, create a new Expected Code box using the icon.
d) Drag the desired code icon from one box to the other.

Figure 22. New Expected Code



4.6 Output Configuration Parameters

The Output Configuration parameters are divided into two groups:

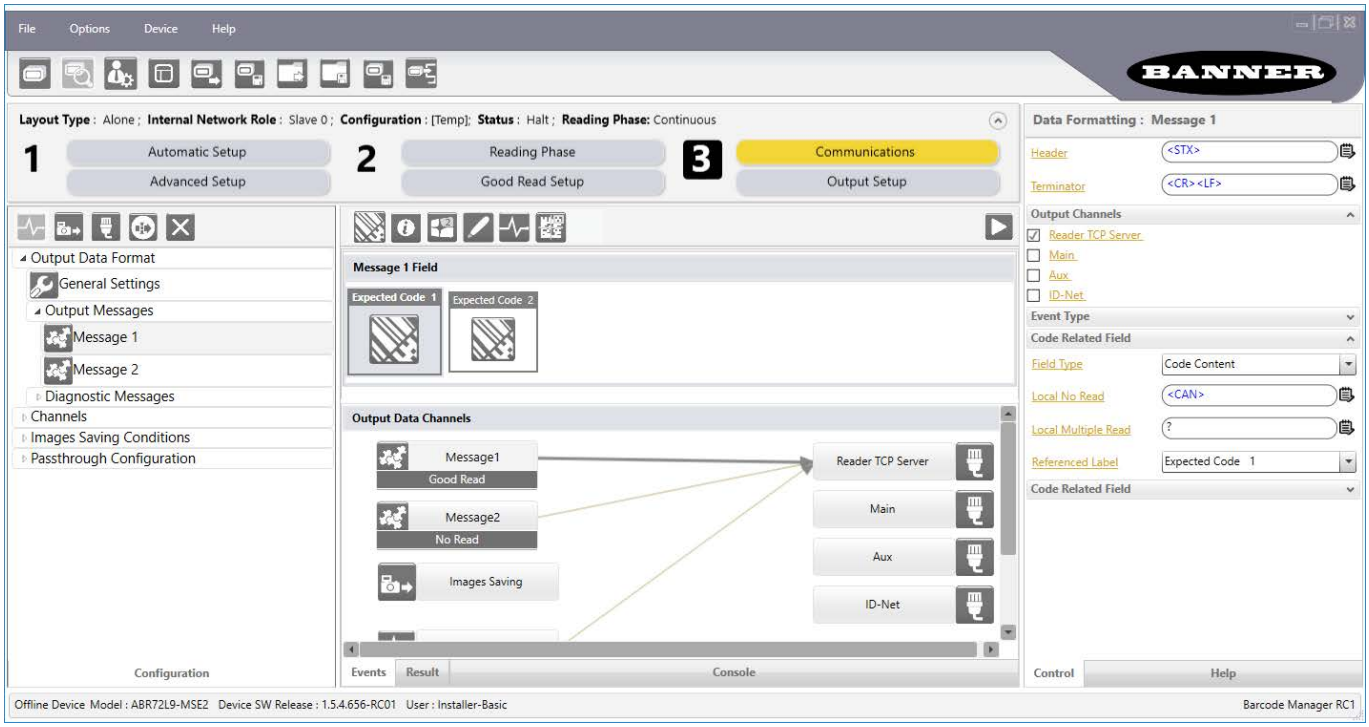
- Communications parameters
- Output Setup parameters

4.6.1 Communications

For details, see Message Formatting on p. 64.

1. Click **Communications**.
2. Configure your application-specific Output Data Message(s) from the Configuration Parameters tree area: Message 1, Message 2, etc.

Figure 23. Communications



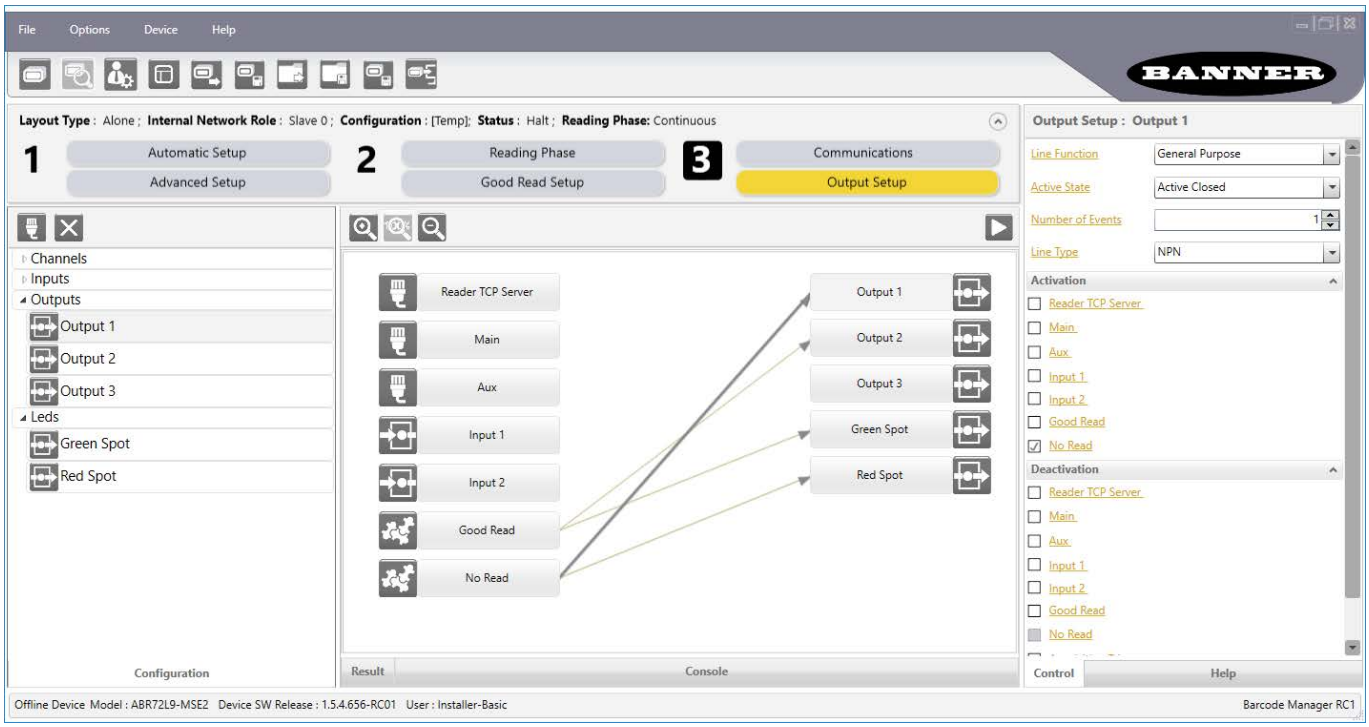
3. Add fields to the output message by clicking on the icons above the Message Field area.
   The fields are appended to the message.
4. Drag the fields to position them between other fields in the message so that the output message is ordered according to your application requirements.

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

4.6.2 Output Setup

1. Configure your application-specific Digital Output(s) and Green/Red Spots (if used) from the Configuration Parameters tree area: Output 1, Output 2, etc.

Figure 24. Output Setup



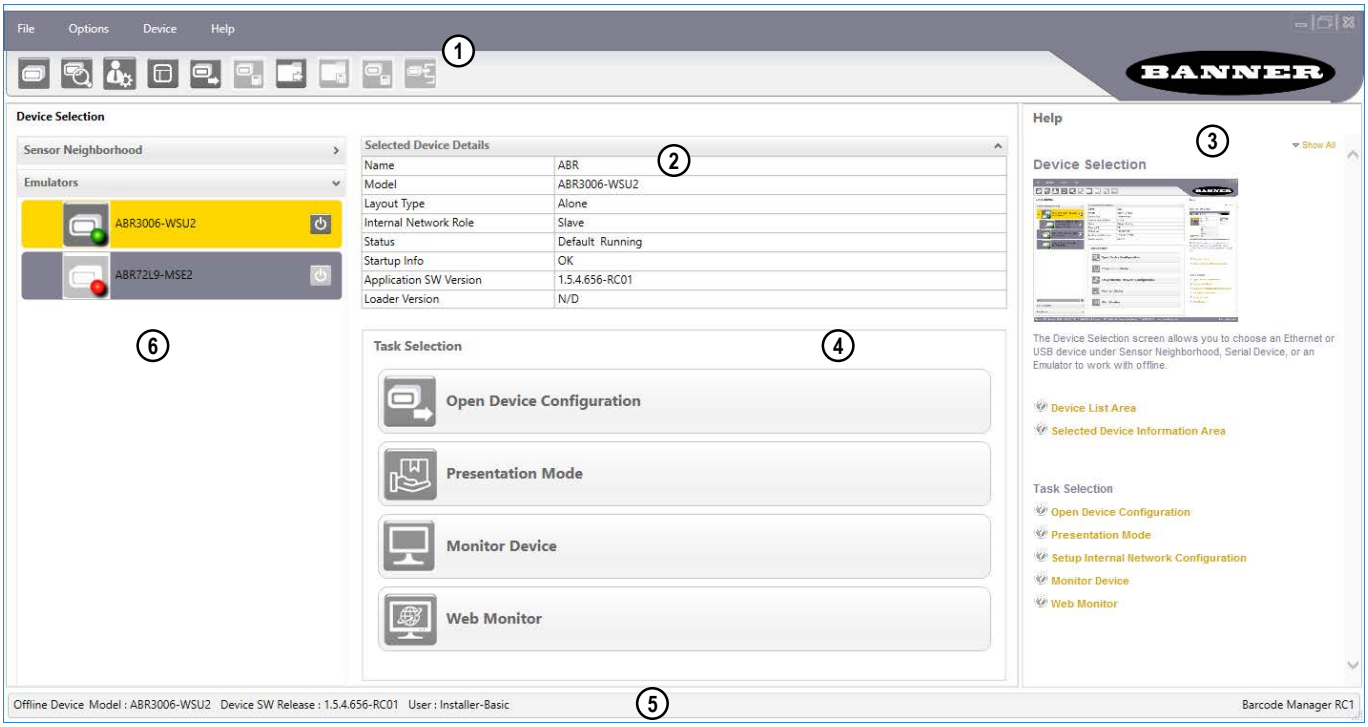
2. Save the configuration from temporary memory to permanent memory, overwriting the previously saved configuration.

5 User Interface

5.1 Barcode Manager Device Discovery Window

After loading your specific device from the discovery window (drag & drop from the **Device Selection** area), the Barcode Manager Device Discovery window includes the following principal areas:

Figure 25. Main Window Area/Device Discovery Window Areas



1. **Main Menu and Toolbar Area**— Allows access to the major program functions and commands. See Barcode Manager Main Menu and Toolbar on p. 25.
2. **Selected Device Information Area**—This area displays device-specific information: Name, Model, Role, running software and version details.
3. **Control/Help Panel**—In the Barcode Manager opening window this panel provides a Help description for Device Selection. After 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 p. 24.
4. **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, or switch to Monitor mode. These selections are also available in the File and Device menus.
5. **Status Bar**—A reserved area that keeps specific information about the connected device: Name, Model, Serial Number, IP Address, Device SW version, user level, and Barcode Manager version. This information is always visible.
6. **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 p. 23. Double-clicking or dragging an available device icon onto the Device Information Area connects it to Barcode Manager for configuration.

5.1.1 Device List Area

The Device List area shows all the discovered devices both on and off the LAN.

The icons indicate the device status.

	Stand Alone device connected to the network with a valid LAN IP address. Available for configuration through Barcode Manager. Stand Alone device connected to Barcode Manager via Serial COM Port.

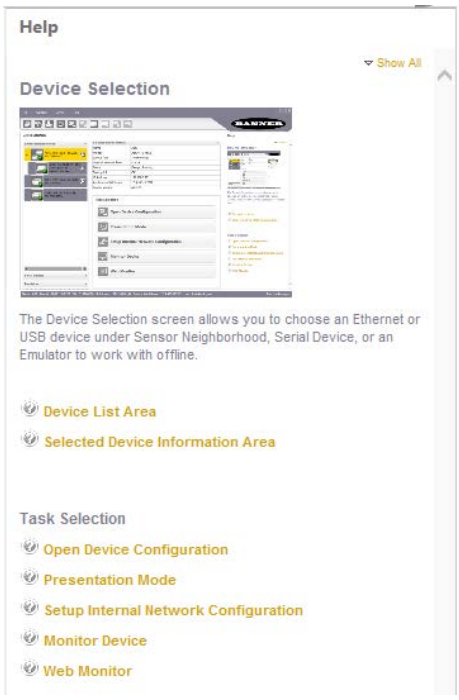
	Stand Alone device connected to the network with a valid LAN IP address but currently connected to Barcode Manager by another user. Not available for configuration through Barcode Manager (double-clicking or dragging the icon has no effect).
	Device connected to the network but without a valid LAN IP address (that is, default IP address). The IP addressing parameters must be modified to connect to the LAN. See Ethernet Device Discovery on p. 9. Device connected to a Serial COM port but not to Barcode Manager.
	Offline device deactivated. Only one offline device be activated at a time can.
	Master device connected to the network with a valid LAN IP address. Available for configuration through Barcode Manager.
	Slave device # connected to the Master via ID-NET and to the network with a valid LAN IP address. Available for configuration directly through Barcode Manager.
	Slave device # connected to the Master via ID-NET network but not to the LAN. Available for configuration through the Master connected to Barcode Manager.
	Slave device # not connected to the Master via ID-NET. Not available for configuration.

5.1.2 Control/Help Panel

Upon opening Barcode Manager, the Device Selection Help Page displays with dropdown descriptions briefly explaining how to connect to your device and the various configuration selections. Click the  icon to open the dropdown description.

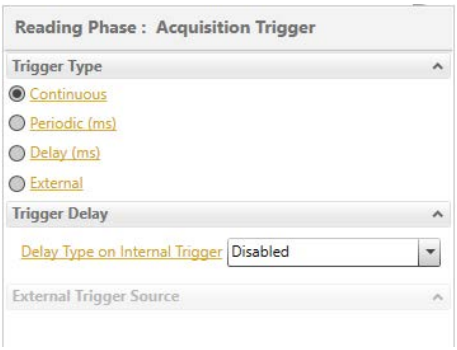
Click **Show All** at the top of the **Help** page to open all collapsed text descriptions.

Figure 26. Help Panel



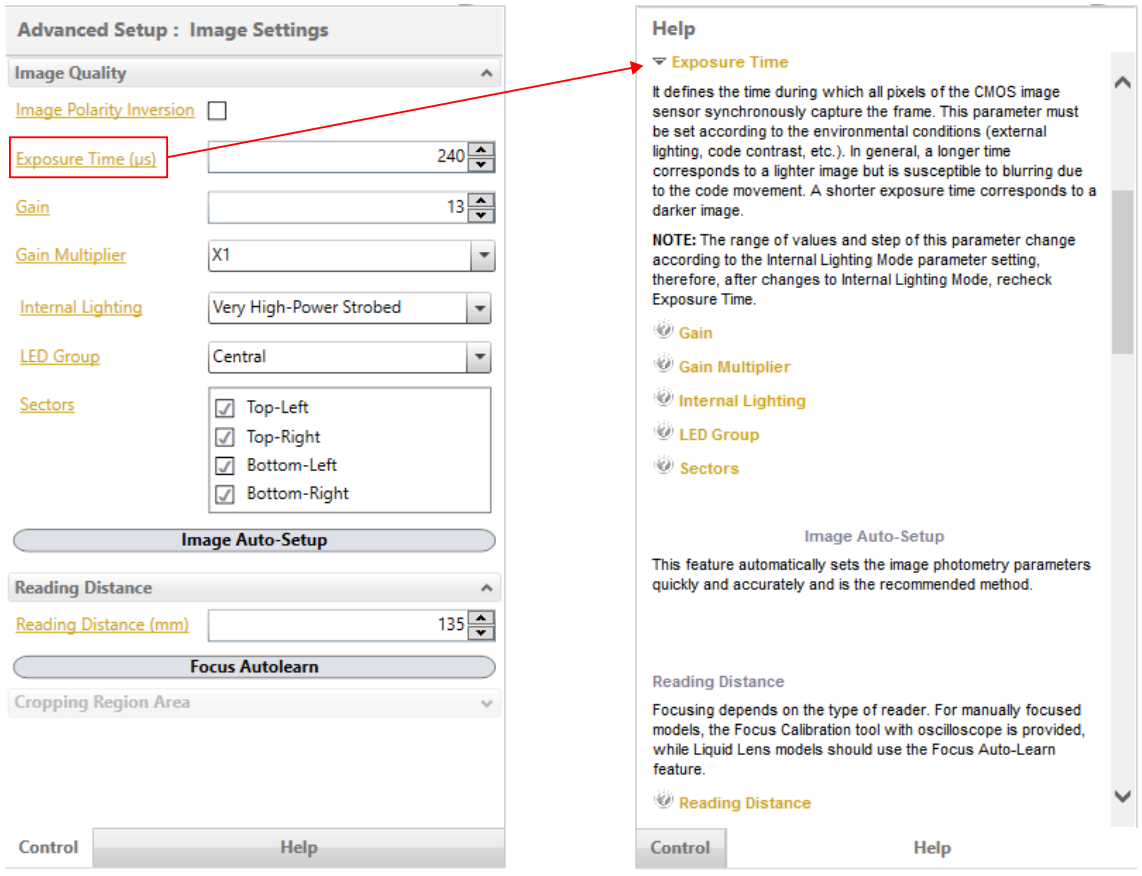
When a new configuration is created or an existing one is opened, the Control page displays in the Help/Control area and shows all of the configuration parameters for the selected configuration step or item.

Figure 27. Configuration Parameters in the Control/Help Area: Reading Phase Example



Clicking on any parameter name with a hyperlink, opens the relative contextual help page and shows the specific parameter description.

Figure 28. Control Panel and Relative Contextual Help Page



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

5.2 Barcode Manager Main Menu and Toolbar

The Main menu and the Toolbar icons are located at the top of the Barcode Manager window.

Figure 29. Main Menu and Toolbar Icons



The Main Menu includes 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 on p. 128 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 on p. 114 for details on using and configuring the Monitor.
- **Exit**—Exits the Barcode Manager user interface.

Options

- **Change Language**—Changes the display language used for Barcode Manager in real time. The selected language is used for successive sessions.
- **UI Settings**—Opens a window where various settings can be made regarding Barcode Manager presentation on the PC. The following Barcode Manager features can be configured: console presentation, Image Buffer positioning and behavior, Emulator 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**—Changes the configuration log level among Verbose, Information, and Error.
- **Change User**—Allows the configuration access level to be changed between Basic-User (no parameter 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 Environment Settings on p. 156) , Advanced Configuration 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 the Important Note, below.
- **Multi-Device Package Update**—Allows updating multiple devices simultaneously. This command is only available when no device is currently connected to Barcode Manager. Update firmware (application program, schema, etc.) either upgrading or downgrading to a different version. See also the Important Note, below. After selecting the software 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.

Important: Before performing any downgrade of device firmware, always restore the reading device default environment parameters using the Smart Teach button procedure on the reading device, (see the device Instruction Manual). Alternatively, execute this command from the Barcode Manager Device menu: **Backup/Restore** > **Restore Defaults** > **Restore Default Environment**.

- **Change Current Configuration**—Opens a window that lists all configuration files currently on the device, and allows selecting a different file to run.
- **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.

Help

- **About**—Opens the information window containing the Barcode Manager program release version number.
- Barcode Manager **User's Guide** —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 Barcode Manager 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—Saves any new unsaved settings to a config file on the reader named `[Temp]` that is not retained over a power cycle.
	Switch to Configuration/Monitor—Toggles between the Configuration environment and the Monitor environment. See Monitor on p. 114 for details.

5.2.1 User Levels

Barcode Manager has a 3-level user interface with the following access rights:

- User-Basic—Only access to the Monitor feature is given. No device configuration.
- Installer-Basic—A subset of program features and configuration parameters is allowed. A Basic Installer does not have access to: Master/Slave configuration, multiple Image Settings configurations, Postal Codes configuration, or Industrial Ethernet configuration.
- Installer-Expert—Access is given to all program features and configuration parameters.

Access each level by selecting it from the **Options** menu.

5.3 Multi Image Acquisition Settings

When controlled variable conditions occur in the application, define Multiple Image Acquisition Settings 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.

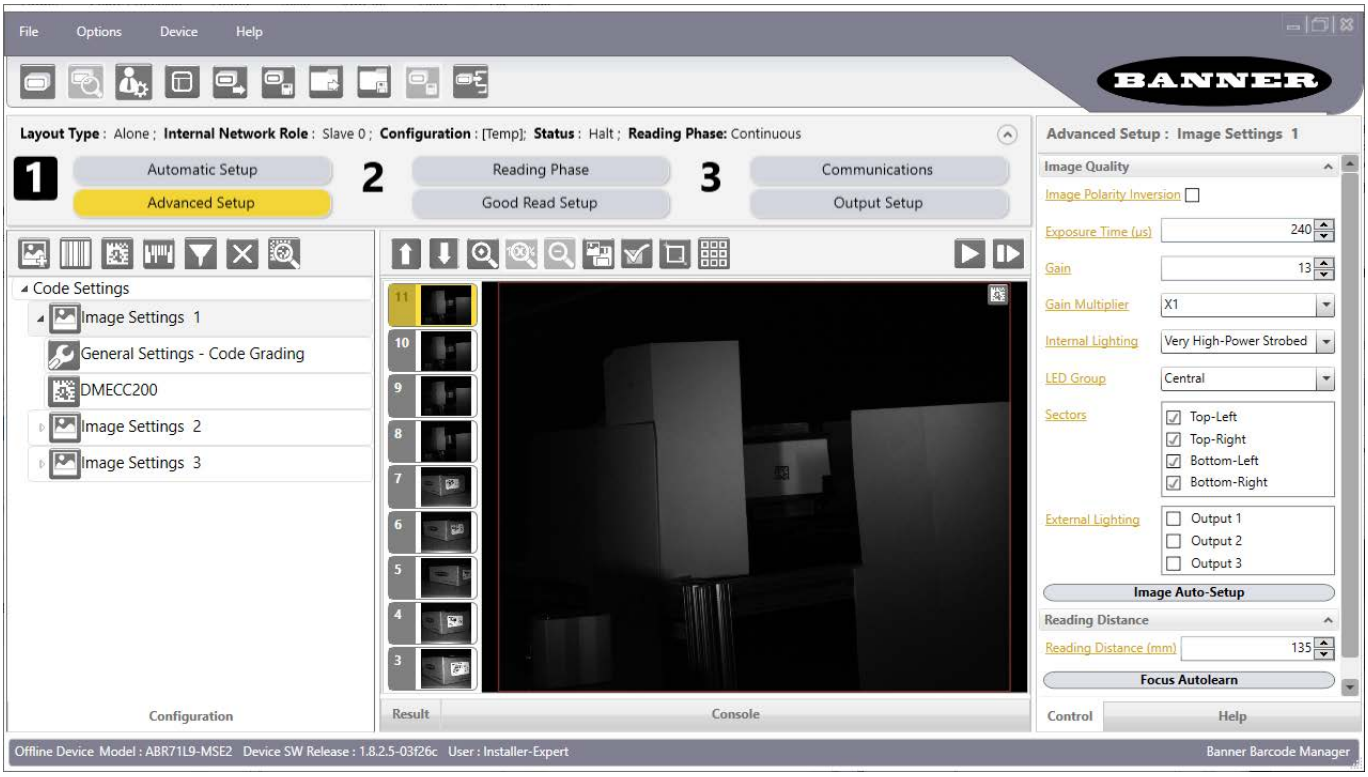
For example, an application may have several stable but different lighting conditions which require different lighting options. One Image Acquisition Setting could enable and use an internal illuminator and another setting could enable and use an external lighting system.

This feature is available for all Operating Modes.

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

Note: The is available to **Installer - Expert** users.

Figure 30. Advanced Setup—Image Settings



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

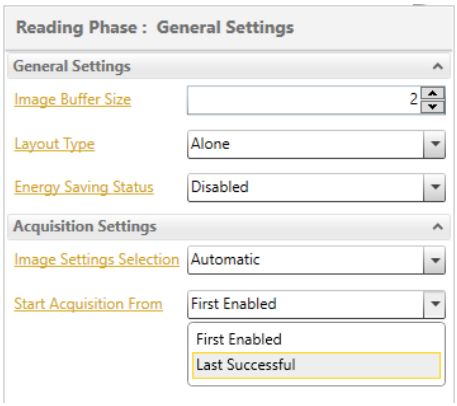
Note: The Automatic Setup procedure can be used for each **Image Setting** that is added through the **Advanced Setup** group, however only one code will be associated with each **Image Setting**.

5.3.1 Automatic Image Settings Selection

If it is not known from one item to the next which reading condition will be presented, the reader cycles through the pre-defined database of Image Settings (one per acquisition) 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 are used cyclically.

Figure 31. Reading Phase—General Settings



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

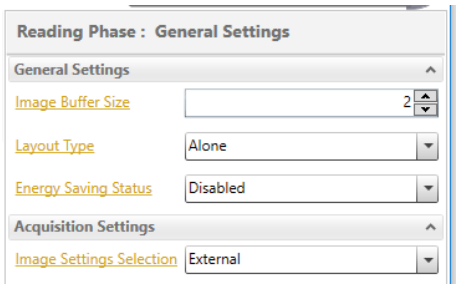
Group items together that have the same lighting conditions to improve the read rate through the **Start Acquisition From** parameter. By choosing the **Last Successful** value, we will start with the Image Setting that last produced a Good Read. For this group of items the last Image Setting used will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

5.3.2 External Image Settings Selection

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

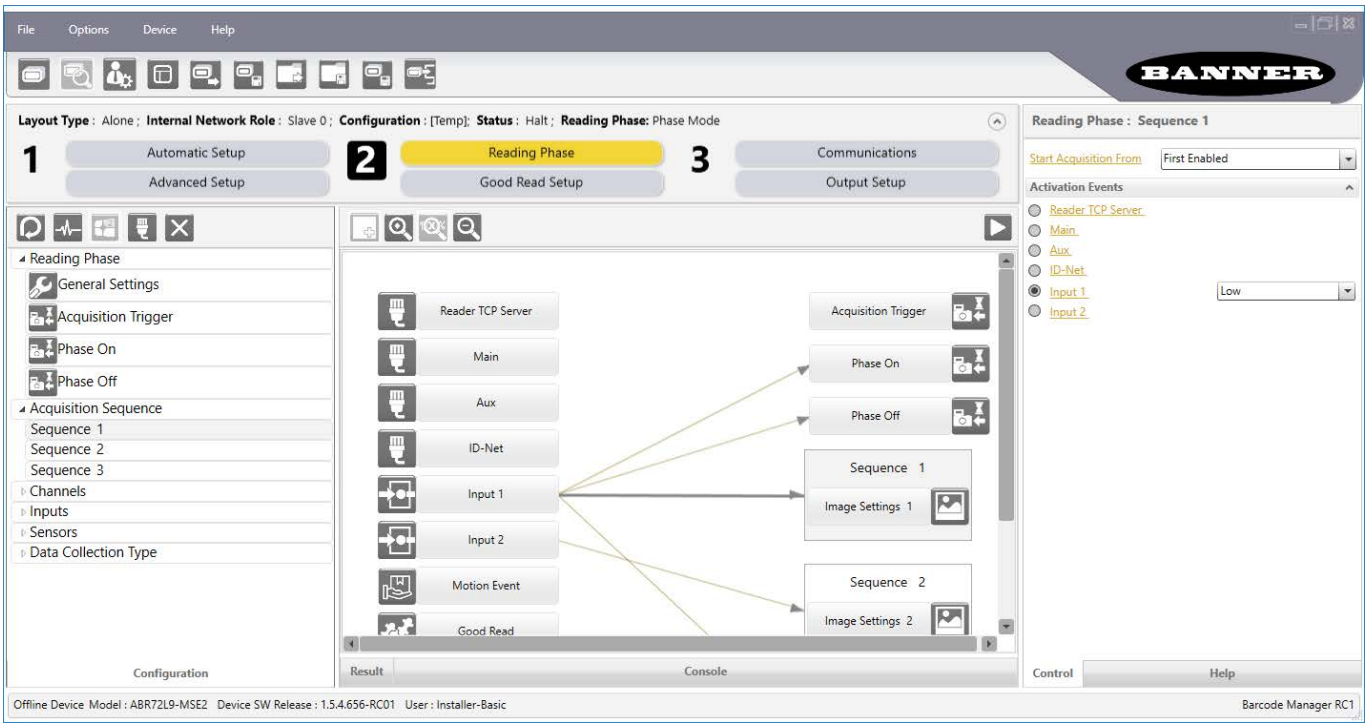
When the **Image Settings Selection** is External, Acquisition Sequences are created and by default each Image Setting has its own Acquisition Sequence.

Figure 32. Image Settings Selection: External



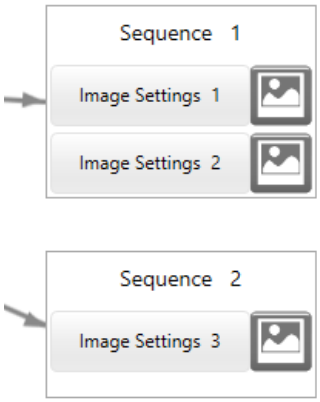
Each **Acquisition Sequence** can be activated exclusively by a single event, either through a string from an available communication channel or by a digital input.

Figure 33. Acquisition Sequence



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.

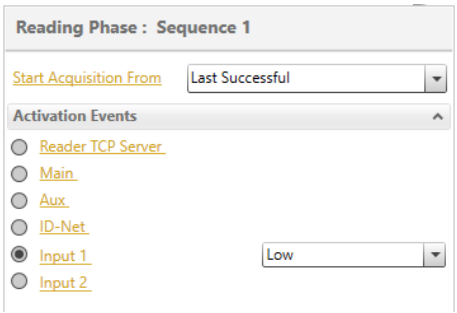
Figure 34. Multiple Image Settings in one Acquisition Sequence



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.

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.

*Figure 35. **Start Acquisition From** Parameter*



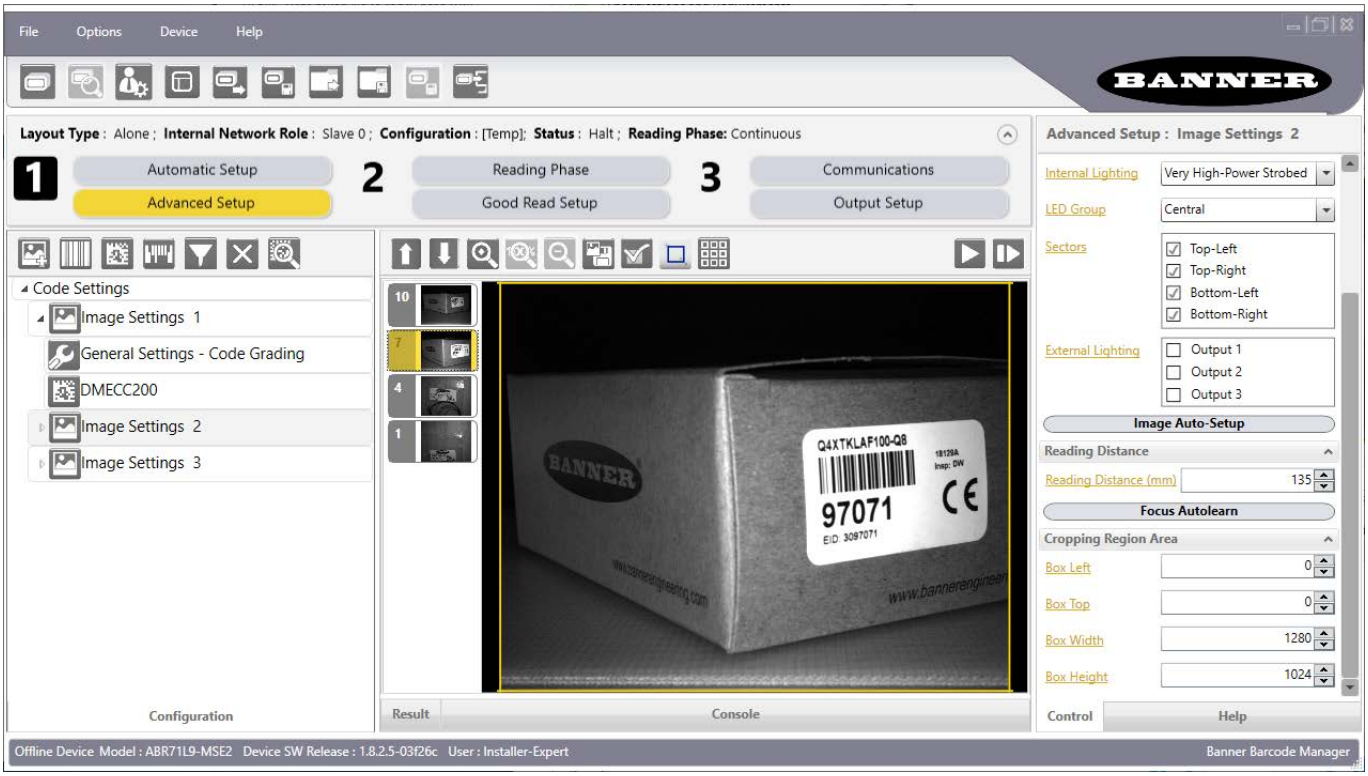
5.4 Image Cropping

In some applications, the Image Cropping feature in Barcode Manager can help to increase decoding performance. Image cropping is performed from the **Advanced Setup** tab by clicking on the **Add Cropping Region** icon.

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, acquisition time is reduced.

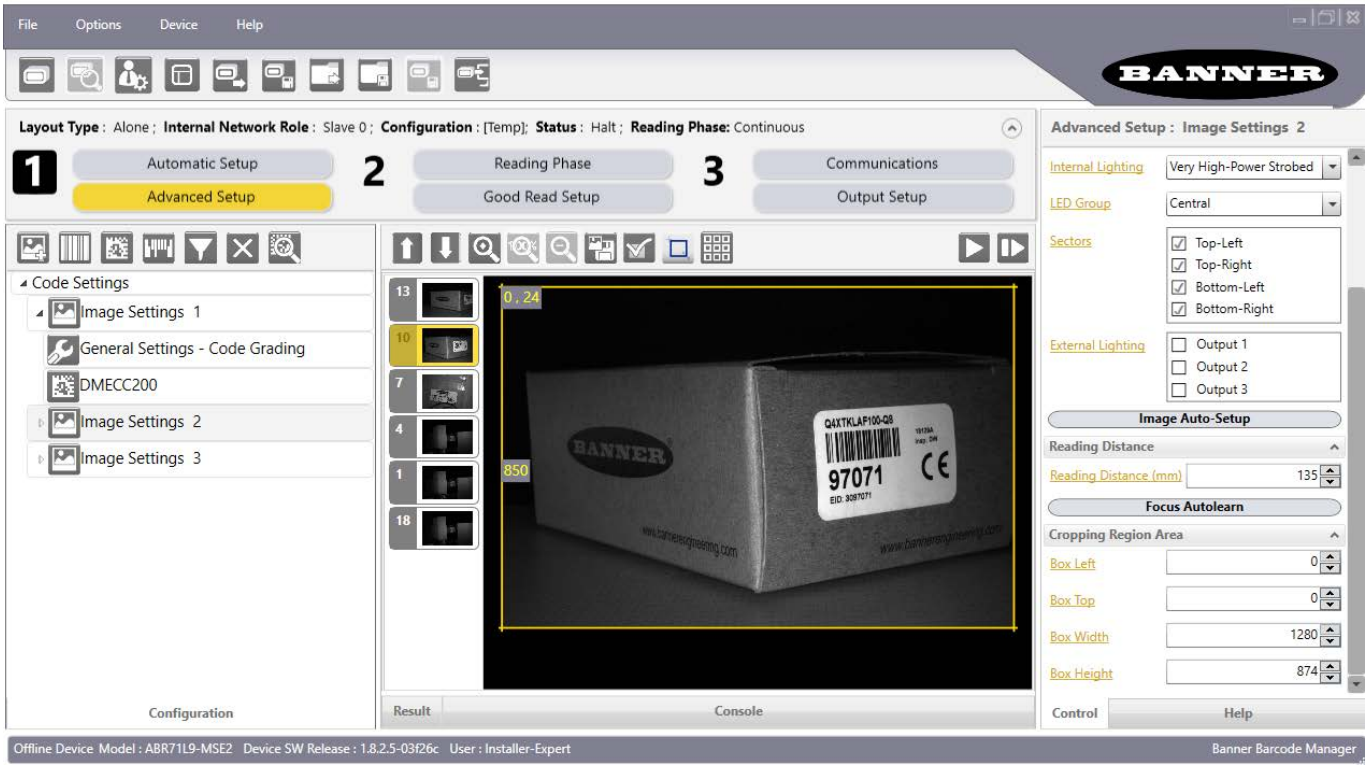
After clicking the **Add Cropping Region** icon, a yellow box appears which defines the outer limits of the FOV.

Figure 36. Yellow Box for Cropping



Resize the yellow box to crop the image to a specific location where codes are present. The numbers in the yellow boxes refer to pixel references.

Figure 37. Image Cropping—Coordinates and Pixels



Move the cropped area by dragging its center.

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.

Figure 38. Cropping Region Area



Note: Decreasing the **Box Height** causes a roughly proportional decrease to acquisition time, but changes to **Box Width** only have a minimal effect on acquisition time.

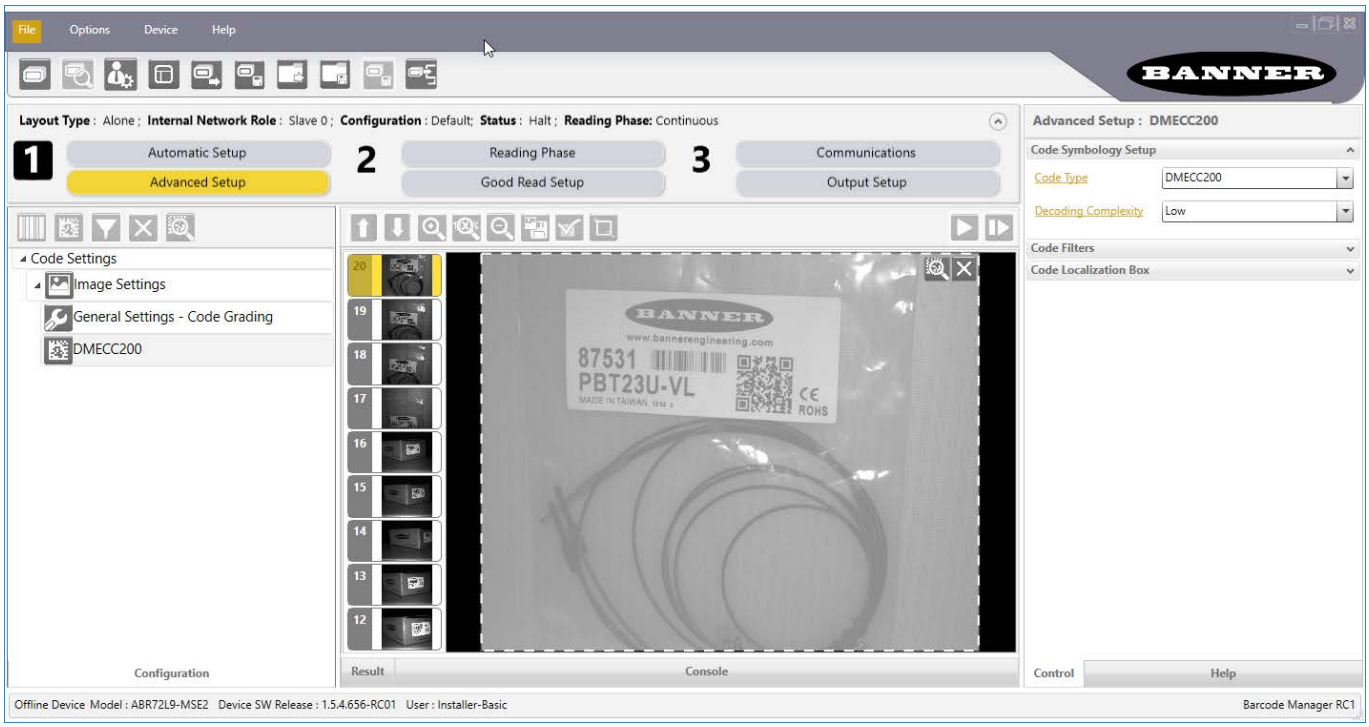
5.5 Code Autolearn Feature

From the **Advanced Setup** page, run the Code Autolearn feature which recognizes all the codes present in the captured image.

1. From the **Advanced Setup** page click the **Learn Codes** icon and select **Code Autolearn**.

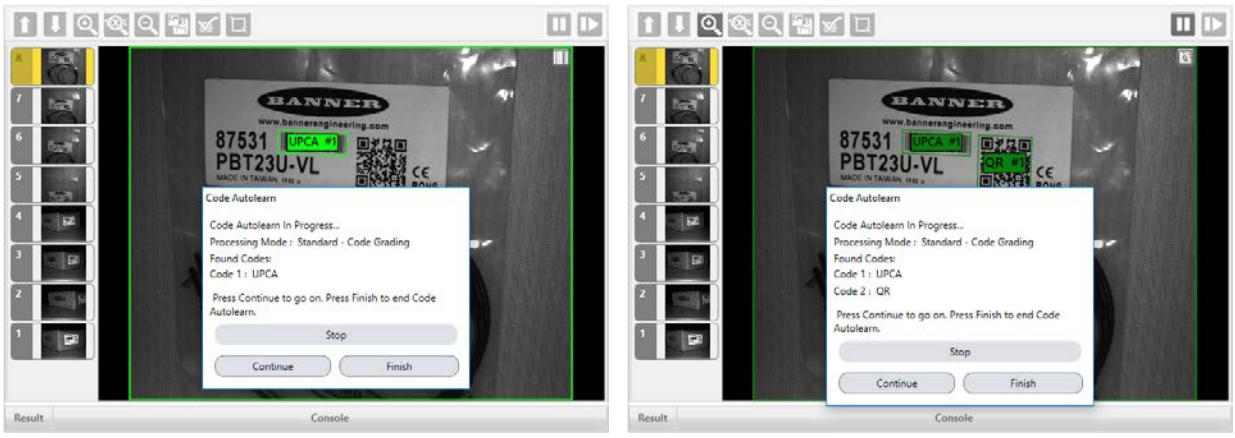
   The Autolearn region (equal to the FOV) is shown in gray. Reduce and/or move the search area by dragging the borders or the center of the area.
   - Reducing the area can reduce the time necessary to find the code(s)
   - Moving the search area allows finding specific code(s) in the image

Figure 39. Autolearn Area



2. Whether the area is reduced or not, start the Autolearn feature by clicking the **Learn Codes** icon in the display area.
   Each autolearn iteration locates a single code symbology. You will be prompted to Continue (if you need to find other codes) or to Finish.

Figure 40. Code Autolearn - Finish and Code Autolearn - Continue



3. When you have located all the code symbologies, click **Finish**.
4. Choose a saving selection and click **Ok**.
   - Add to Current Configuration
   - Replace Current Configuration
   - Discard Autolearn Result

5.6 Image Filtering

Image Filtering is typically done in Direct Part Marking (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 sections detail the DPM parameters used to enhance decoding capabilities.

5.6.1 Direct Part Marking (DPM) Applications

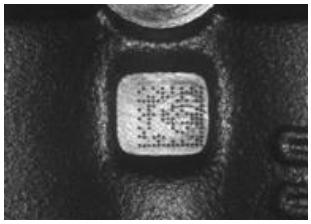
For Data Matrix family codes the Decoding Complexity parameter is available when Processing Mode is set to Standard. This allows selection of the decoding algorithm that best suits the printing/marking technique used to create the symbol and the overall printing/marking quality.

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.**

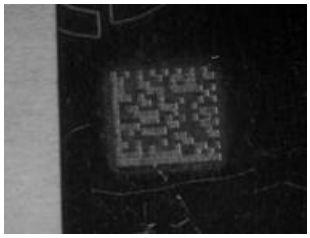
Figure 41. Problematic Direct Part Marking Examples



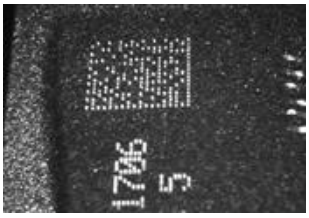
Washed out and Axial Distortion



Dot Peening On Scratched Surface



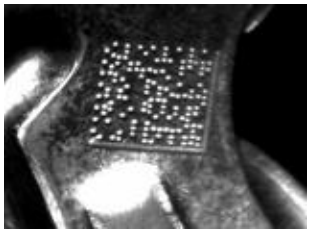
Low Contrast Problem



Background Problems



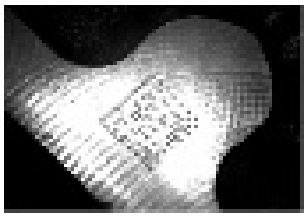
Marked On Curved Shiny Surface



Axial distortion



Half moon effects



Shiny surface, noisy background



Low contrast, noisy background

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

5.6.2 Image Filters

In Barcode Manager, 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 Set an Image Filter: Example 1 on p. 38 and Set an Image Filter: Example 2 on p. 40 for examples.

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

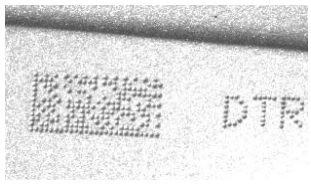
The Image Filter sets the filter to be applied to the image before being processed. This parameter can be used to successfully decode particular ink-spread printed codes (for example, direct part mark codes).

Different filters can be applied to a single code or group of codes in one or more Image Settings. See Set an Image Filter: Example 1 on p. 38 and Set an Image Filter: Example 2 on p. 40.

Erode

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

Figure 42. Erode Filter Examples



Before - No Read

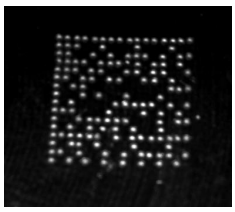


After - Readable

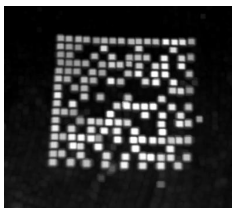
Dilate

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

Figure 43. Dilate Filter Examples



Before - No Read

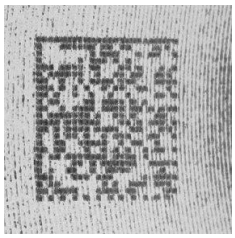


After - Readable

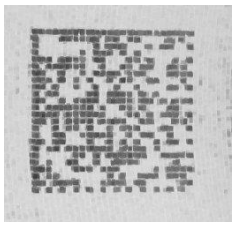
Close

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

Figure 44. Close Filter Examples



Before - No Read

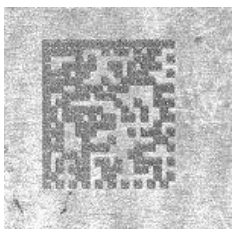


After - Readable

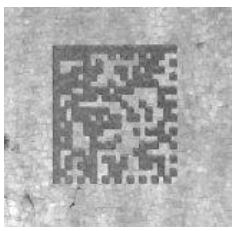
Open

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

Figure 45. Open Filter Examples



Before - No Read



After - Readable

Contrast Stretching

The **Contrast Stretching** filter maximizes image contrast.

Figure 46. Contrast Stretching Filter Examples



Before - No Read

After - Readable

Histogram Equalization

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

Figure 47. Histogram Equalization Filter Examples



Before - No Read

After - Readable

Smoothing

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

Figure 48. Smoothing Filter Examples



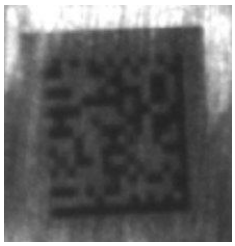
Before - No Read

After - Readable

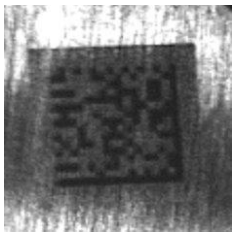
Sharpening

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

Figure 49. Sharpening Filter Examples



Before - No Read



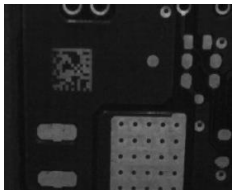
After - Readable

Deblurring

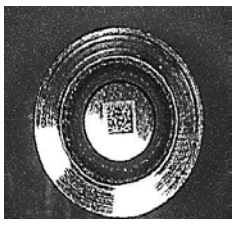
The **Deblurring** filter improves blurred images.

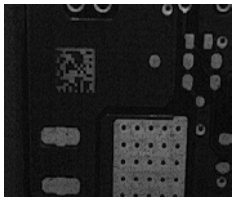
Figure 50. Deblurring Filter Examples





Before - No Read



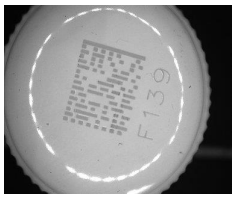


After - Readable

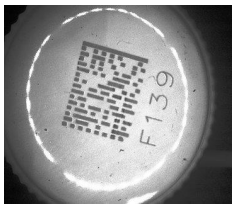
Black Enhancement

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

Figure 51. Black Enhancement Filter Examples



Before - No Read



After - Readable

White Enhancement

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

Figure 52. White Enhancement Filter Examples



Before - No Read

After - Readable

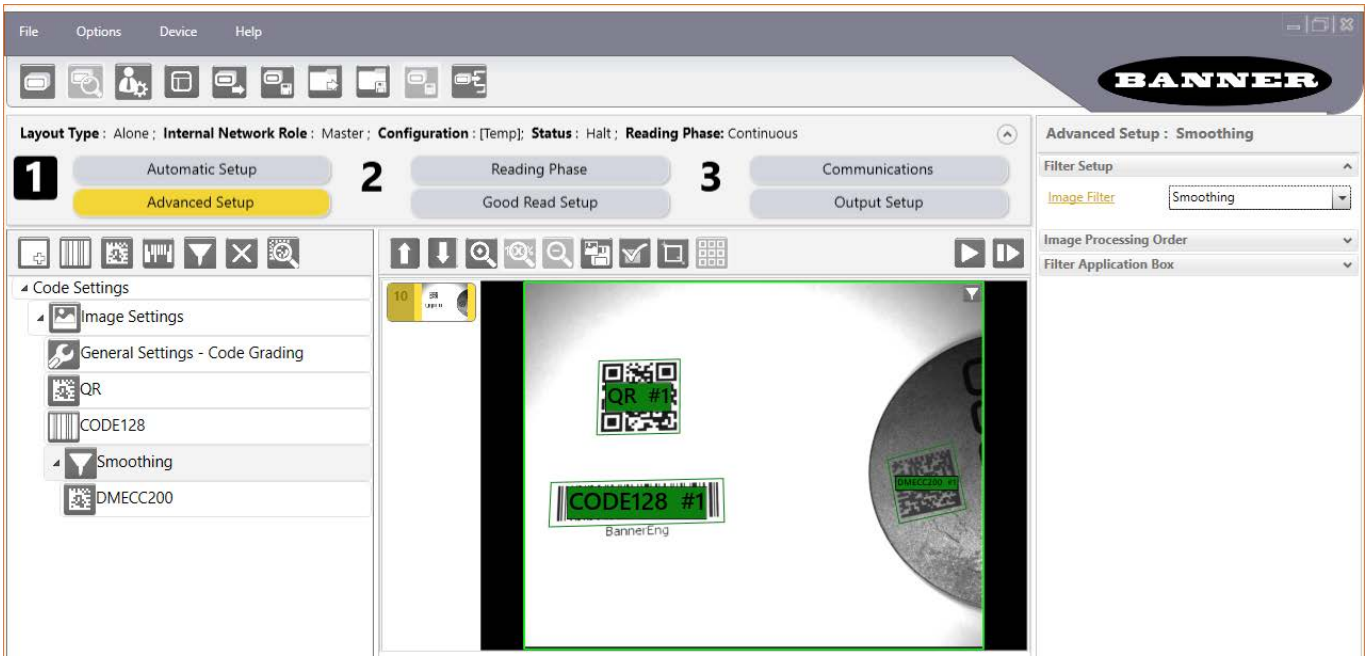
5.6.3 Set an Image Filter: Example 1

To demonstrate how to apply an Image Filter through Barcode Manager, 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.

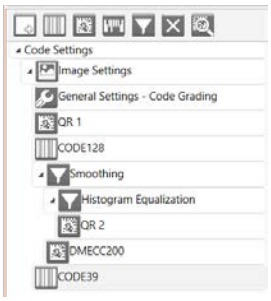
Because all codes can be found in any area of the image, both Code Localization and Filter Application boxes are left at their default values, all covering the entire image area. The resulting image is shown.

Figure 53. Code Localization and Filter Application at Default



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.**

Figure 54. Configuration Parameters



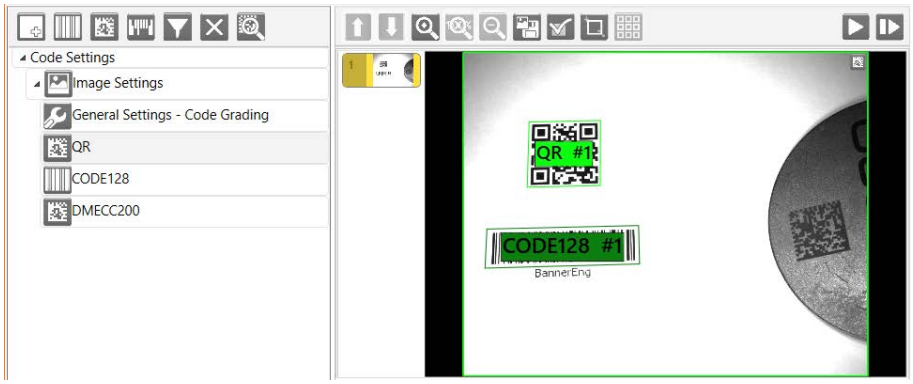
Because the visual effects in the image area are overlapped, you should not use this as a reference to determine how and where Image Filters are applied. You must read Image Filter application from the structure in the configuration parameter tree area. The example in the figure to the left shows:

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

Use the following steps to apply the Smoothing filter to Example 1.

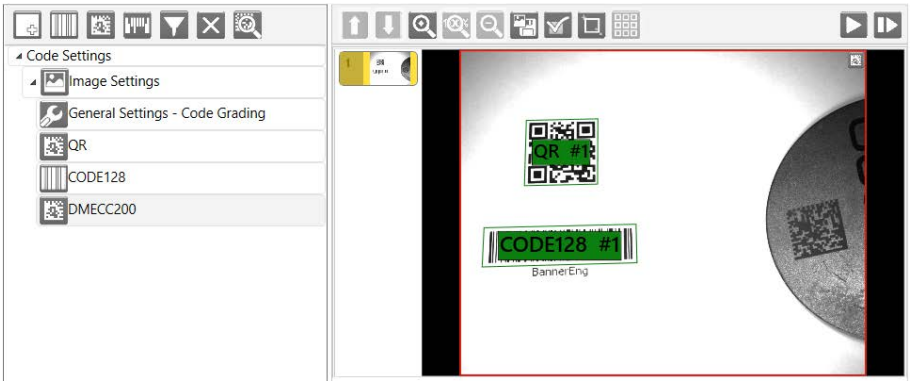
1. From the Image Settings group, click **Add 1D Code** or **Add 2D Code** to add the Code Symbologies that must be decoded: Code 128, DMECC200 and QR Code.
   The Code 128 and QR Codes are successfully decoded; Code Localization boxes, as well as code bounding boxes, are colored green.

Figure 55. Code 128 and QR Codes Successfully Decoded



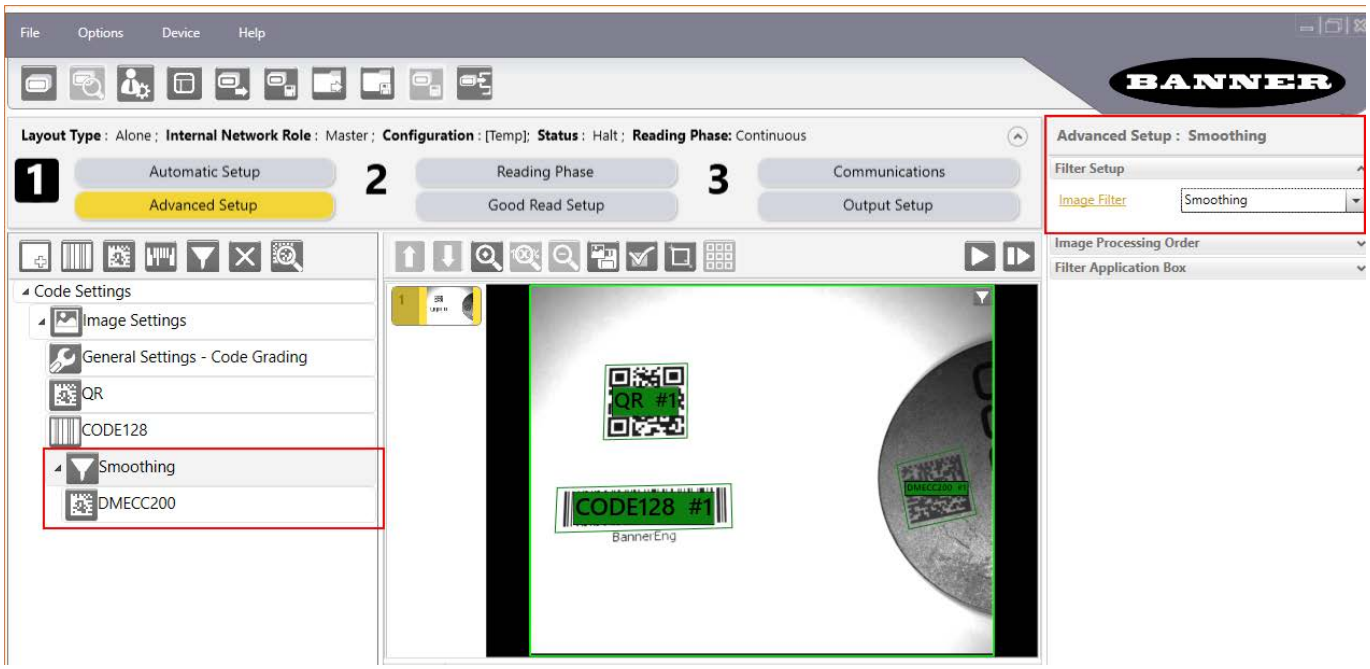
2. Click the Data Matrix ECC 200 code (**DMECC200**) in the parameter tree.
3. Click to add the image filter.
   Notice that the Data Matrix DPM code is not successfully decoded. The Code Localization box is colored red.

Figure 56. Data Matrix DPM Code Not Successfully Decoded



4. Select the desired Image Filter type for correct decoding.

Figure 57. Select Image Filter



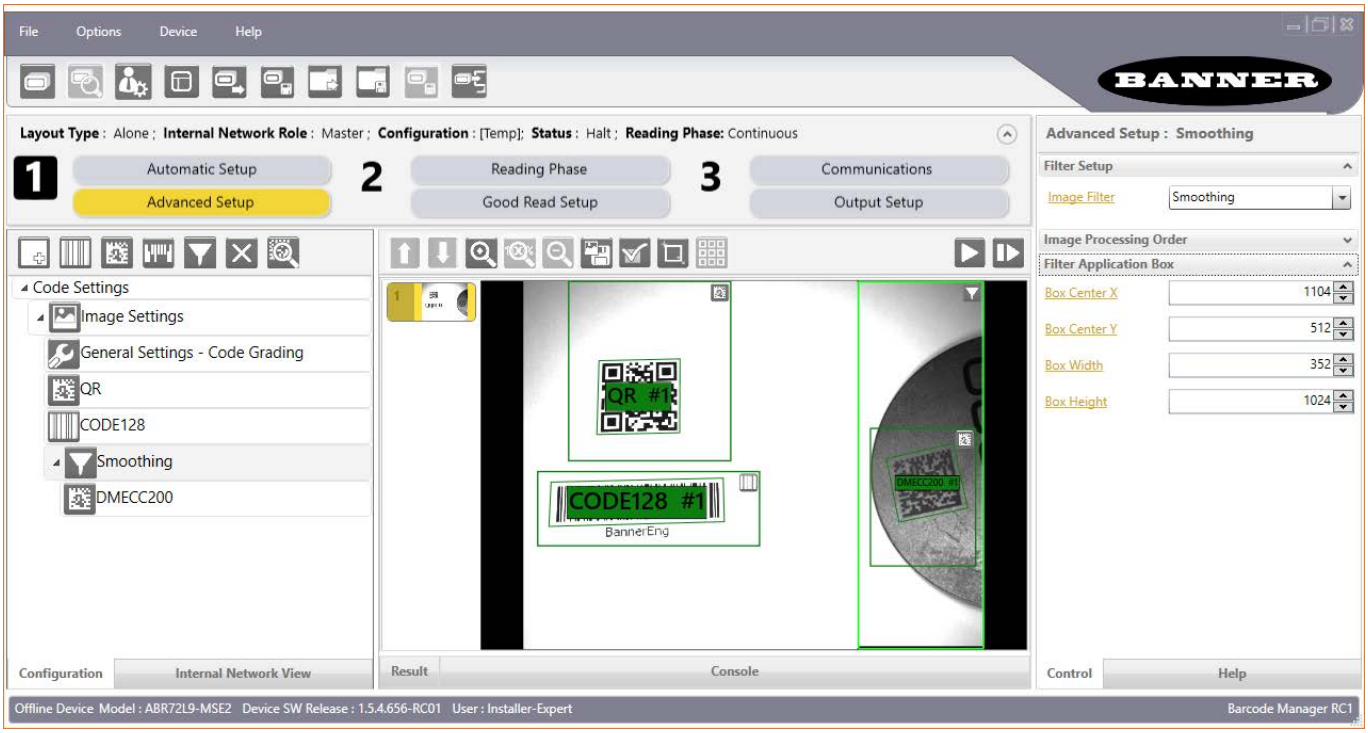
The Data Matrix DPM code is now successfully decoded. The Code Localization box, as well as the code bounding box, are colored green. Note that the icon in the upper right of the image changed to .

5.6.4 Set an Image Filter: Example 2

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

In this case, apply an Image Filter to the Data Matrix ECC 200 DPM code, and 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.

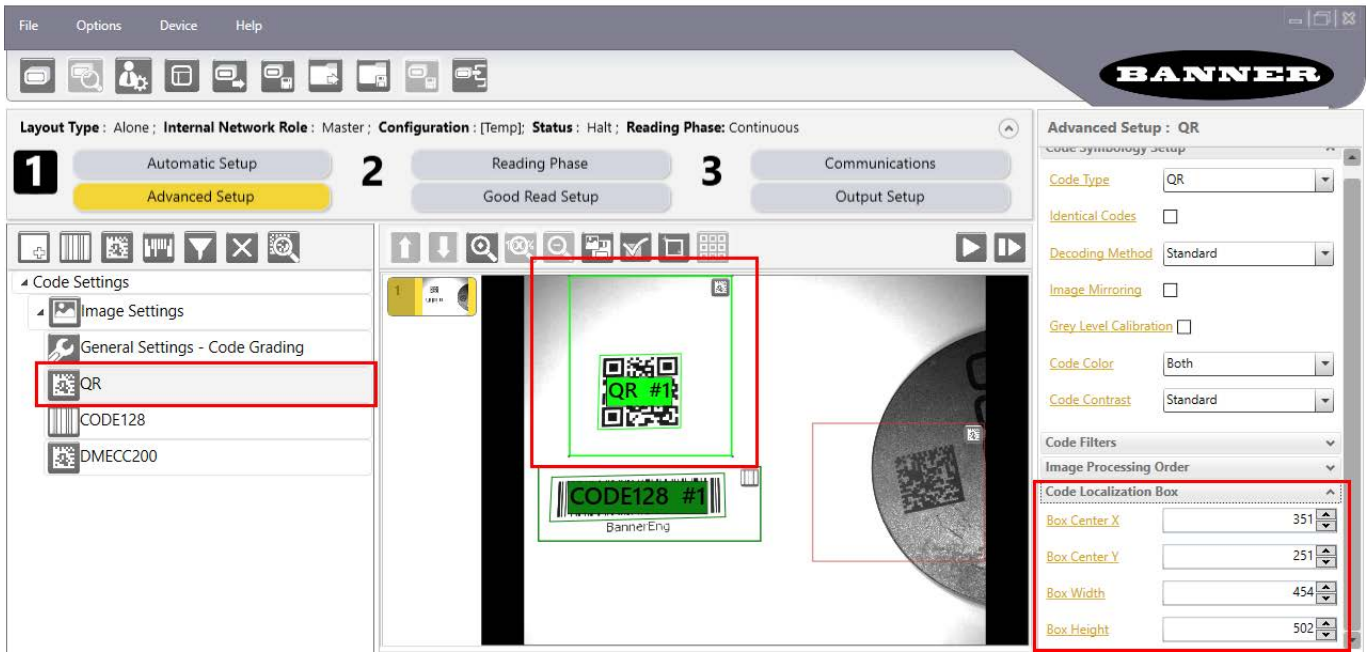
Figure 58. Find Code in a Specific Position



Use the following steps to apply the Smoothing filter to Example 2.

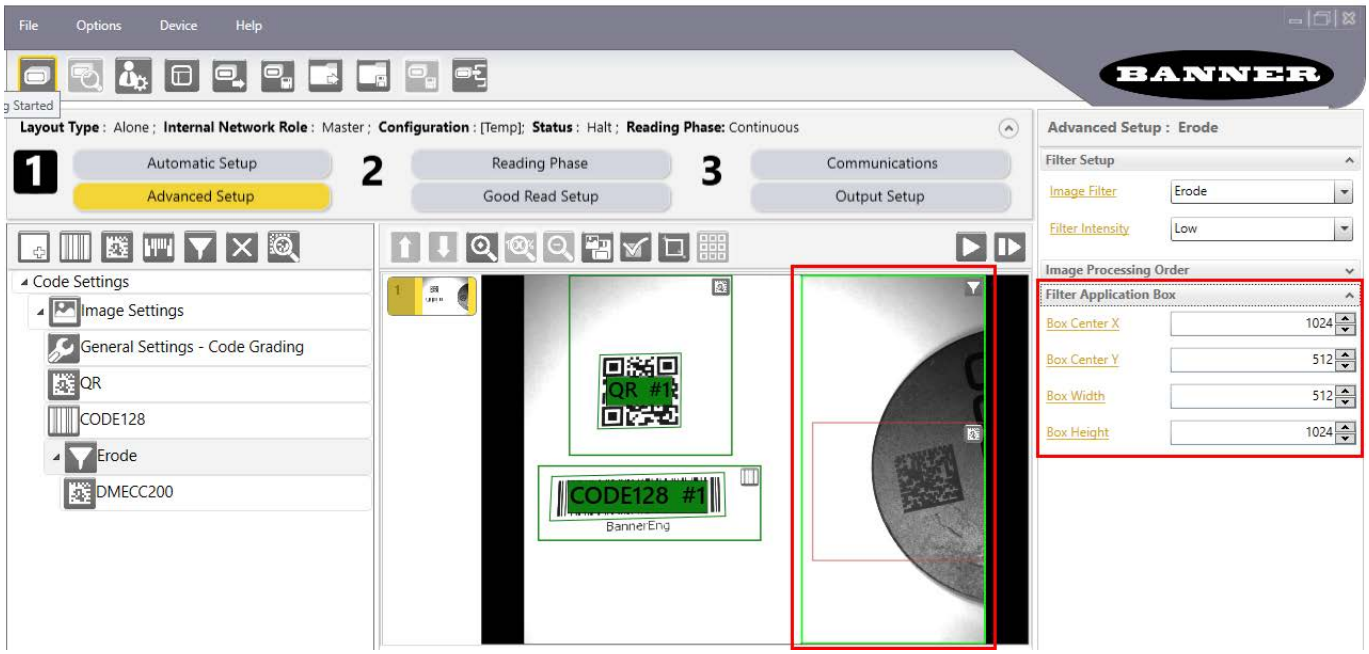
1. From the Image Settings group, click **Add 1D Code** or **Add 2D Code** to add the Code Symbologies that must be decoded: Code 128, DMECC200 and QR Code.
2. For each Code Symbology, resize and position its Code Localization Box to the image area where the code is found.

Figure 59. Resize and Position the Code Localization Box



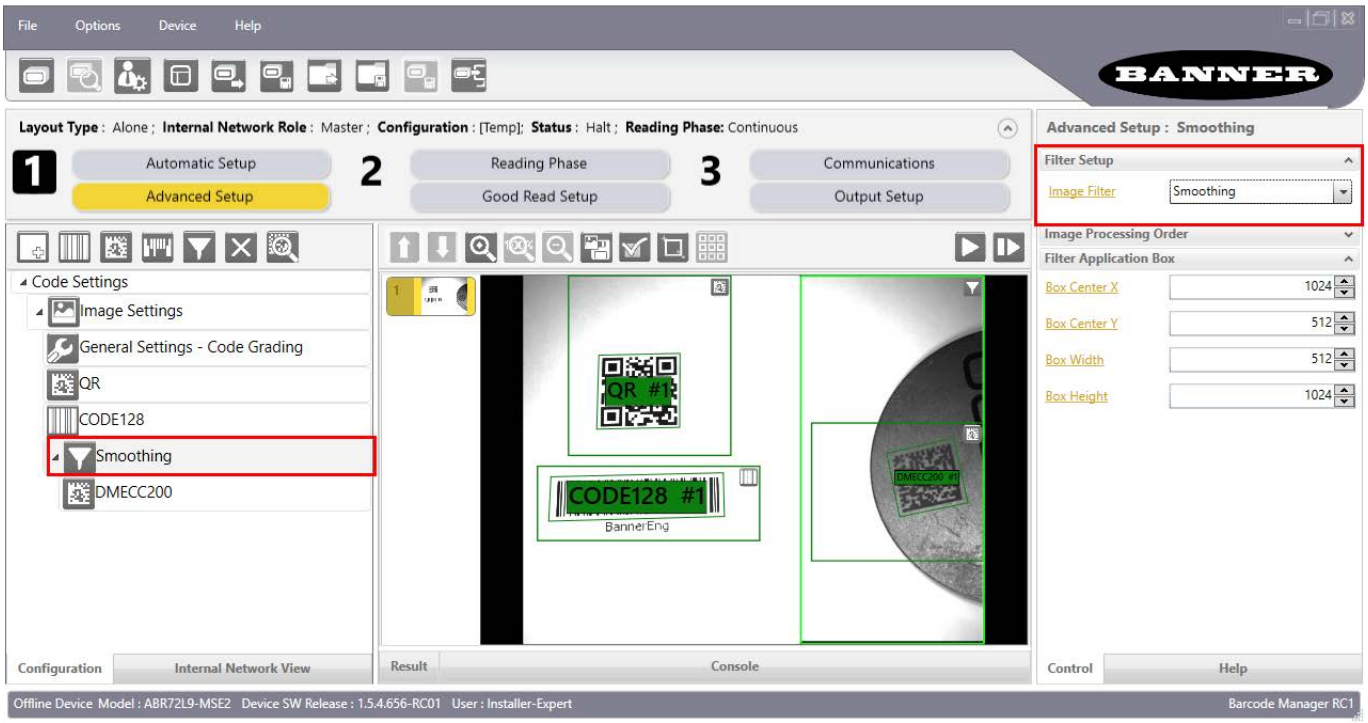
3. Click the Data Matrix ECC 200 code (**DMECC200**) in the parameter tree.
4. Click to add the image filter.
5. Resize and position the Filter Application Box around the image area where the filter will be applied.
   The relative Code Localization box must fit inside its Filter Application box.

Figure 60. Code Localization Box Inside the Filter Application Box



6. Select the desired Image Filter type and Filter Intensity for correct decoding.

Figure 61. Set Image Filter and Filter Intensity



The Data Matrix DPM code is now successfully decoded only within the expected position. The Code Localization box, as well as the code bounding box, are colored green.

5.7 ABR Input/Output Control

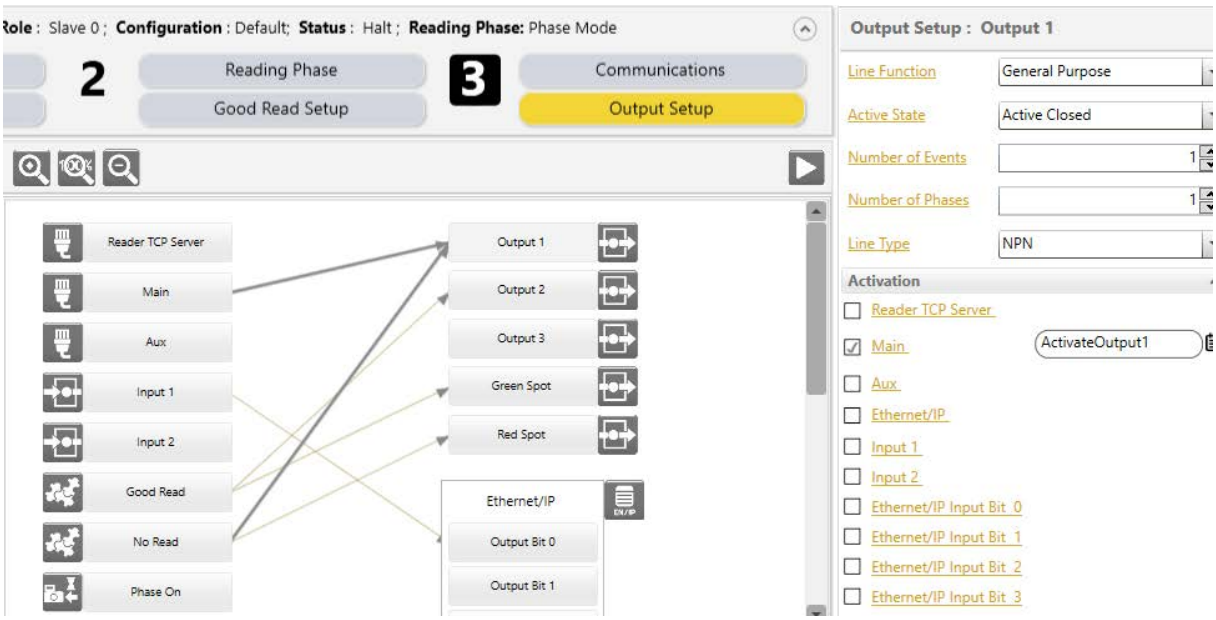
Input signals and control output signals of many types are compatible with the ABR. This is done by discrete I/O wires or by digital communications interfaces such as serial or Ethernet, including the Industrial Ethernet protocols EtherNet/IP™ [7], MODBUS® [8] TCP, SLMP, and PROFINET® [9] [10]. See the ABR reader manuals for more information on Industrial Ethernet configuration.

Configure control signals on the **Reading Phase**, **Communications**, and **Output Setup** pages. Each page lists available Input or Output signals in the **Configuration** panel. When a signal checkbox is selected, an arrow is automatically drawn in the middle panel to indicate the sources and destinations in each relationship.

Sometimes a new parameter becomes available after selecting the signal checkbox. In the following example, when the Main serial port was selected as an activation event for discrete Output 1, a text field displayed. If the exact text in this field is received over the Main serial port, Output1 is activated.

[7] EtherNet/IP™ is a trademark of ODVA, Inc.
[8] MODBUS® is a registered trademark of Schneider Electric USA, Inc.
[9] The ABR is PROFINET compatible but not PROFINET certified.
[10] PROFINET® is a registered trademark of PROFIBUS Nutzerorganisation e.V.

Figure 62. Input/Output Control



5.8 Backup and Restore through Barcode Manager

3000 Models

Barcode Manager allows Backup and Restore to be performed to the configuration PC via file.

Barcode Manager provides complete backup and restore functions (Configuration and Environmental parameters) for 3000 model readers.

Backup and Restore functions provide parameter storage including all configuration jobs present on the reader.

Important: It is strongly recommended to save all configurations to backup files.

When performing Backup and Restore to/from file, device firmware can be included.

7000 Models

Barcode Manager allows Backup and Restore to be performed to the configuration PC via file or to an external storage device such as TCNM-ACMK-100.

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 Barcode Manager for all reading devices when connected to: TCNM-ACBB1 + TCNM-ACMK-100.

Note: Before executing a Backup to a TCNM-ACMK-100 backup module make sure the Write Protection switch is set to Unlocked.

5.8.1 Perform a Backup

Use the following steps to perform a backup.

1. Go to **Device** > **Backup/Restore**.
2. Select either **Single Reader Backup** (to file on PC; 3000 and 7000 models) or **Backup to external storage device** (7000 models).

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

The message "Temporary configuration will not be included in the backup. Do you want to proceed?" displays. A configuration in temporary memory will not be saved. Save the configuration to the reader before performing Backup.

3. If you are performing a backup to a file, select whether or not to include the firmware.
4. Browse to the desired location and click **Save**.
   The backup begins.

   At the end of the backup, Barcode Manager shows a message indicating successful completion.
5. Click **OK**.

5.8.2 Perform a Restore

Use the following steps to perform a restore.

1. Go to **Device** > **Backup/Restore**.
2. Select either **Single Reader Restore** (from file on PC; 3000 or 7000 models) or **Restore from external storage device** (7000 models).

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

   If restoring ID-NET, this may take a few minutes.

   At the end of the restore, Barcode Manager shows a message indicating successful completion.
3. Click **OK** to restart the device and apply the changes.

5.8.3 Replace a Device

Important: 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. Make sure the new device has been previously set to factory default.
3. Connect the new device.
4. Run the Restore procedure by selecting either **Single Reader Restore** (from file on PC; 3000 or 7000 models) or **Restore from external storage device** item (7000 models). See Perform a Restore on p. 44.

Note: In case of Backup or Restore operation failures, error messages are displayed in the Monitor Diagnostic page.

5.9 Reset to Defaults

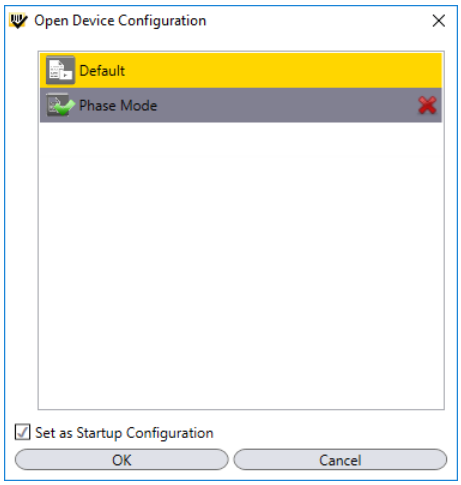
The device parameters are divided into two main classes, Configuration and Environment, which are affected differently by the Restore Defaults commands.

- 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.

5.9.1 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 Device Configuration** list.

Figure 63. Open Device Configuration



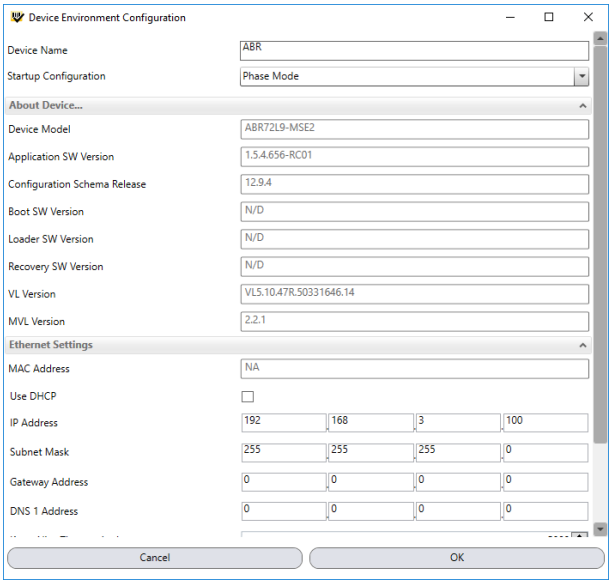
The same action can be performed from the **Device menu** > **Backup/Restore** > **Restore Defaults** > **Restore Default Startup Configuration**. The default configuration is set to run at startup and the reader resets.

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

5.9.2 Restore Default Environment

Restore Default Environment returns all Environment parameters to their factory default settings. For Ethernet models , the default IP address is restored, as well as all the parameters managed in the **Device Environment Configuration** window.

*Figure 64. **Device Environment Configuration** Window*



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

5.9.3 Restore Factory Defaults

To return a device to its absolute FFctory default parameters (for example device replacement), use the **Restore Factory Defaults** command. You will be prompted to confirm.

All environment parameters are restored to Factory default values and **any existing configurations stored on the device are erased**. The device resets and starts in run mode with the factory default configuration.

5.10 Software Reset

At any time the device can be reset by the **Restart Device** command in the Barcode Manager **Device** menu.

Important: Save the current configuration before restarting.

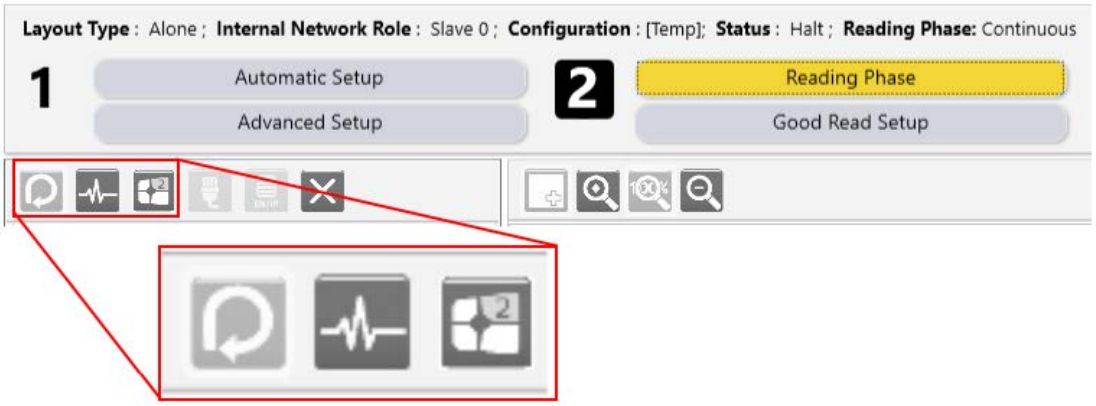
6 Reading Phase (Trigger) Modes

6.1 Introduction

The ABR Barcode Reader has three Reading Phase modes to take images. Sometimes multiple images are taken and their results combined in an attempt to read and process barcode data. The Reading Phases are:

Continuous Mode	No trigger; free-running, self-triggering
One Shot Mode	One single image taken for every external trigger signal received
Phase Mode	Multiple images taken for every external trigger signal received

Figure 65. Continuous Mode, One Shot Mode, and Phase Mode, respectively



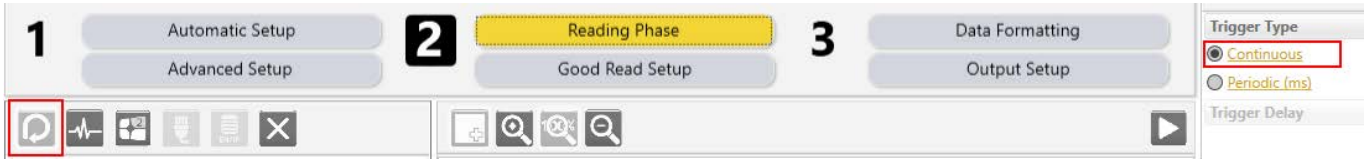
Every Reading Phase goes through the following three steps for each image and typically transmits data when the Phase ends:

Step:	Description	Factors that affect the duration
Image Exposure	The image sensor gathers light to create the image	The exact time is set in the ABR configuration to optimize the brightness of the image
Image Acquisition	The image sensor transfers the image to memory where it can be processed	ABR model and number of pixel rows cropped out
Image Processing	The ABR processor runs reading algorithms to find barcodes, decode them, and extract the desired data for output	Image contents, ABR model, ABR reading and timeout configuration

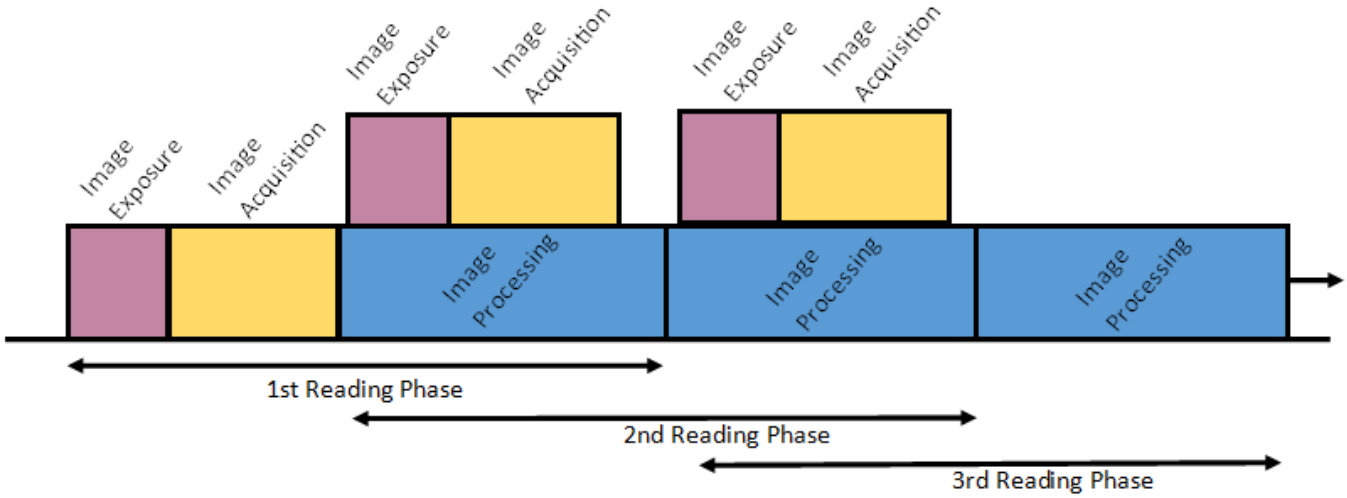
The options for Reading Phase control are configured on the **Reading Phase** screen of Barcode Manager.

6.2 Continuous Mode

6.2.1 Acquisition Trigger: Trigger Type = Continuous



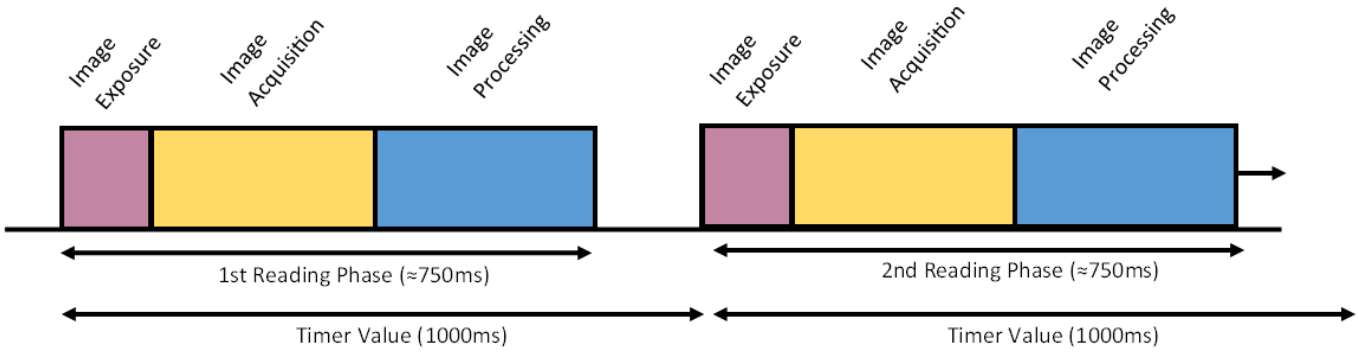
The ABR acquires images and processes them as fast as it is capable by overlapping acquisitions and processing, putting images into an image buffer until they can be processed. This means that the ABR usually acquires images at about the rate of the image processing duration, because this often takes longer than the acquisition.



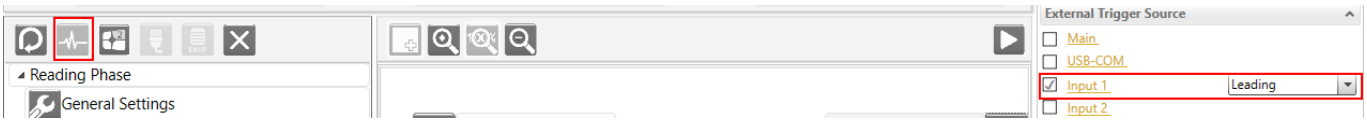
6.2.2 Acquisition Trigger: Trigger Type = Periodic (ms)



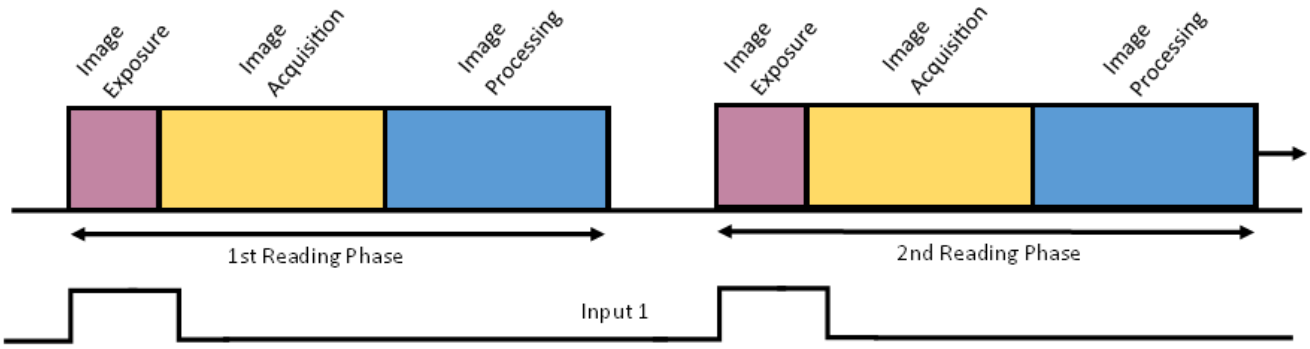
The ABR acquires images at specific, timed intervals. If the processing takes too long and extends past the end of the periodic timer, and the image buffer is full, the ABR skips taking an image.



6.3 One Shot Mode



In One Shot Mode, the ABR is triggered by an external signal such as an Input 1 leading edge signal. The ABR takes one image and the Reading Phase ends as soon as the image is processed. One Shot Mode does not delay the output transmissions; data is sent when the processing finishes. Output delays are only possible in Phase Mode.

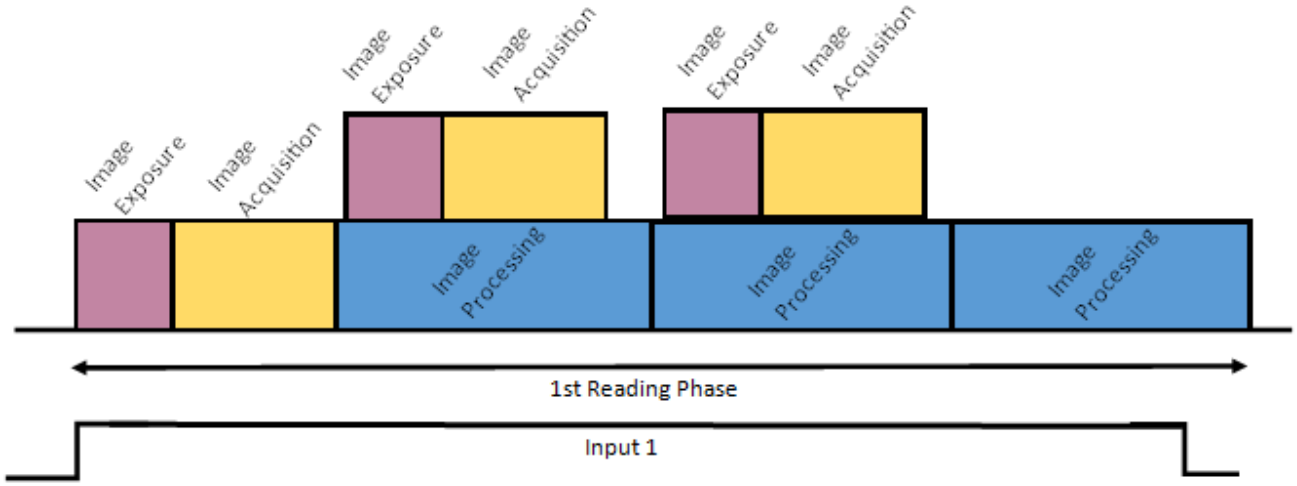


6.4 Phase Mode

6.4.1 Acquisition Trigger: Trigger Type = Continuous



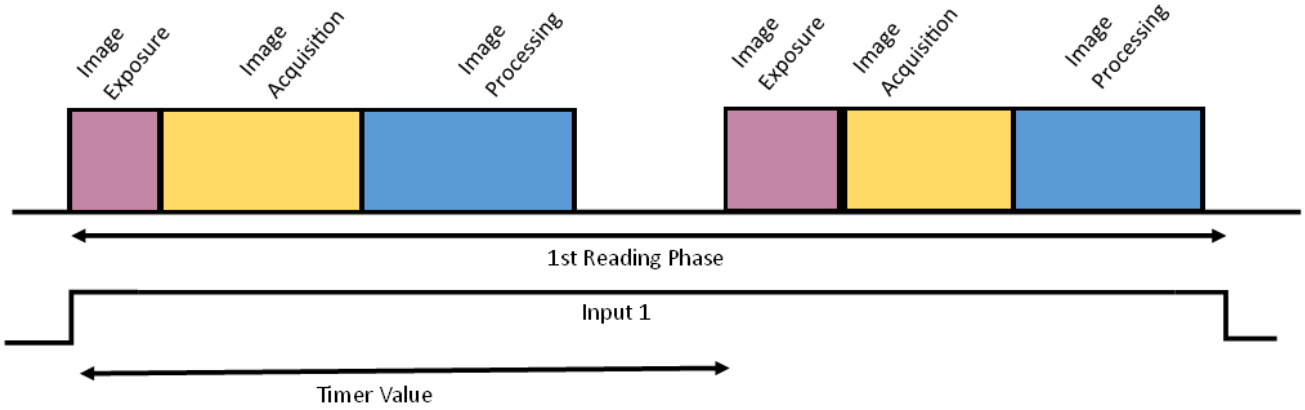
The first image acquisition is initiated by an external trigger signal, defined as the Phase On event. The ABR begins taking a new image immediately after the last image has finished acquisition and continues taking images at the fastest possible rate, based on how long the image processing takes. This continues either until the Phase ends from an external signal defined as the Phase Off event, or when a successful good read is achieved. Data from all images in a Reading Phase are (typically) combined into one data output transmission at the end of the Phase.



6.4.2 Acquisition Trigger: Trigger Type = Periodic (ms)



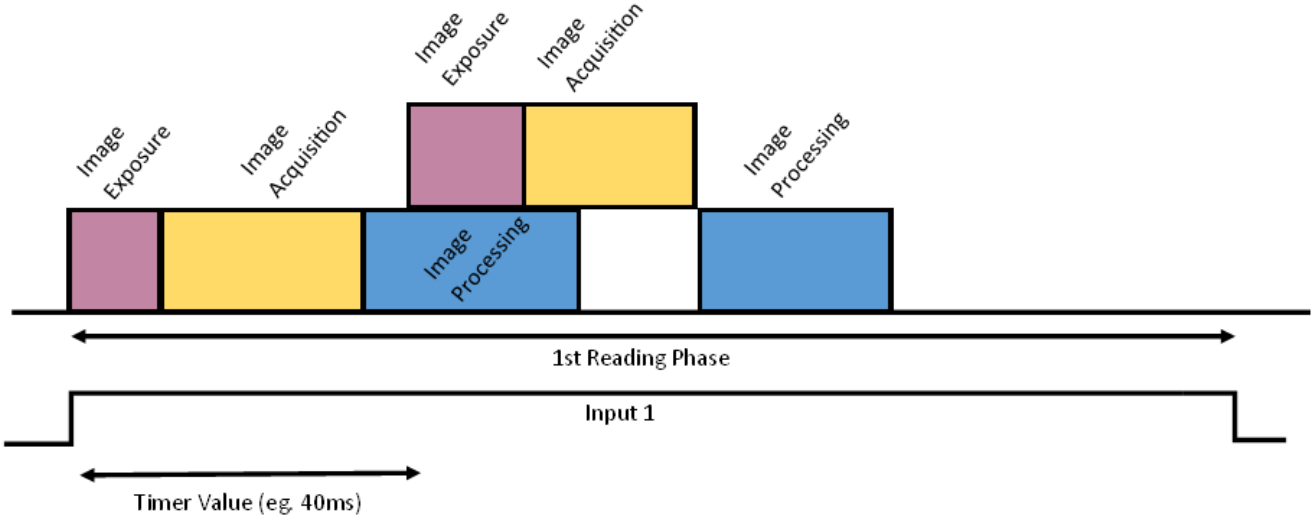
Same as Phase Mode (Trigger Type: Continuous), but the ABR continuously acquires images at specific timed intervals during a Reading Phase. If the processing takes too long and extends past the end of the periodic timer, and the image buffer is full, the ABR skips taking an image.



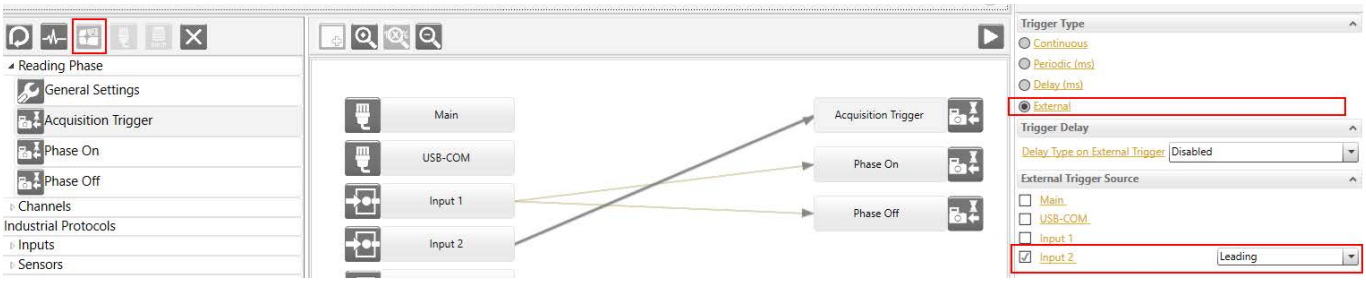
6.4.3 Acquisition Trigger: Trigger Type = Delay (ms)



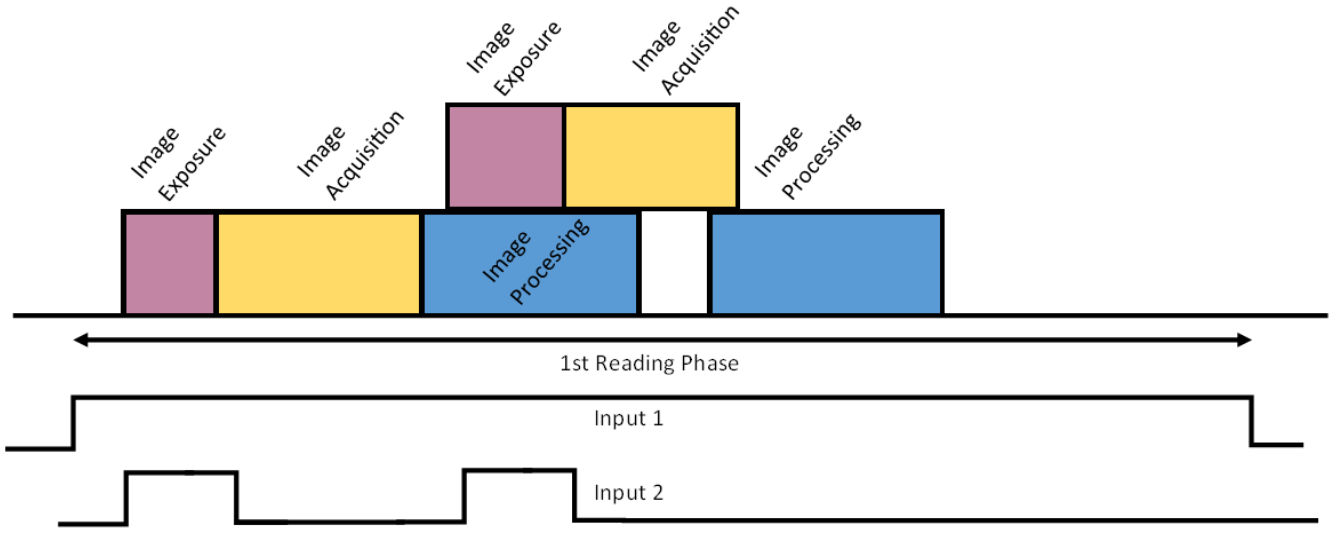
Same as Phase Mode (Trigger Type: Continuous) except the ABR acquires images at specific timed moments after the Reading Phase begins. When all the defined moments are past, no more images are taken, but the Reading Phase can still be active and processing the images. The ABR has an adjustable-size image buffer for buffering acquired images. This means that it can acquire new images into memory while previous images are still decoding, as in this example with an image taken 0 ms after Phase On and 40 ms after Phase On.



6.4.4 Acquisition Trigger: Trigger Type = External



Same as Phase Mode (Trigger Type: Delay) except that the ABR acquires images only when driven by a separate external signal. In this case, Input 1 starts and ends the Phase, and Input 2 controls when to take images.



7 Data Collection Methods

7.1 Code Collection

Valid Operating Modes:

- Continuous
- One Shot
- Phase Mode

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 on the **Communications** page.

Example

Figure 66. Example Good Read Setup: Code Collection



As shown in the figure, 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	Output
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	Output
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.

7.2 Code Combination

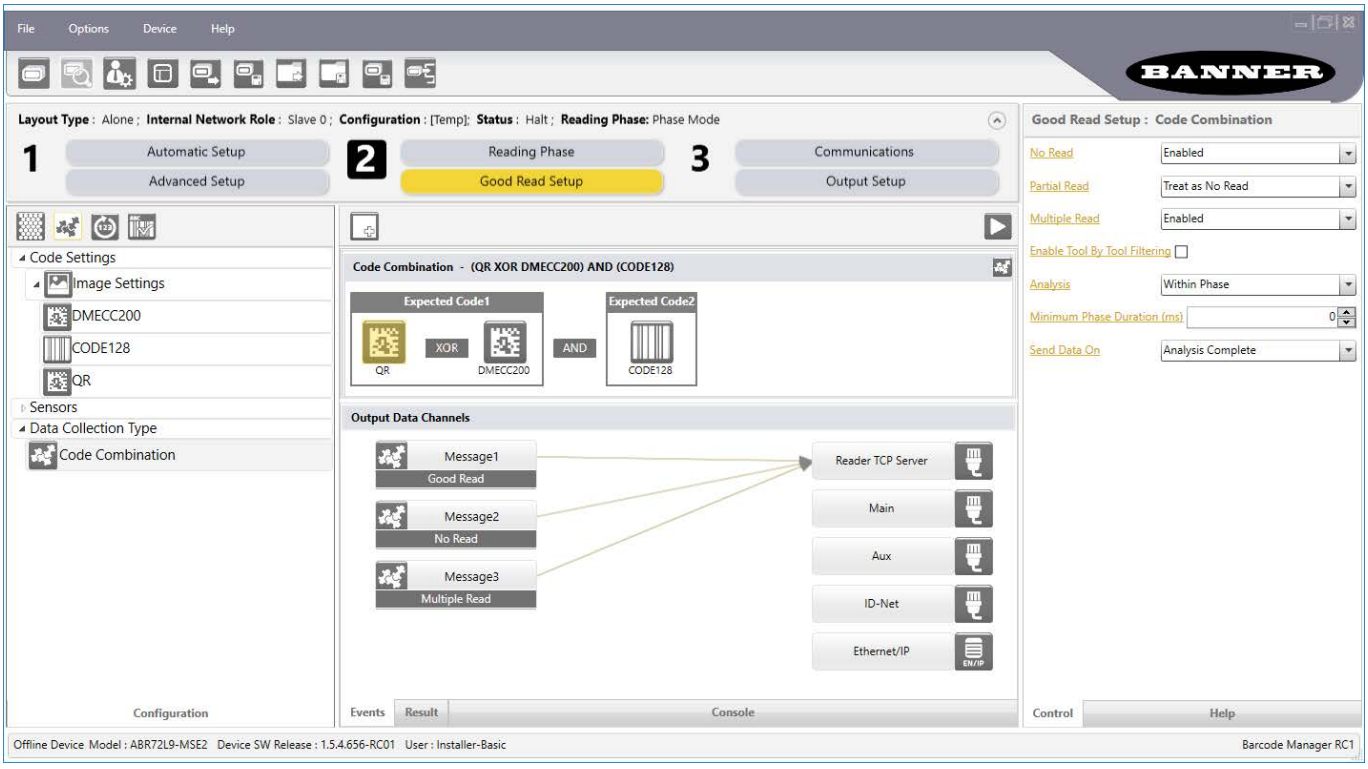
Valid Operating Modes:

- Continuous
- One Shot
- Phase Mode

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

Figure 67. Example Good Read Setup: Code Combination



As shown in the figure , 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	Output
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.

7.3 Code Presentation

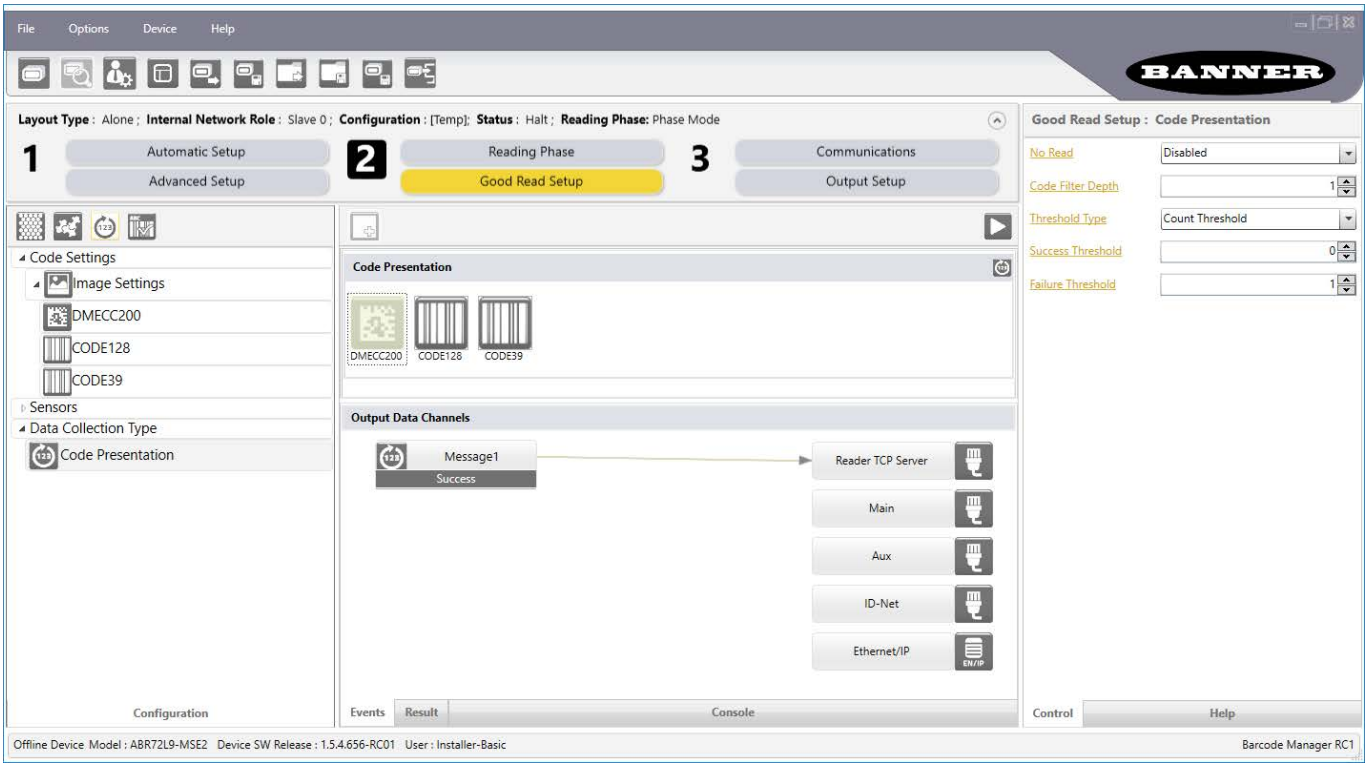
Valid Operating Modes:

- 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

Figure 68. Example Good Read Setup: Code Presentation



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
- Multiple reads are handled by the **Threshold Type Count Threshold**: Success Threshold=0, Failure Threshold=1

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

7.4 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.

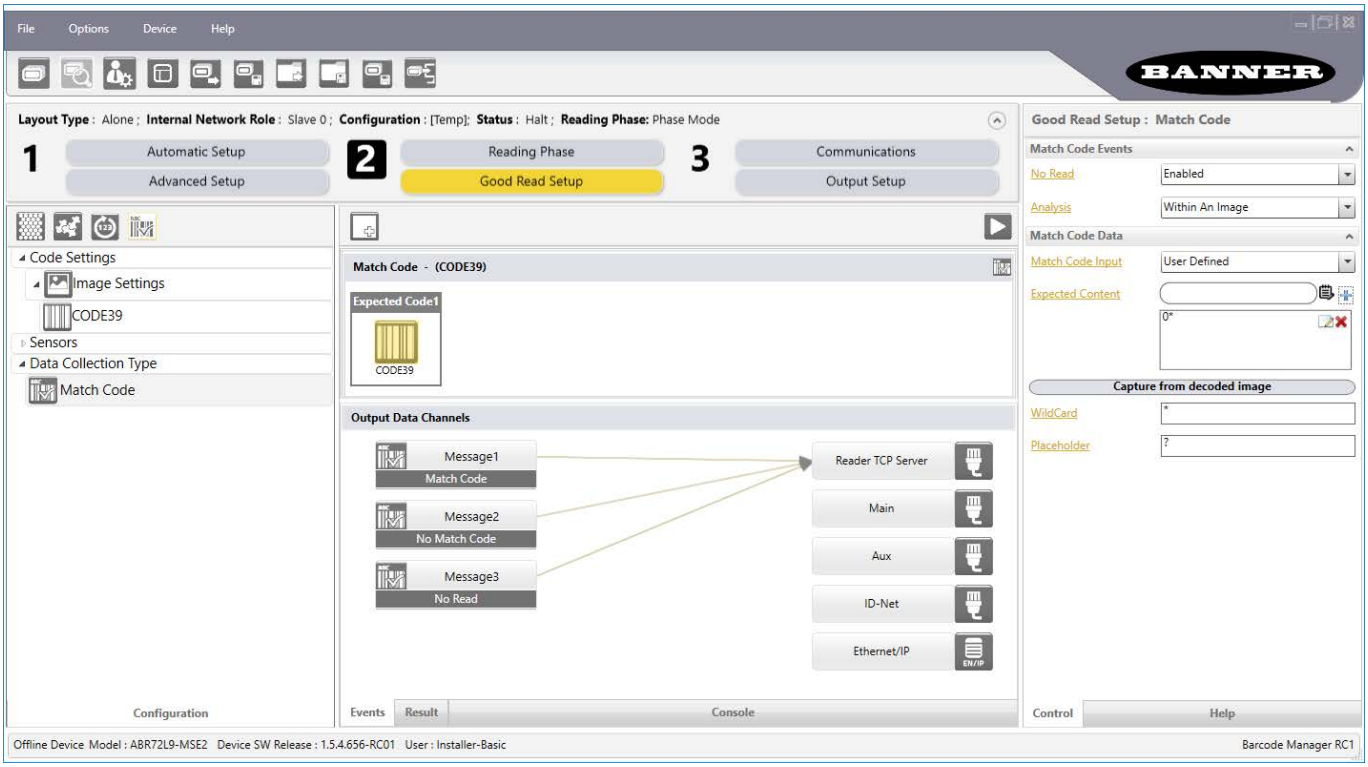
7.4.1 User Defined

Valid Operating Modes:

- Continuous
- One Shot
- Phase Mode

Example

Figure 69. Example Good Read Setup: Match Code - User Defined



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	Output
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.

7.4.2 Input Based

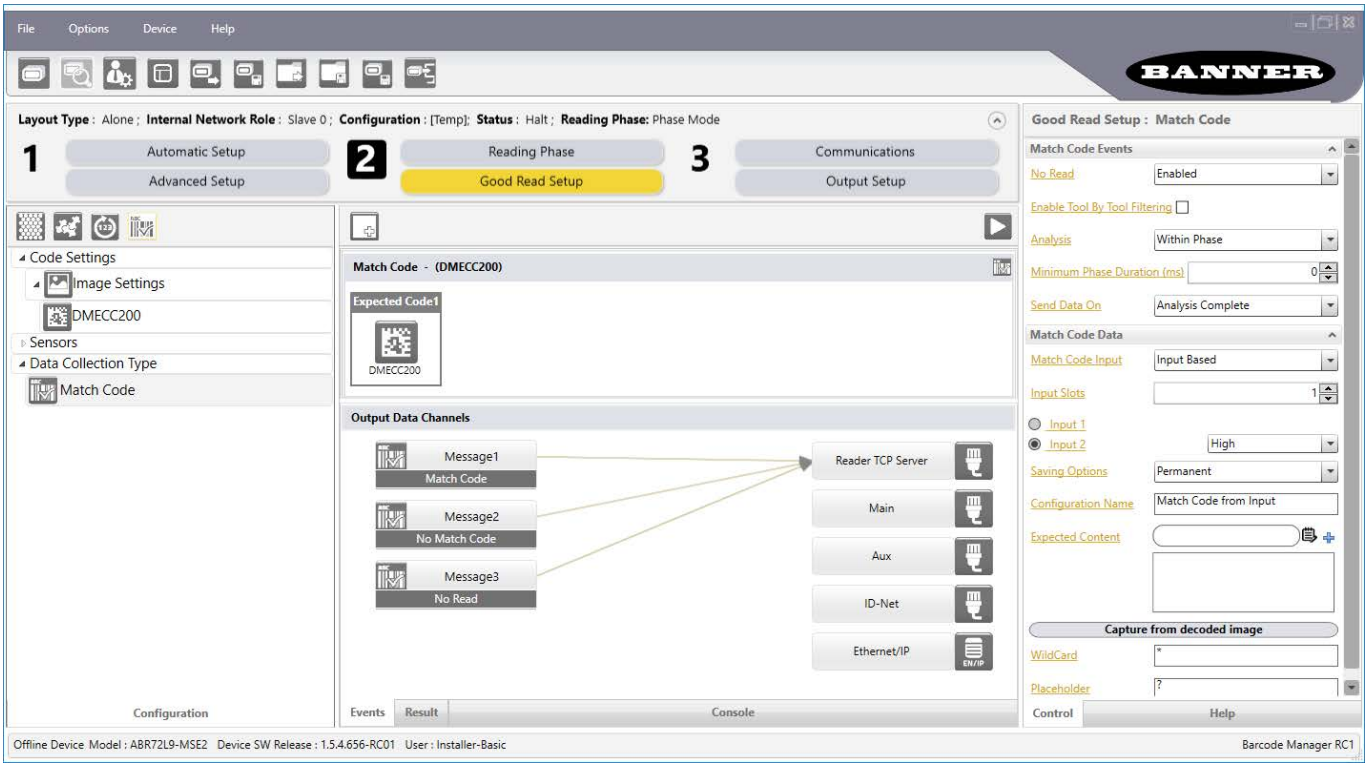
Valid Operating Modes:

- 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

Figure 70. Example Good Read Setup: Match Code - Input Based



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

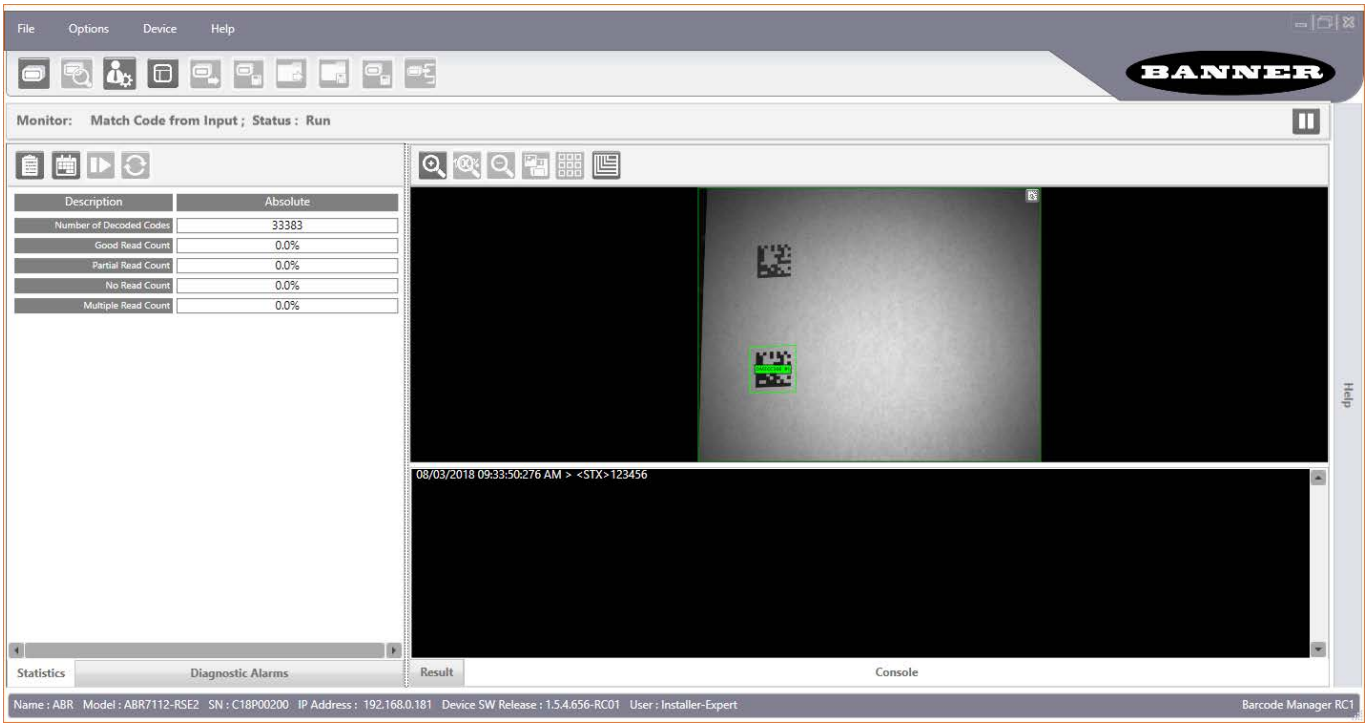
- Data Matrix ECC 200 type is enabled and present in the Match Code
- **No Read** is **Enabled**
- **Match Code Input** is **Input Based** (captured by the Input 2 High state), Saved to Permanent memory with configuration job name = “Match Code from Input”

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

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

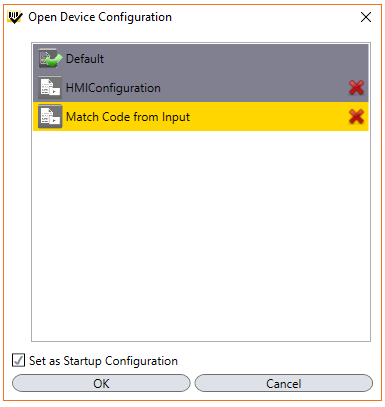
The code is read, saved in the Expected Content list, and the entire job will be saved to the configuration name “Match Code from Input”.

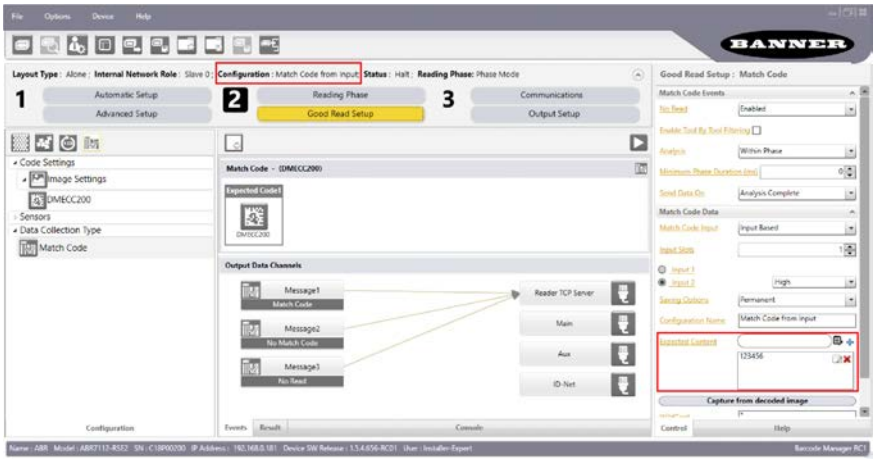
Figure 71. Monitor Match Code from Input



Return to the Configuration Environment.

Figure 72. Match Code from Input





If you choose the Temporary Saving Option, save the job manually from the configuration environment.

Case	Output
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.

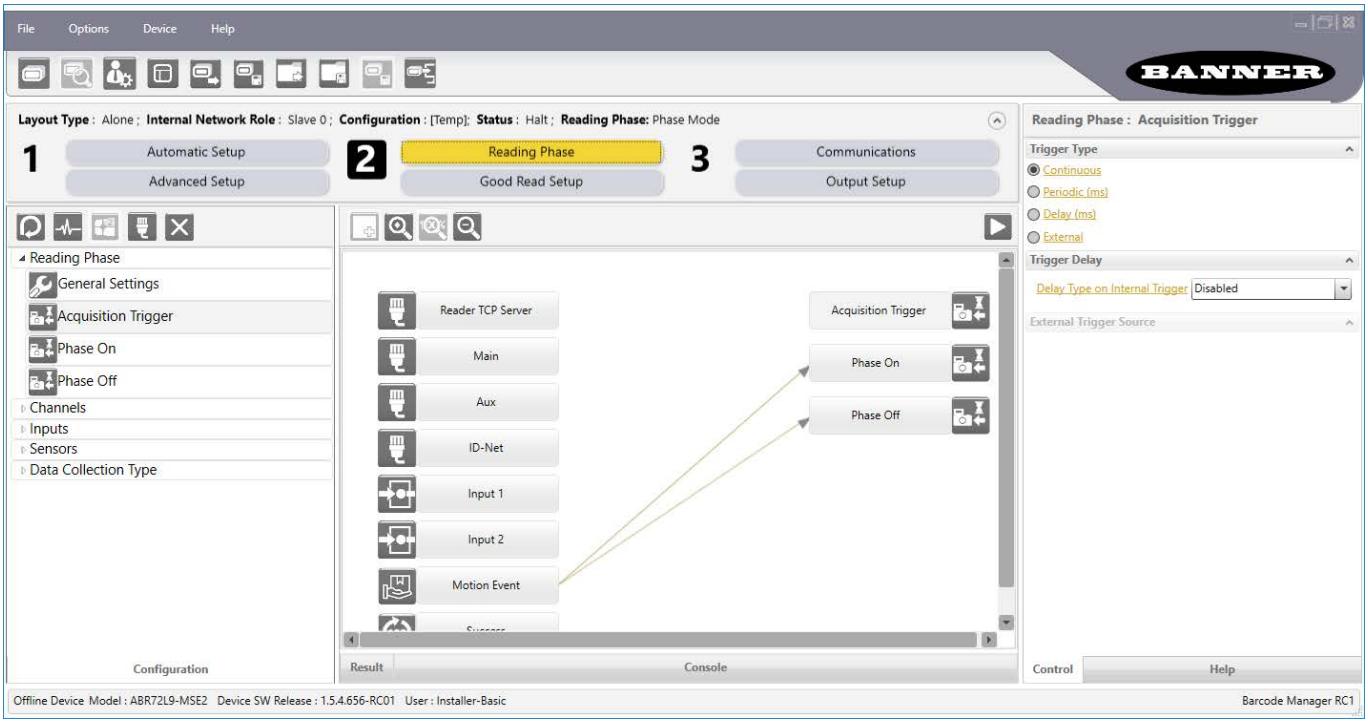
7.5 Presentation Mode

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 are presented to the device. Select **Presentation Mode** from the **Task Selection** area.

Presentation mode includes the following default parameters:

- 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 seconds
- Data Matrix symbology enabled
- Output Message to the Host is sent to the TCP Channel
- Green Spot is illuminated on successful read

Figure 73. Presentation Mode



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.

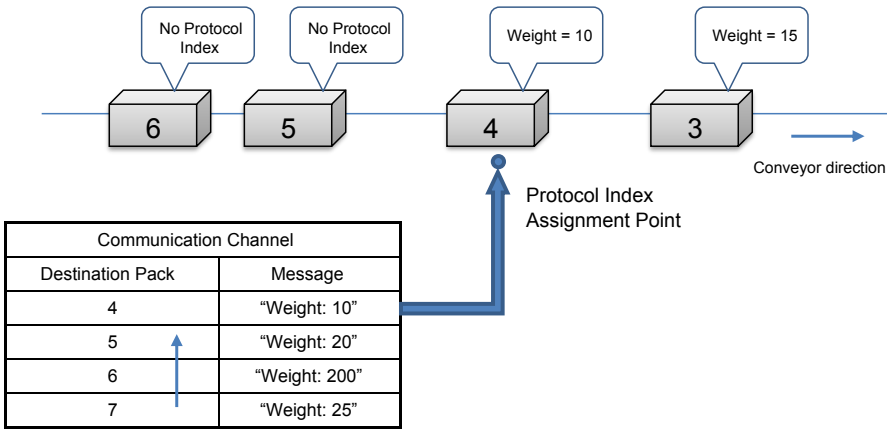
Presentation Mode can also be used with the Code Collection Data Collection method.

8 Protocol Index

8.1 Overview

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

Figure 74. Protocol Index Sample Application



Protocol Index can be used in Phase Mode during the active reading phase.

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.

Figure 75. Protocol Index 1



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.

Parameters

Header

Sets the Header character or string for the Protocol Index message.

Each defined protocol index (one per source channel) can have a different Header. Either the Header or the Terminator can be blank, but not both. Typically if the Protocol Index Message Length Type is Variable Length, then both the Header and Terminator will be needed. If the Message Length Type is Fixed Length, then at least the Header or the Terminator is necessary.

Terminator

Sets the Terminator character or string for the Protocol Index message.

Each defined protocol index (one per source channel) can have a different Terminator. Either the Header or the Terminator can be blank, but not both. 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.

Message Length Type

The Protocol Index message can be either variable or fixed length.

Variable Length: Whatever is received between the Header and Terminator is considered a valid Protocol Index message.

Fixed Length: Whatever is received between the Header and Terminator is considered a valid Protocol Index message only if the received data equals the specified Message Length.

Message Length

Defines the number of characters (from 1 to 255) in the fixed length Protocol Index message.

Three cases are defined:

- Only Header: Only the number of characters received after the Header will be accepted as a valid Protocol Index message. The parser reinitializes and waits for a new Header.
- Only Terminator: Only the number of characters received before the Terminator will be accepted as a valid Protocol Index message.
- Both Header and Terminator: Any message having more or less characters between the Header and Terminator will not be accepted as a valid Protocol Index message.

Source Channel

Selects which communication channel will be used for the specific Protocol Index message. Only one Protocol Index message can be assigned to each available communication channel.

8.2 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.

Figure 76. Code Protocol Index Field



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

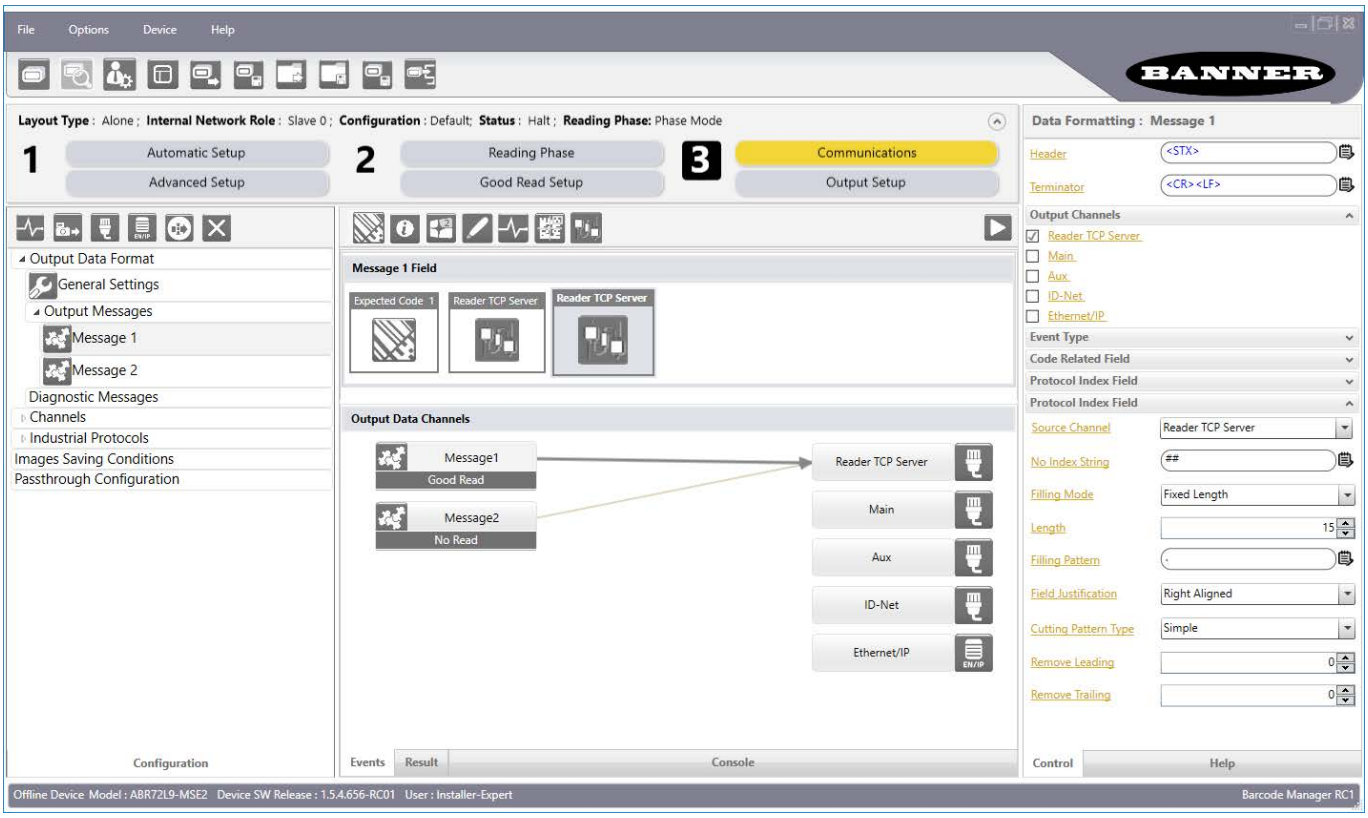
8.3 Parsing a Complex Protocol Index Message

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 Output Data message.

Figure 77. Code Protocol Index Field 1



Figure 78. Code Protocol Index Field 2



9 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 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 p. 69.

This section outlines the main characteristics of the Output Message and describes the standard Output Message tools.

Information relative to code reading is transmitted in standard formats on the device's selected interface. This is referred to as the OUTPUT MESSAGE.

The general format of the OUTPUT MESSAGE is: `<HEADER><RESULTS><TERMINATOR>`

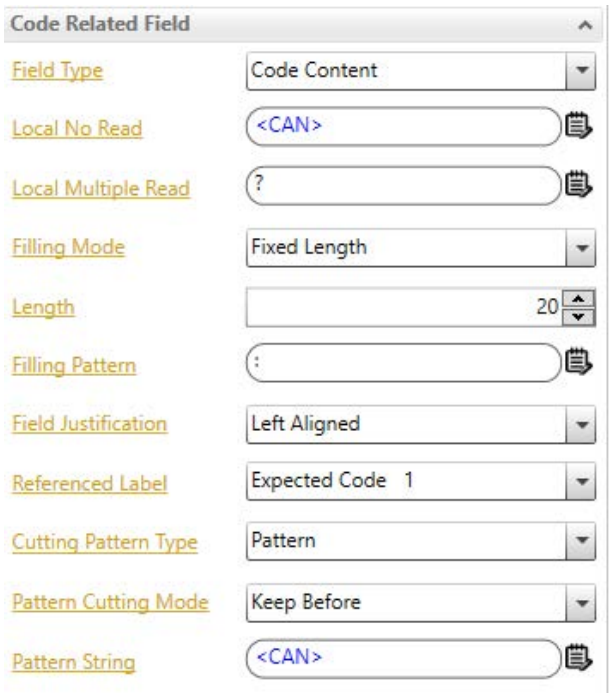
The RESULTS component is composed of several optional fields which are used in different combinations to create different output messages (such as Successful Read, No Read, Statistical Data, Diagnostic Data, etc.). These fields can be inserted into the output message in any order. A list of the formatting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

Fields	Icon	Meaning
Codes Related		Report various code related information types in the output message: Code Content, Number of Characters, Code Symbology, 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.
Global Reading		Offers different types of information depending on the data collection Analysis Mode and on the Operating Mode. When analyzing 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.
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.

9.1 Field Length Management

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

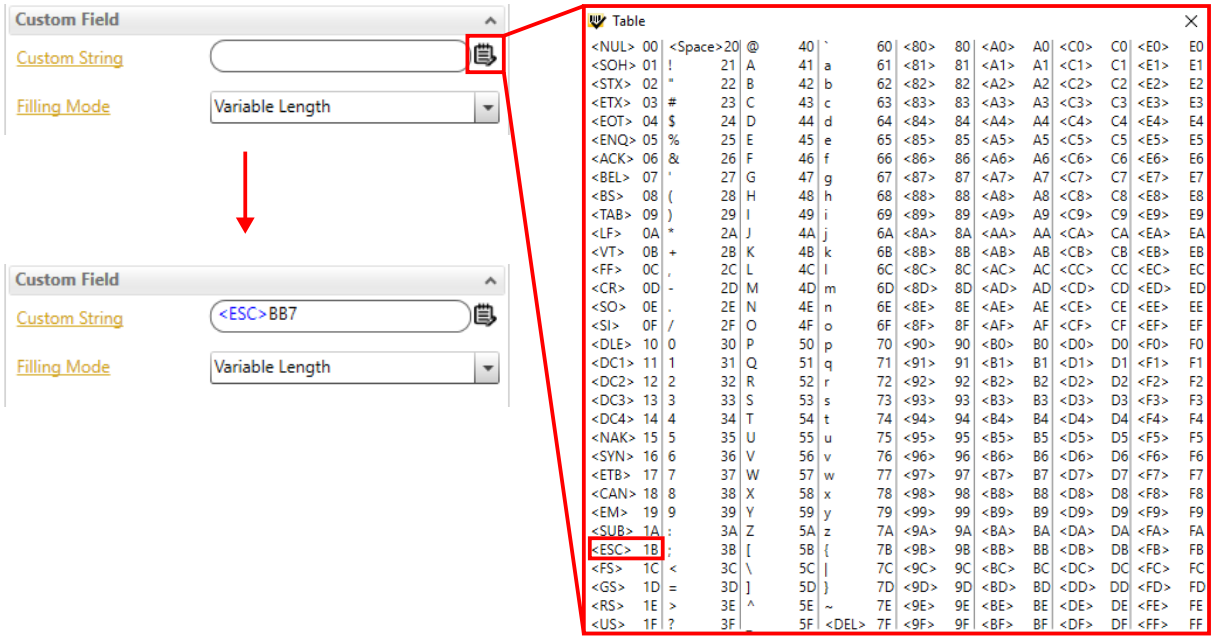
Figure 79. Code Related Field



9.2 String Entry

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

Figure 80. Input ASCII Characters



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

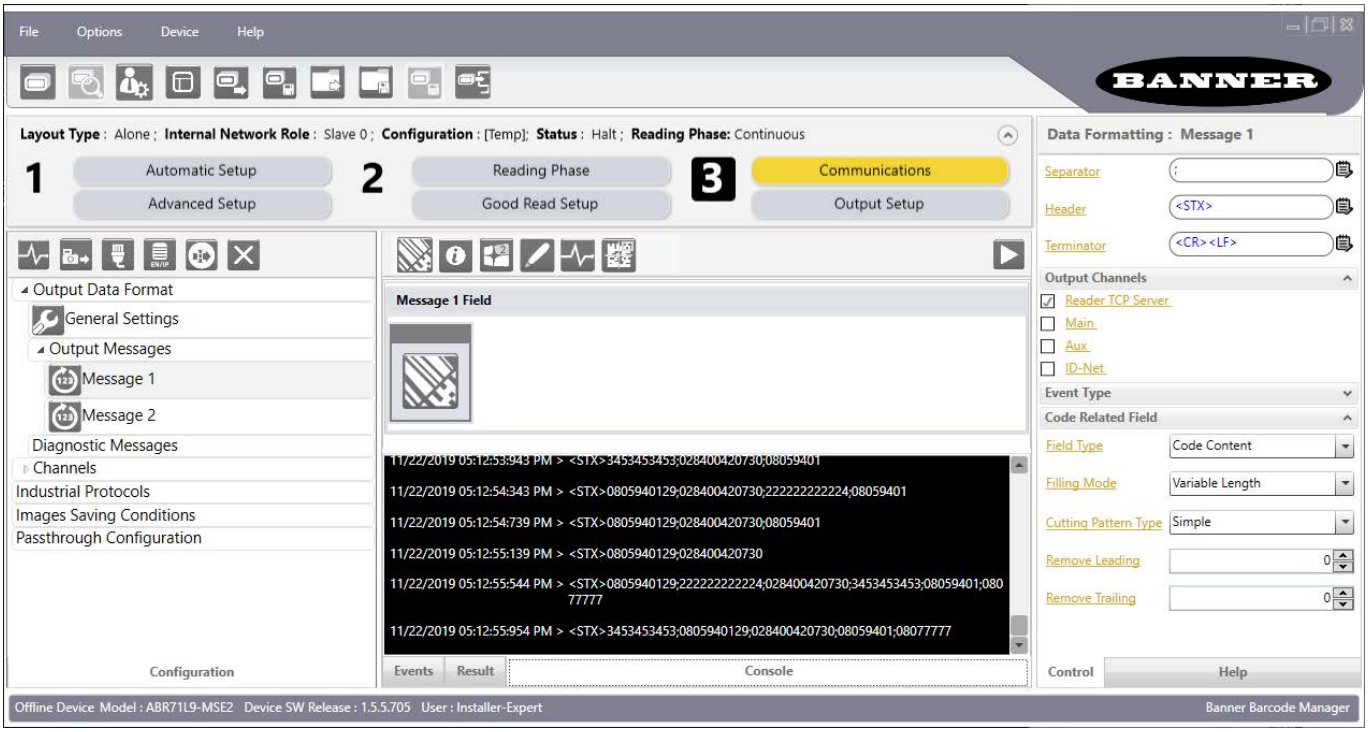
It is faster to type normal characters directly from your keyboard, however, non-printable characters must be entered using the table.

9.3 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]`

Example

Figure 81. Separator

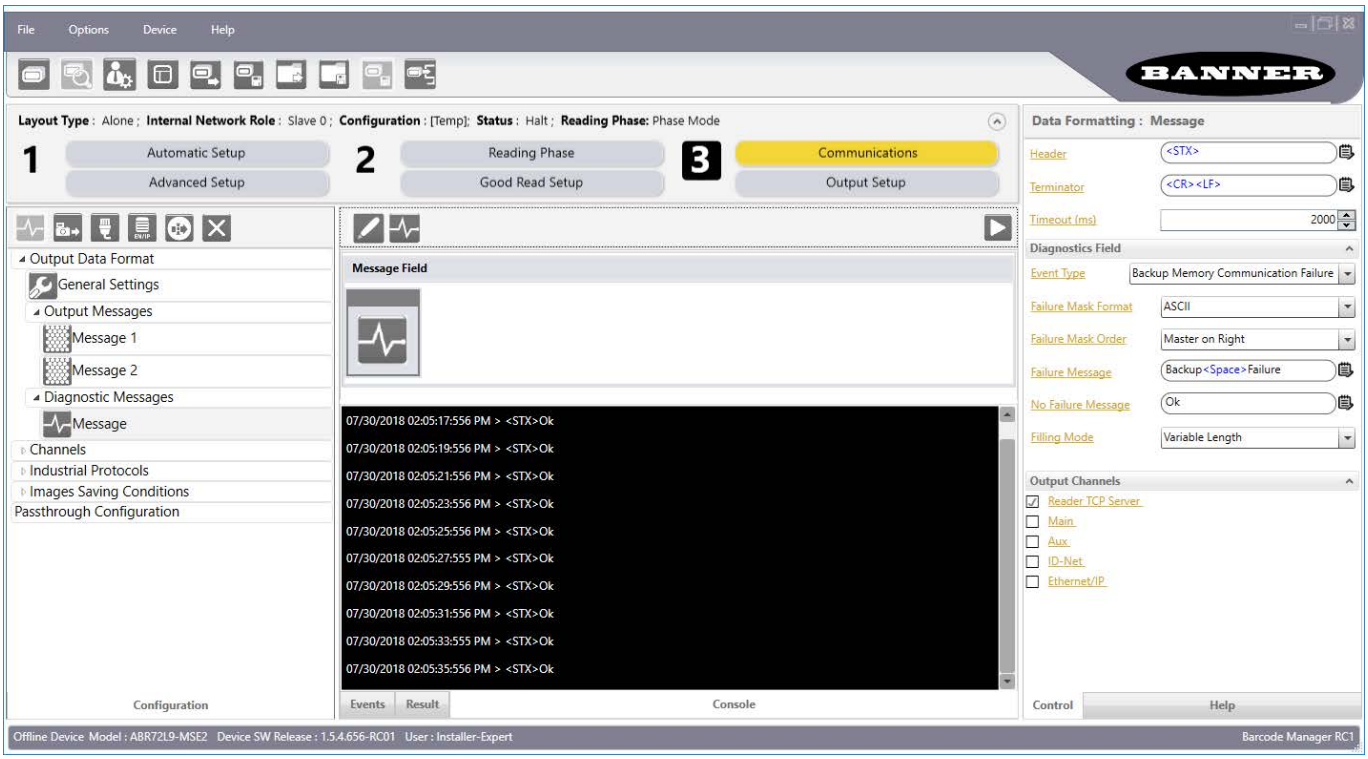


9.4 Independent Diagnostic Messages

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.

Figure 82. Backup Memory Communication Failure



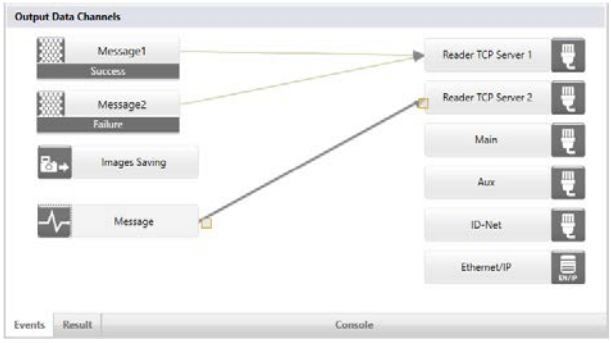
Typically only the Failure string is defined to avoid unnecessary message traffic.

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 Reader TCP Server connection, as shown.

Figure 83. Send Monitoring to a Different Reader TCP Server Connection



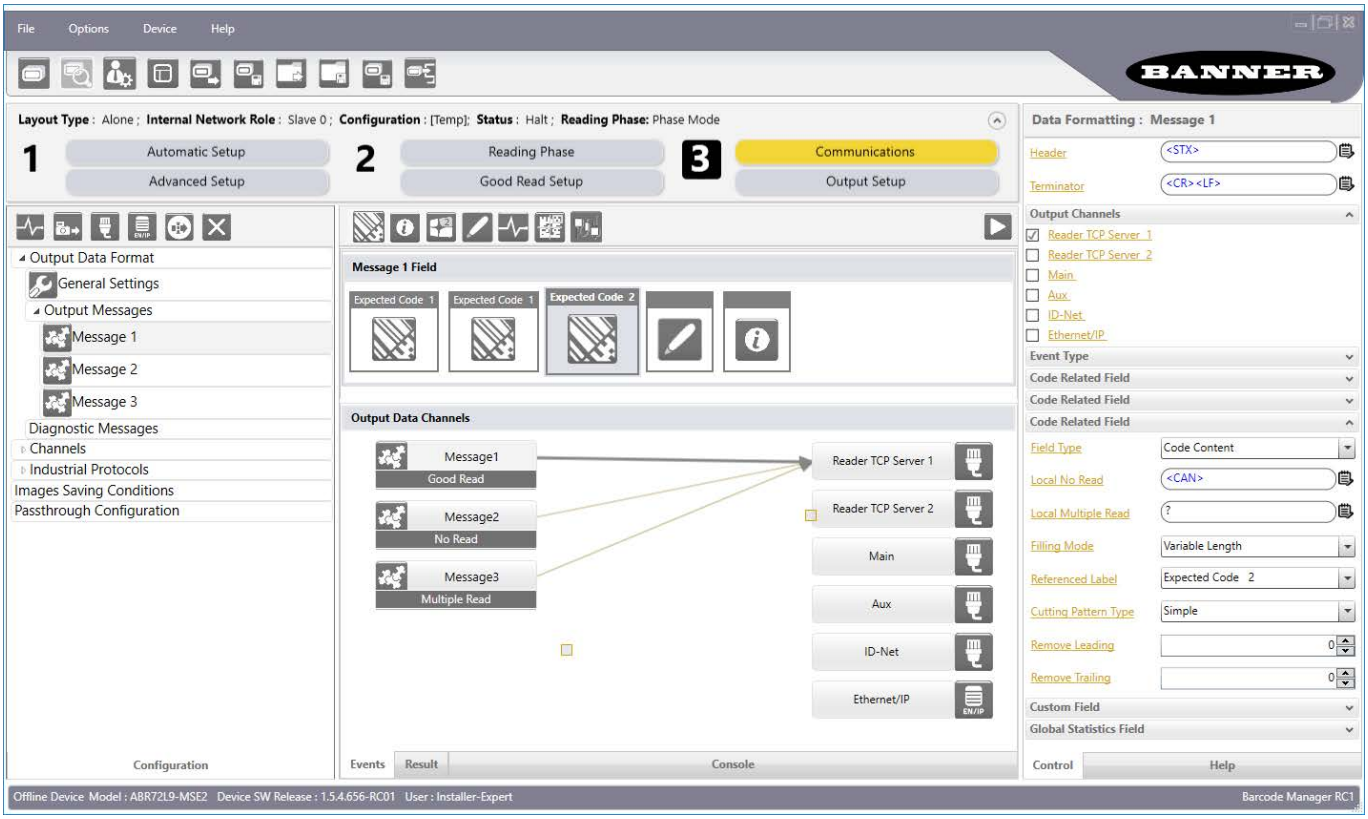
Several Diagnostic fields can be added to the Diagnostic Message for complete monitoring.

9.5 Code Combination Message Formatting Example

In Code Combination the expected result is always known in terms of code reading. To simplify these examples, Code Cutting or Justification will not be considered. All fields are considered with the default value as Variable Length fields.

Recalling the example in Code Combination on p. 53.

Figure 84. Code Combination Message Formatting Example



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.

Drag them into different positions to change the ordering of the message fields.

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

Then modify the formatting parameters of each field in the Control panel.

The 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
```

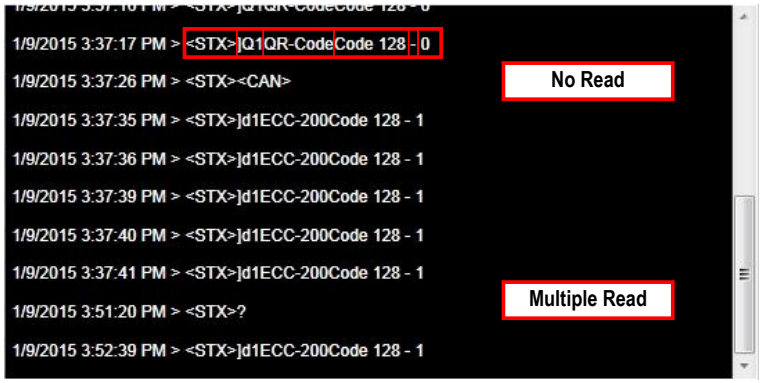
```
Message 2 = No Read
Custom Field
Custom String = <CAN>
```

```
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.

Figure 85. Code Combination Message Formatting Example



9.6 Script Formatter

Script formatting allows 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 **Communications** > **General Settings** > **Enable Script Formatter** checkbox. The **Script Formatter** option is available to **Installer-Expert** users.

The Script Formatter replaces the Output Messages branch. It is complete with several editing tool icons to facilitate scripting within the Barcode Manager interface.

Figure 86. Script Formatter



After enabling the Script Formatter, all output messages must be generated by the script. The scripts are written in JavaScript language, conforming to ECMAScript 5.0/5.1 Language Specification. This chapter does not cover all the commands possible in JavaScript, so there are several other scenarios that are possible beyond what is listed here.

Scripts are allowed to read the following data:

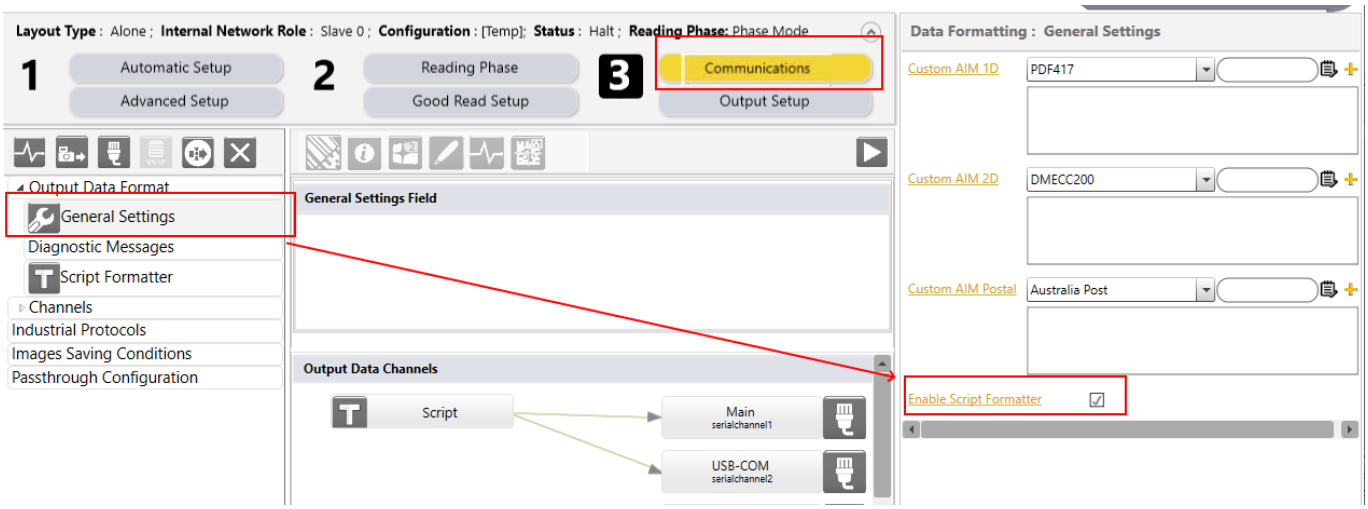
- Decoding data
- Code Quality Metrics (grading)
- Device statistics

- Device diagnostics
- Data sent to the script by Host Mode Programming commands

To enable the Script Formatter, go to **Communications** > **General Settings** and select the **Enable Script Formatter** checkbox.

Note: The Script Formatter option is available to Installer-Expert users.

Figure 87. Enable Script Formatter



Click **Script Formatter** to access the window. Any of the scripts in this document can be copied and pasted into the script formatter.

9.6.1 Scripting Basics

Simple Script Examples

The default script looks like this:

```
function onResult(result, output) {
    // write your formatting function here
}
```

The code between the `{}` braces of the `onResult` function runs every time an ABR is ready to send data. The *result* object is all the input data from the reading attempt. The *output* object is the data sent back to the output channels when the function is finished executing. Use `//` slashes to denote a comment (no code is executed).

The following script shows how to send out data from the script:

```
function onResult(result, output) {
    // write your formatting function here
    output.setMessage('Hello World!\n');
}
```

In this case the data "Hello World!" is followed by the ASCII control character Line Feed (LF). ASCII characters can be written as their Hexadecimal value or their escape sequence. This means that replacing the `\n` with a `\x0A` would make no difference because `0A` is the Hex value for Line Feed and `\n` is the escape sequence programming shorthand for Line Feed. Either way, the data appears in the console as:

```
Hello World!<LF>
```

Creating Variables

Variables are pieces of memory that are named and allocated in a script to store and use data.

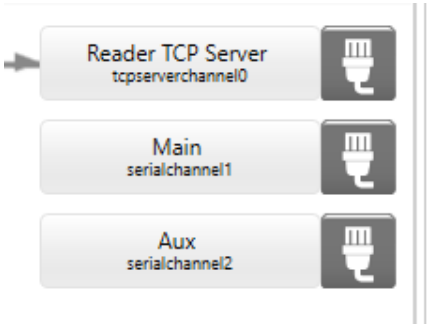
The following example shows how a variable is created. In this example, a message is sent only once. The counter variable increases past zero and nothing more is sent until the ABR reboots or the script changes. This is because every time the `onResults()` function runs, the output object is emptied.

```
var counter = 0;

function onResult(result, output) {
    // this works only once
    if (counter == 0)
    {
        output.setMessage('Hello World!\n');
    }

    counter = counter + 1;
}
```

Sending Separate Data to Different Channels

An option with the `setMessage` command specifies that the message gets sent not to all channels, but to one specific channel, using the channel names listed in ABR on the **Communications** page.

Figure 88. Channel Names



The following example shows how to send a default message to all active ports, but then change the message to the TCP Server port.

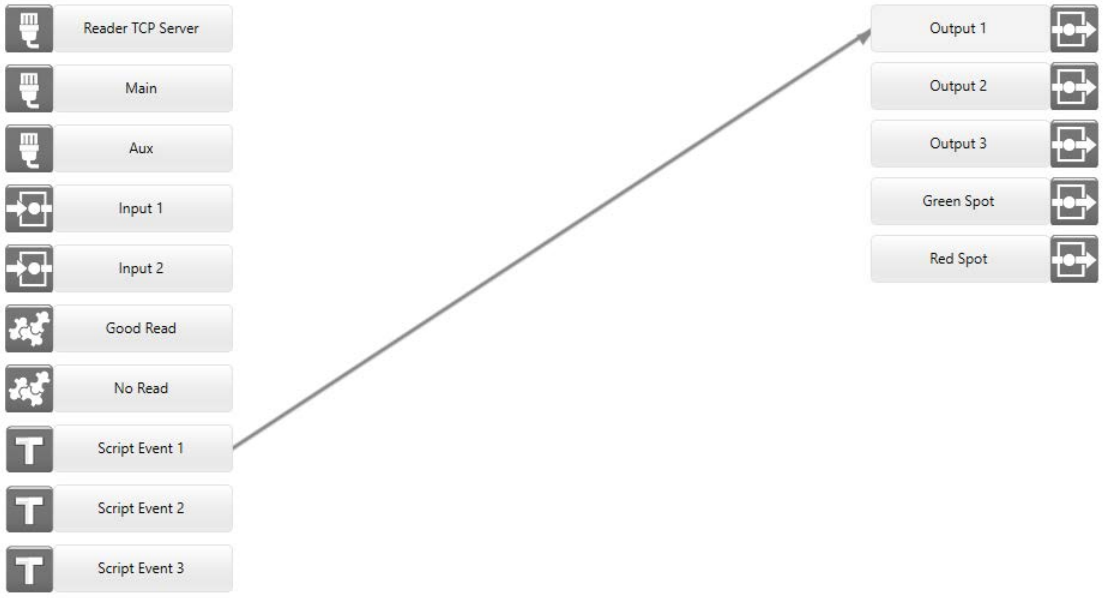
```
var EndLine = '\n';

function onResult(result, output) {
    // this is the default message
    output.setMessage('Hello everybody!' + EndLine);
    //this is a specific channel message
    output.setMessage('Hello TcpServer!' + EndLine,
                                'tcpserverchannel0');
}
```

Controlling Discrete Outputs

There are three available script events that can be triggered from a script and then connected on the Output Setup page to enable discrete outputs. This example script activates **Script Event 1**, and by connection **Output 1** if configured as in Figure 89 on p. 72, on the first Reading Phase after power up.

```
var counter = 0;

function onResult(result, output) {
    // this works only once
    if (counter == 0 )
    {
            output.setEvent1();
    }
    counter = counter + 1;
```

Figure 89. Tying Script Events to Physical Discrete Outputs



Saving Text Files Over FTP

The `setFTPFileName` property of the output object is used to control the filename and to select the FTP Client channel used for sending a text file to an FTP server. This can be used for data logging purposes. This function only writes a new file or overwrites an old file, it does not append data to an existing text file.

The following script shows how to save the string "message" into a text file named `data1.txt`:

```
function onResult(result, output) {
output.setMessage('message');
output.setFtpFileName('data1.txt');
}
```

This following script shows how to save the same string and file over a 2nd FTP Client channel:

```
function onResult(result, output) {
output.setMessage('message');
output.setFtpFileName('data1.txt','ftpclientchannel2');
}
```

Script-Controlled Image Saving Over FTP

The `saveImages` property of the output object is used to control the sending of images over FTP. This including what images are sent and what the image files are named. To use this function you must supply at least one argument, a number or array of numbers, or comma-separated list of numbers representing the image IDs of the images you want to save.

For example, in a result analysis where only one image is taken, the following script saves the image, using the default naming:

```
function onResult(results, output)
{
 var imageID = results.imageId;  // Load the image ID
 output.saveImages(imageID);  // Save the image using default naming
}
```

This function only works if you configure an Images Saving Condition for FTP and enable the **Image Saving Controlled by Script** setting on the **Images Saving Conditions** page.

The optional second argument is used for specifying the filename of the image. The following is an example of a script that saves the only image taken, but with the filename set to be "Image" plus the imageID number:

```
function onResult(result, output) {
 var ImageIDandFilename = [];
 var msg = 'Image' + result.imageId;
 ImageIDandFilename.push({id: result.imageId, filename: msg});  //load the array with the new filename
 output.saveImages(ImageIDandFilename);   // Save the image with custom name specified before
}
```

Using Array Variables

Arrays are variables that contain multiple values or elements and have an index value in `[]` brackets to specify the address of the element to be used.

```
var myfriends = ['Alice', 'Bob', 'Carl'];

function onResult(result, output) {
    var msg;    // a local variable
        // writes the number of friends
    msg = myfriends. length + ' friends: ';
        // adds their names
    msg = msg + myfriends[0];
    msg = msg + ',' + my friends[1];
    msg += ',' + myfriends[2];
    output.setMessage(msg + '\n');
}
```

This script results in the full array contents and its length sent. It appears in the Barcode Manager Console like this:



Useful String Methods

The following methods may be useful when dealing with text strings and the need to search or edit the string.

String Method	Description
`indexOf(`*`search =Value[, fromIndex]`*`)`	Returns the first position of the specified substring
`lastIndexOf(`*`searchValue[, fromIndex]`*`)`	Returns the last position of the specified substring *(Ecmascript 6 but they work)*
`startsWith(`*`searchString[, position]`*`)`	Returns `true` if starts with the specified substring
`endsWith(`*`searchString[, length]`*`)`	Returns `true` if ends with the specified substring
`includes(`*`searchString[, position]`*`)`	Returns `true` if contains the specified substring
`repeat(`*`count`*`)`	Returns a string consisting of *count* string repeats
`substr(`*`start[, length]`*`)`	Returns the specified subset of the string, by specifying the *start* and a *length*
`substring(`*`start[, end]`*`)`	Returns the specified subset of the string, by specifying the *start* and *end* indexes
`slice(`*`start[, end]`*`)`	Like substring, extracts a section of a string and returns a new string
`toLowerCase()`	Returns the string in all lowercase
`toUpperCase()`	Returns the string in all uppercase
`trim()`	Trims whitespace from the beginning and end of the string

Using Regular Expressions for Complex String Matching

Regular expressions are patterns used to match character combinations in strings. In JavaScript, regular expressions are also objects. For example, regular expressions can be written as:

Expression	Description
`/ab+c/i`	A constant *regular expression* that matches any string beginning with `a` or `A`, followed by at least `b` or `B`, and a `c` or `C`

Expression	Description
`RegExp('ab+c', 'i')`	Same as the previous expression, but in this case the arguments can also be variables. The first part specifies the pattern to be matched. The second part can specify zero, or one or more flags. For example: • `i`—ignores case in match • `g`—matches the pattern globally in the string

The following is a list of regular expression special characters:

`.`	(the dot) matches any single character except line terminators
`\d`	matches any digit (arabic numeral); equivalent to [0-9]
`\w`	matches any alphanumeric character from the basic Latin alphabet; equivalent [A-Za-z0-9]
`\s`	matches a single white space character, including space, tab, form feed, line feed
`[xyz]`	`[a-c]` matches any one of the enclosed characters; the `-` hyphen specifies a range of characters; a `^` carat in the first character specifies the complementary set
x*`|`*y	matches either *x* or *y patterns*
`^`	matches the beginning of the string
`$`	matches the end of the string
(*x*)	matches *x* pattern and remembers the match
`\`*n*	references back the *n*-th pattern remembered
(?:*x*)	references *x* pattern and does not remember the match
x*	matches the preceding item *x* 0 or more times
***x*+**	matches the preceding item *x* 1 or more times
***x*?**	matches the preceding item *x* 0 or 1 times
***x*{*n*}**	matches exactly *n* occurrences of the item *x*
***x*{*n*,}**	matches at least *n* occurrences of the item *x*
***x*{*n*,*m*}**	matches *n* to *m* occurrences of the item *x*

Regular expressions can be very complicated. For more information, see: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions.

The following are additional string methods that use regular expressions:

String Method	Description
`match(`*regexp*`)`	Returns an array with the matching parts of a string
`replace(`*regexp*`|`*substr*`, `*newSubstr*`|`*function*`)`	Returns a new string with some or all matches of a pattern replaced
`search(`*regexp*`)`	Returns the position of the first match between a regular expression and this string
`split(`*[separator[, limit]]*`)`	Splits a String object into an array of substrings; *separator* specifies the string at which each split should occur and can be a string or a regular expression; *limit* specifies maximum number of splits

9.6.2 Data Objects

The data that can be referenced in scripts often need to be referenced as a value within a property, that is nested within another property. For example, the X coordinate of the center of the code that fulfilled Expected Code slot 1 in Code Combination mode is typically referenced in a script as:

```
result.slots[0].codes[0].center.x
```

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 p. 75).
slots	Array	The array of all the configured Slot objects (see Slot Object on p. 77). 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 p. 77).
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.
redingCount	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.
imageProcessingTime	Number	Average time used to decode an image.
imageAquisitionTime	Number	Average time used to acquire an image (milliseconds).
imageWaitTime	Number	Average time the device is busy between acquisition and decoding of an image (milliseconds).
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.

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 p. 77).
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. (See www.AIMglobal.org.)
angle	Number	The code angle [degrees].
ppe	Number	The code pixels per element.
moduleSize	Number	The code module size [mils].

Property	Type	Description
decodingTime	Number	The decoding time in microseconds.
overrunTime	Number	The time in microseconds elapsed after the tool timed out.
symbolSize	String	The code symbol size (if applicable).
quality	QualityMetrics	The Quality Metrics object that contains all the quality metrics info for the code (see Quality Metrics Object on p. 76).
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.

Parameters Object – For External Access to Script Variables with Host Mode Programming (HMP)

The Parameters global object allows adding custom variables that can be set and requested over Ethernet or Serial ports using Host Mode Programming. To set up a variable for external access, define a variable like the following example, called matchcode:

```
Parameters.matchcode = 'Alice';
```

A host controller can read the value of this variable using the following HMP command:

```
SCRIPT_GET_PARAM matchcode<CR><LF>
```

Or the host controller can write a new value to this variable using the following HMP command:

```
SCRIPT_SET_PARAM matchcode "Bob"<CR><LF>
```

This technique is useful in complex match code applications. See Host Mode Programming on p. 79 for info on HMP.

Quality Metrics Object

QualityMetrics object type 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

For example, the reference:

```
result.codes[0].quality.overall.grade
```

returns an "A" for a good barcode with an overall quality grade of A.

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.

Point Object

Point Object Type 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). Point objects are contained within the center and corners properties within the code object.

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 associated with the Expected Code. The slot object is within the result object.

Property	Type	Description
codes	Array	The array of all the Code objects assigned to this slot (see Code Object on p. 75).

Image Object

Image Object provides access to the data of a single decoded image, which are within the result object.

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

Statistics Object

Statistics Object provides access to global device statistics.

These values can be reset by a power cycle or with Barcode Manager software. Since these are global values, they are not contained within any other object, and are references in the script like this:

`Statistics.fps`—references the current frame rate of the ABR

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.

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 ID-NET 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. So the reference

  ```
  Diagnostics.slaveNoReply.alarm
  ```

  would be true if a Slave No Reply alarm is active

Property	Type	Description
slaveNoReplay	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.
protocolIndexFailure	Object	Protocol Index 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.

9.6.3 Example Scripts

See the latest example scripts at www.bannerengineering.com.

10 Host Mode Programming

10.1 Conventions

The notation in this manual uses angular brackets (< >) to represent **single byte character values** which are equivalent. See . For special keys or some non-printable characters, the ASCII symbol of the character name is shown between angular brackets.

Example:

<ESC> means the ESC key character byte

<STX> means the start transmission character byte

<space> means the whitespace character

For non-printable characters, the decimal value of the character byte is shown between angular brackets.

Example:

<13><10> = Carriage Return character byte, Line Feed character byte

The following representations using these three methods are equivalent:

<ESC> = <27> = <$1B_H$>

and each sends character byte 0x1B to the reader.

The program used to generate the strings for Host Mode Programming must be able to manage them correctly.

10.2 Host Mode Programming

Host Mode programming allows the performance of a partial configuration of the reading device as an alternative method to configuration using Barcode Manager. It consists of substituting or modifying the ABR Family device configurations by sending command strings over one of the communication channels.

Probably the most useful command is to change configuration jobs from among those saved in the device memory, directly from the Host.

These strings take the form of command sequences that are specific to each saved configuration. They can be personalized to make command recognition easier. They are transmitted from the Host system to the ABR reader on the following interfaces:

- On-board Ethernet TCP/IP Server interface
- Main RS232/RS422 serial interface
- Auxiliary RS232 serial interface

To send the programming strings, switch the reader to **Host Mode**.

Important: Barcode Manager must be disconnected from all reader interfaces before entering Host Mode. By entering Host Mode, any Barcode Manager Monitor or Smart Teach connections to the reader will be closed.

Ethernet Interface

The programming commands and strings can be sent to the reader using the on-board Ethernet TCP/IP Server interface. Connect to the device IP address, port number: 1023 (pre-defined).

Serial Interfaces

The programming commands and strings can be sent to the reader at the programmed baud rate of the selected serial interface. For example, if the baud rate of the Auxiliary interface is programmed at 115200 bps, the command must be sent at 115200 bps.

10.2.1 Rockwell Automation PLC Support for HMP Commands by Ethernet

Only certain Rockwell Automation PLC's support open socket connections, allowing them to send HMP commands as a TCP Client to the ABR's TCP Server interface. This is a partial list of hardware that supports sockets. See Rockwell support document 470690 for more information:

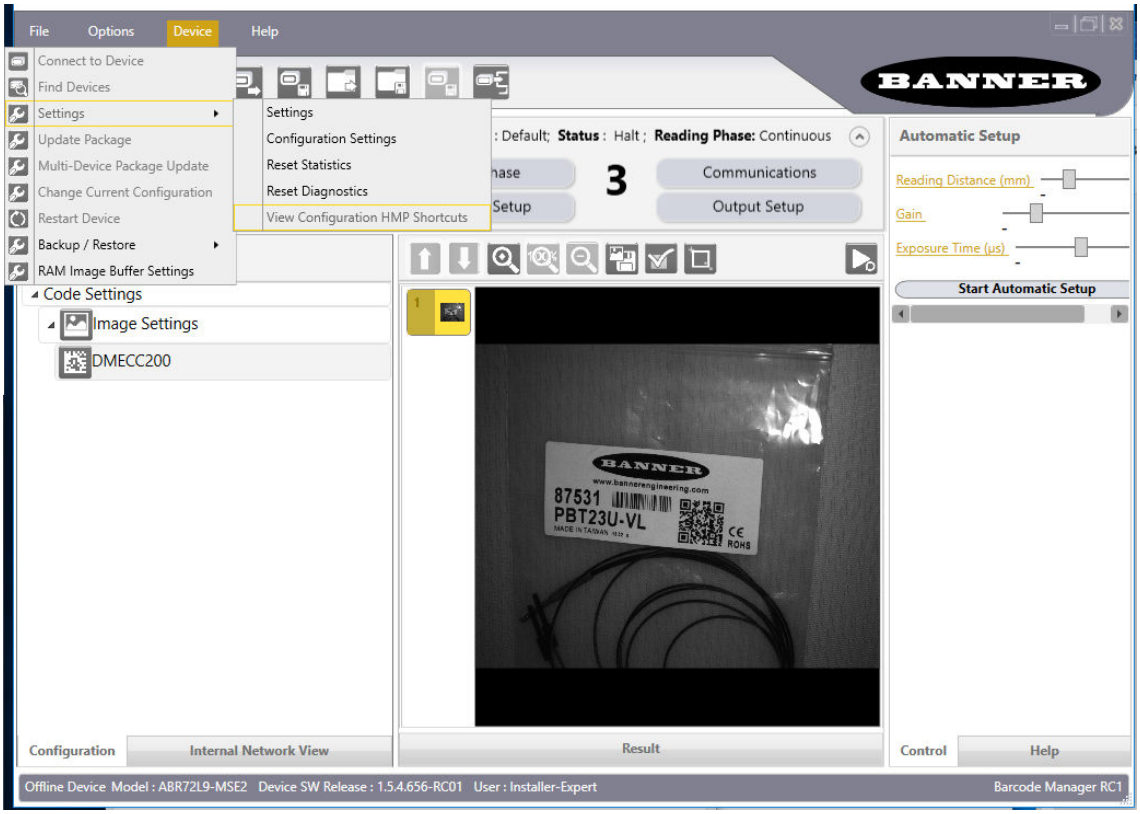
- 1756-EN2T, 1756-EN2F, 1756-EN2TR, 1756-EN3TR ControlLogix® Ethernet/IP communication modules, firmware revision 5.007 or later
- 1756-EWEB ControlLogix Ethernet/IP web server module, firmware revision 4.006 or later
- 1768-EWEB CompactLogix Ethernet/IP web server module, firmware revision 1.002 or later

- 1769-L30ER, 1769-L30ERM, 1769-L30ER-NSE, 1769-L33ER, 1769-L33ERM, and 1769-L36ERM CompactLogix controllers, firmware revision 20.011 or later
- 1769-L24ER-QB1B, 1769-L24ER-QBFC1B, 1769-L27ERM-QBFC1B CompactLogix controllers, firmware revision 20.011 or later
- 1769-L16ER, 1769-L18ER, 1769-L18ERM CompactLogix controllers, firmware revision 20.011 or later
- Micrologix™ 1400

10.2.2 Collecting HMP Strings from Barcode Manager

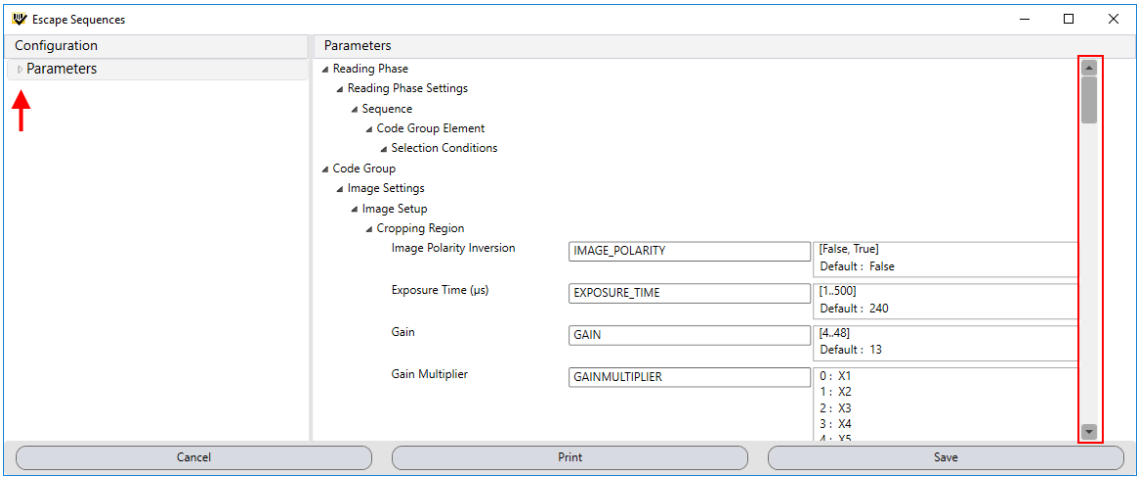
First, collect the application specific Programming Strings from the Barcode Manager Configuration. Go to **Device** > **Settings** > **View Configuration HMP Shortcuts**.

Figure 90. Menu Tree to Select View Configuration HMP Shortcuts



This opens the **Escape Sequences** window.

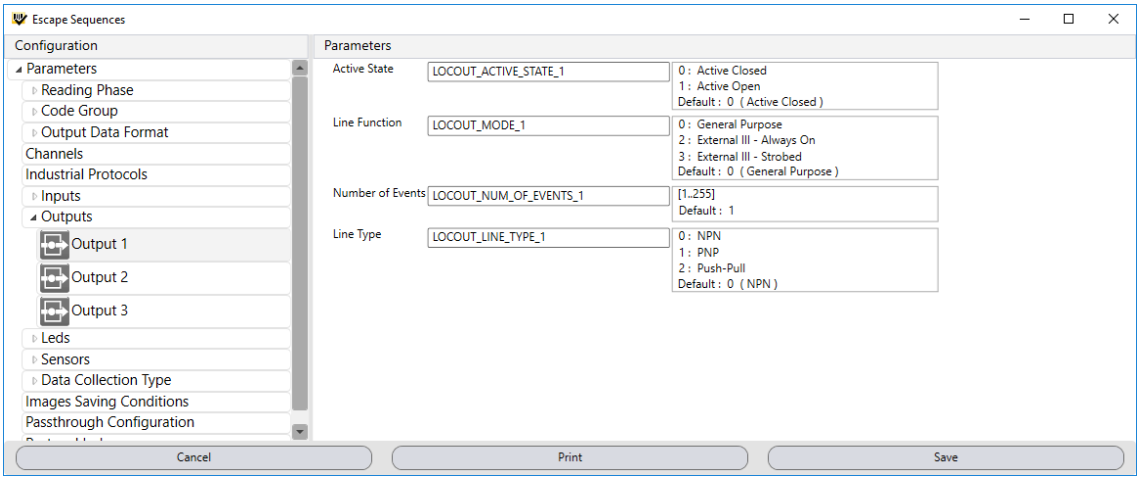
*Figure 91. **Escape Sequences** Window*



To view the full extension of configuration parameters, drag the vertical slider on the right side of the window. Otherwise click the **Parameters** triangle to open the tree to select any individual branch and view only the branch specific parameters.

Each configurable parameter is shown with its programming string name and a list of its configuration parameter values, including the default setting.

Figure 92. Configuring Parameters



To personalize the programming strings from any view in this window, click the string name and enter a new name (no spaces are allowed). Then click **Save**.

Click **Print** to produce an HMP reference sheet.

10.3 Programming Commands

10.3.1 Connection to Device

	Description	Host Command	Reply Message
1	**Enter Host Mode**	<ESC>[C	<ESC>H<CR><LF>
	After entering this command, the device responds with the first reply message and then waits for the following command.		
2	**Enter Programming Mode**	<ESC>[B	<ESC>S<CR><LF>
	After entering this command, the device responds with the second reply message. One or more programming strings, as shown in Programming Strings on p. 81, can now be sent.		
3	**Exit Host Mode**	<ESC>[A	<ESC>[X
	This command must always be transmitted to end the programming session. The configuration job management depends on the commands issued prior to exit from Host Mode. If the SAVE command was issued before exit from Host Mode, then the job is saved in permanent memory (Flash). If the SAVE command was not issued before exit from Host Mode, then the job is saved in temporary memory (RAM).		

10.4 Programming Strings

10.4.1 Definitions

Common definitions for each parameter are:

Name (parameter name)

Name of the parameter available in the Barcode Manager Configuration Parameters panel. The other definitions change according to the parameter type.

Type (parameter type)

Parameter type is essential to decide the features of the command used to set its new value. Types are:

Type 0: Integer (numeric)
Type 1: Enumerative (numeric list)
Type 2: BOOLEAN (true, false)
Type 3: String (printable characters)

Integer (Type = 0)

Range: Minimum and maximum values allowed for the parameter

Default: Default value (integer)

Example: *Exposure Time (µs)* parameter

- Name: Exposure Time (µs)
- Type: 0
- Range: 6 to 500 (depending on model)
- Default: 240 (depending on model)

Enumerative (Type = 1)

Item List: List of the values allowed for the parameter

Default: Default entry (0 = first entry of the list, 1 = second entry of the list, etc.)

Example: *Gain Multiplier* parameter

- Name: Gain Multiplier
- Type: 1
- Item List: X1, X2, X3, X3, X4, X5, X6, X7, X8, X9, X10
- Default: 0 (Entry 0 = X1)

BOOLEAN (Type = 2)

Value: true = Enabled; false = Disabled

Default: Default value (String)

Example: *Identical Codes* parameter

- Name: Identical Codes
- Type: 2
- Value: false, true (case sensitive)
- Default: false (Disabled)

String (Type = 3)

Length: Minimum to maximum number of characters allowed for this parameter

Default: Default value (String)

Example: *Wildcard* parameter

- Name: Wildcard
- Type: 3
- Length: Min to Max number of allowable characters in string
- Value: Any character from 00 to FF (hex)
- Default: *

10.4.2 Programming Strings

Set Parameter Value in Current Configuration

The ABR device single parameter programming string must have the following format:

```
SET<space>PSTR<space>VAL <LF>
```

where:

- SET: The programming command (all caps)
- PSTR: The unique parameter string containing no spaces
- VAL: Character or string indicating the assigned parameter value (case sensitive)

If the command is correct, the device immediately implements the new value and confirms with the following message:

```
ACK<LF>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

and programming data will not be updated in this case.

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Get Parameter Value in Current Configuration

The ABR device single parameter get string must have the following format:

```
GET<space>PSTR <LF>
```

where:

- GET: The programming command (all caps)
- PSTR: The unique parameter string containing no spaces

If the command is correct, the device replies with the value string for the current parameter:

```
ACK VAL <LF>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Get Configurations List (Jobs)

The ABR `get configurations list` command must have the following format:

```
GET_JOBS_LIST <LF>
```

If the command is correct, the device confirms with the following message:

```
ACK<LF>
Default <LF><CR>
.... <LF> <CR>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Save Current Configuration (Job)

The ABR `configuration saving` command must have the following format:

```
SAVE<space>configuration_name <LF>
```

After making changes to a configuration, this command should always be transmitted before the exit from programming mode command.

If the command is correct, the device confirms with the following message:

```
ACK<LF>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Note: To overwrite the current configuration, the configuration name must exactly match the name saved on the device. Names can contain spaces and are case sensitive. If the name does not match, then a copy of the configuration will be saved with the new name. The Default configuration cannot be overwritten.

Set Startup Configuration (Job)

The ABR `startup configuration setting` command must have the following format:

```
STARTUP_CFG<space>configuration_name <LF>
```

If the command is correct, the device confirms with the following message:

```
ACK<LF>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Note: The configuration names must match those saved on the device. They can contain spaces and are case sensitive. This command does not change the current configuration. It also does not require the SAVE command to be issued.

Change Current Configuration (Job)

The ABR c`hange configuration` command must have the following format:

```
CHANGE_CFG<space>configuration_name <LF>
```

If the sequence is correct, the device confirms with the following message:

```
ACK<LF>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Note: The configuration names must match those saved on the device. They can contain spaces and are case sensitive. This command does not change the Startup Configuration.

Match Code Content

The ABR `match code` command must have the following format:

```
MATCH<space>code_content <LF>
```

After using this command, the SAVE configuration command should always be transmitted before the exit from host mode programming command.

If the command is correct, the device replies with the code_content string for the Expected Code parameter of the current Match Code configuration:

```
ACK code_ content <LF>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, or if the current configuration is not a Match Code data collection type, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Note: The configuration must be a Match Code data collection type previously saved on the device. The code_content can contain spaces and is case sensitive. This command overwrites the current Expected Content parameter value. If the Input Slots setting is greater than one, so that there are multiple slots for expected code content, then the Match command will overwrite the oldest code content in the list.

Get Match Code Expected Content

To get the expected content of a match code configuration, use the following format:

```
GET_ASCII MATCH_CODE_EXPECTED_CONTENT1<CR> <LF>
```

If the command is correct, the device replies with the expected_content string (in ASCII format) of the current Match Code configuration:

```
ACK expected_content <LF>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, or if the current configuration is not a Match Code data collection type, the device replies with the following message:

```
NACK <LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Send Configuration (Upload Job) to Device

Important: This command requires a dedicated program to handle file transfer protocol.

It is possible to send a .bncfg (job) file to the ABR device, saving it to permanent memory.

Note: Any ABR configuration can be easily exported to a .bncfg file from Barcode Manager by selecting **File** > **Save** > **Save Configuration on PC**.

The send configuration to ABR device command must have the following format:

```
SEND_CFG<space>configuration_name <LF>
```

Within 10 seconds of sending the SEND_CFG command, the .bncfg configuration file transmission must be initiated. The entire file content must be encapsulated using a CRC32 checksum control and transmitted to the device using X-Modem protocol.

Failing to initiate transmission within 10 seconds results in the device replying with a Timeout message, and the command is aborted.

The byte stream containing the configuration that is sent to the device is structured with a 4-byte header containing the total stream length, followed by a data_buffer section containing the whole .bncfg byte sequence, followed by a 4-byte CRC32 terminator storing the value computed on the previous length + data_buffer sections content.

Length (4 bytes)	Data Buffer	CRC (4 bytes)

Length (4 bytes): Specifies the number of bytes of the message (Data buffer + CRC field).

Data buffer: The bytes of the file to transmit.

CRC: The CRC32 checksum based on both Length + Data buffer.

Important:

- The data format used is BIG ENDIAN
- The packet timeout during transmission is 10 seconds: on timeout expiration command is considered aborted
- Only .bncfg files are allowed for upload

If the transmission is completed correctly, the device confirms with the following message:

```
ACK<LF>
```

If transmission is not initiated within 10 seconds, the device replies with the following message:

```
Timeout! <LF>
```

If the command is written incorrectly (for example, file format, content, or CRC error) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Note: The saved configuration has the name given in the configuration_name parameter. Names can contain spaces and are case sensitive. Do not include the extension. If a configuration with the same name already exists in the device memory, it is overwritten by the new file. This command does not change the Current Configuration nor the Startup Configuration. Use the appropriate commands to change it.

Note: There is no Host Mode Programming command available for downloading a configuration from a device.

Get Device Information

The ABR `get device information` command must have the following format:

```
GET_INFO <LF>
```

If the sequence is correct, the device confirms with the following message:

```
ACK Device Information <LF>
```

The following is an example of a successful response:

```
ACKModel name: AVR7:116-RSE2<LF><CR>
MAC address: 00:07:BE:06:79:32<LF><CR>
Serial number: C18G08093<LF><CR>
Device name: ABR<LF><CR>
Application version: 1.5.4.661<LF><CR>
Current Job: Conveyor1Reader1<LF><CR><LF>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Send Images

For configurations that allow Image Saving—Send Image On: Demand, it is possible to start the FTP images transfer from the device to an FTP Server through this command.

The ABR `send images` command must have the following format:

```
SEND_IMAGES<space><NodeNumber>[<space><IdStart>][<space><IdStop>] <LF>
```

where:

`SEND_IMAGES`:

The programming command (all caps).

`NodeNumber`:

(mandatory)
Specifies the node number (0 = Master, 1 to 31 = Slaves).
The asterisk character * can be used to send a broadcast command. In this case the Master and all Slaves will start to transfer images to the FTP Server.

`IdStart`:

(optional)
The phase number from which image transfer begins.

`IdStop`:

(optional)
The phase number from which image transfer ends.

Note: If the optional parameters are omitted, the entire image buffer of the device (or devices) will be sent.

Pre-requisites

- The device configuration runs in Phase Mode or OneShot
- An FTP Server is available and configured to allow uploads on the same TCP/IP network
- After entering the host programming sequence, wait at least 50 ms for the device to correctly setup internal data structures before issuing SEND_IMAGES commands
- The device configuration (Communications) must include an FTP client channel which must be configured to access the FTP Server (user/password, folder, ...)
- The device configuration (Communications) must include an Image Saving Condition, which must be configured with:
  - **Image Destination** > **On Remote**
  - **Send Image On** > **Demand**
  - **Saving Condition** > **[choose condition]** —Good Reads only, No Reads only, Both
  - **Image Format** > **[choose format]**—available formats are RAW, BMP, PNG, JPG, Tiff
  - **Remote Folder** > **[choose subfolder name on the FTP Server under which images will be saved]**
  - **Send Image From** > **ABR FTP Client**

See Image Saving for more details.

Saved Image Filenames

On the FTP Server, the saved images filenames have the following syntax:

`IMG_nnnnnnnnnn_ss_ff_ccc.ext`

where:

`nnnnnnnnnn`: phase# 10 digits
`ss`: slave id (00 for Master, 01-31 for slaves)
`ff`: frame sequence # for the current image within the related phase
`ccc`: # of valid barcodes detected by the reader who produced this image (000 for noread)
`ext`: extension to indicate image format (RAW, JPG, BMP, PNG, TIFF)

If the transmission is completed correctly, the device confirms with the following message:

`ACK<LF>`

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

`NACK<LF>`

Note: The transferred images come from the device internal circular buffer, which is used for this specific purpose. The buffer size depends on the complexity of the configuration as it is allocated during runtime. The buffer is managed in a round-robin fashion, therefore requesting of phases that are too old in the buffer may result in the reader not being able to transmit images, even if returning an ACK response (to be changed in future releases). On IdStart>IdStop errors, NACK is returned.

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Images can also be sent to a TCP Channel On-Demand.

Important: Sending images on demand to a TCP channel requires a dedicated program to correctly handle file transfer.

Get Command Status

This command generically returns information about the last command executed on the device. It is currently used after a `Send Images` command to monitor the downloading progress.

Note: Take into account that as the device continues to capture images, the circular image buffer changes and may be overwritten.

The ABR `Get Command Status` command must have the following format:

`GET_CMD_STATUS <LF>`

If the last command sent to the device was `SEND_IMAGES`, the device responds with the following message:

`SEND_IMAGES<space>ErrorCode[<space>Info1][<space>Info2][<space>Info3][<space>Info4] <LF>`

where:

Error Code	Description
0	No error
1	Generic error
10	Command not found (syntax error)
11	Wrong arguments (syntax/semantic)
20	Send images in progress
21	Send images ok (completed)
22	Send images failed

and:

Info Field	Description
Info1	Number of images sent
Info2	Total number of images saved in device image buffer
Info3	Beginning image phase # range
Info4	Ending image phase # range

Example Get Command Status responses on success

Host	<>	Device	Description
SEND_IMAGES * 3 10	>		Host requests device to send images from phase #3 to phase #10.
	<	ACK	Device accepts command.
GET_CMD_STATUS	>		Host requests last command status info.
	<	SEND_IMAGES 20	Device reports SEND_IMAGE is still in progress.
GET_CMD_STATUS	>		Host requests last command status info.
	<	SEND_IMAGES 21 1 3 10 12	Device reports SEND_IMAGES has completed successfully: sent 1 image from a total of 3 located in phase range #10 to #12.

Example Get Command Status responses on failure

Host	<>	Device	Description
SEND_IMAGES * 3 10	>		Host requests device to send images from phase #3 to phase #10.
	<	ACK	Device accepts command.
GET_CMD_STATUS	>		Host requests last command status info.
	<	SEND_IMAGES 20	Device reports SEND_IMAGES is still in progress.
GET_CMD_STATUS	>		Host requests last command status info.
	<	SEND_IMAGES 22 0 0 -1 -1	Device reports SEND_IMAGES has failed: 0/0 images sent.

For most other commands the response is: `NACK<LF>`

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

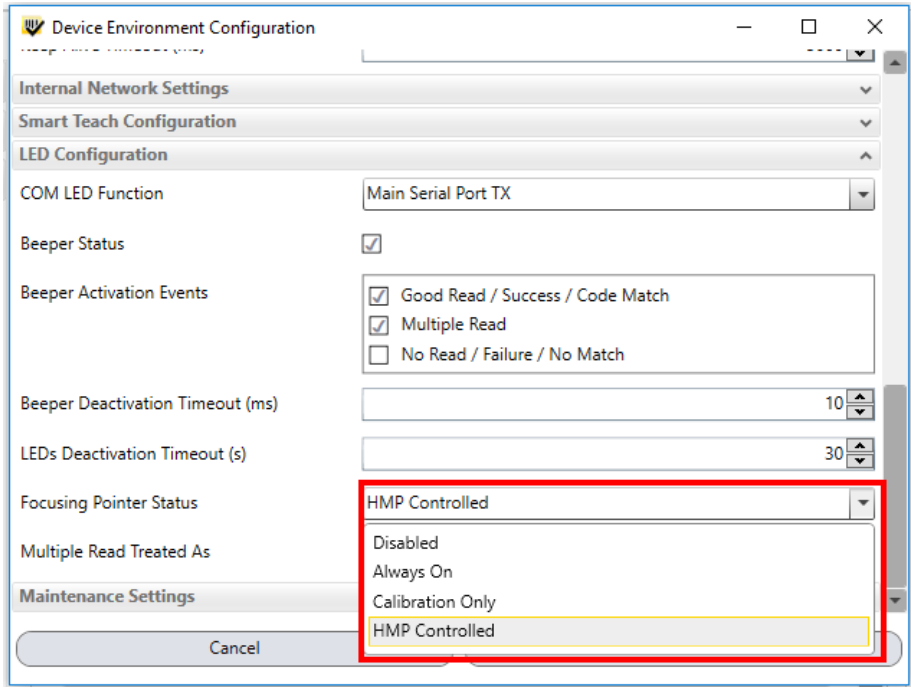
`NACK<LF>`

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Set Focusing Pointer

This command can be used only if the device has the **Focusing Pointer Status** set to **HMP Controlled** in the Barcode Manager **Device Environment Configuration** settings.

Figure 93. Setting the Focusing Pointer Status



The ABR `device set pointer` command must have the following format:

`SET_POINTER<space><ON|OFF> <LF>`

If the command is correct, the device immediately sets the focusing pointer ON or OFF and confirms with the following message:

`ACK<LF>`

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message and programming data will not be updated in this case:

`NACK<LF>`

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Note: This setting is not saved in permanent memory. The Focusing Pointer defaults to the OFF state when power is cycled.

Set Image Cropping Region

To set the image cropping region, use the following format:

`SET_CROPPING <left> <top> <width> <height> [<image-settings-index>]<LF>`

Where:

- `<left>` is the cropping region offset from the image sensor left edge
- `<top>` is the cropping region offset from the image sensor top edge
- `<width>` is the cropping region width
- `<height>` is the cropping region height
- `<image-settings-index>` (optional, default 1) is the index of the Image Settings to modify; this allows the use of the command with multiple Image Settings

If the command is correct, the cropping region is applied, and the device confirms with the following message:

```
ACK <LF>
```

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message and programming data will not be updated in this case:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Note: If the parameters do not respect the image sensor constraints (that is, width and right multiple of 32), the command will fail.

Read Input

The ABR `read input` command must have the following format:

```
READ_INPUT<space><Input1|Input2> <LF>
```

If the sequence is correct, the device confirms with the following message:

```
ACK Input n state <LF>
```

Input n state depends on the configured active state.

if Active Closed (default): 0 (open—no current flow); 1 (closed—current flow); -1 (error)
if Active Open: 0 (closed—current flow); 1 (open—no current flow); -1 (error)

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Read Output

The ABR `read output` command must have the following format:

```
READ_OUTPUT<space><Output1|Output2|Output3> <LF>
```

If the sequence is correct, the device confirms with the following message:

```
ACK Output n state <LF>
```

Output n state depends on the configured active state.

if Active Closed (default): 0 (open—no current flow); 1 (closed—current flow); -1 (error)
if Active Open: 0 (closed—current flow); 1 (open—no current flow); -1 (error)

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

```
NACK<LF>
```

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

Run Smart Teach Function

The ABR `Smart Teach` command must have the following format:

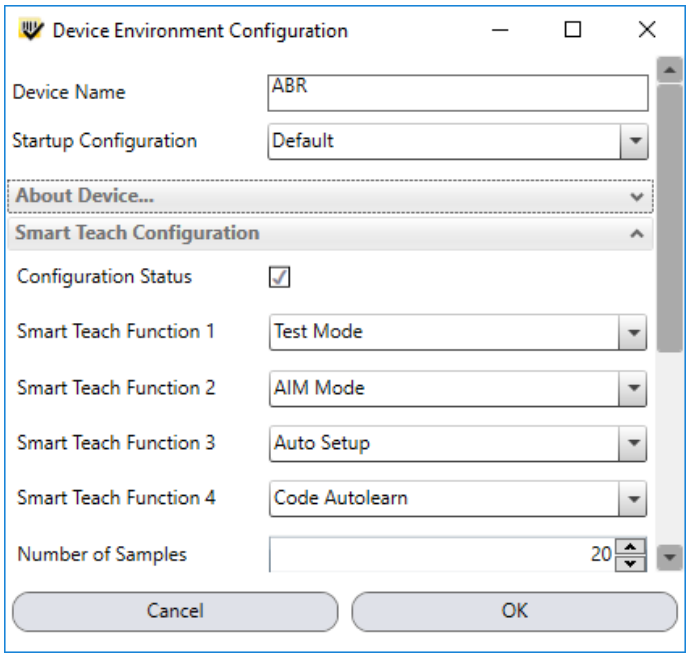
```
XPRESS<space><1|2|3|4> <LF>
```

If the sequence is correct, the device immediately executes the Smart Teach function associated with the selected number and confirms with the following message:

```
ACK<LF>
```

The function that is selected depends on how the reader is configured. The default settings are shown and modified in the **Settings** window from the **Device** menu:

Figure 94. Configuring Smart Teach Function



To exit the Smart Teach function before its associated timeout, execute the same command a second time (toggle). If the associated timeout has already expired, then executing the Smart Teach command a second time will run the function again.

If the command is written incorrectly (for example, a typing error in the file) or is corrupted in transmission, the device replies with the following message:

`NACK<LF>`

After transmitting the reply message, the device waits for another programming string or the `exit from host mode programming` command.

10.4.3 Examples

1. Set Image Setting 2 *Exposure Time* parameter to 400 µs:
   - Type: 0
   - Range: 6 to 500
   - Data: 400

   The programming string is: `SET<space>EXPOSURE_TIME1<space>400 <CR><LF>`
2. Set *Gain Multiplier* parameter to X3:
   - Type: 1
   - List: X1, X2, X3, X4, X5, X6, X7, X8, X9, X10
   - Data: 2

   The programming string is: `SET<space>GAINMULTIPLIER<space>2 <CR><LF>`
3. Set Data Matrix ECC 200 *Image Mirroring* parameter to *enabled*:
   - Type: 2
   - Item List: true (Enabled), false (Disabled)
   - Data: true

   The programming string is: `SET<space>2D_IMAGE_MIRRORING<space>true <CR><LR>`
4. Set Data Matrix ECC 200 Image Filter *Placeholder* character to &:
   - Type: 3
   - Item List: any printable character
   - Data: &

   The programming string is: `SET<space>2D_PLACEHOLDER<space>& <CR><LF>`
5. Transfer all pending images from all devices in the cluster:
   The programming string is: `SEND_IMAGES<space>*<CR><LF>`
6. Transfer images from slave 2 starting from phase 123:
   The programming string is: `SEND_IMAGES<space>2<space>123<CR><LF>`

7. Transfer images from Master for phases 10 to 12:
   The programming string is: `SEND_IMAGES<space> 0<space>10<space>12<CR><LF>`

10.5 ABR Family Parameters List

The following tables present a complete list of parameters that can be modified using Host Mode Programming.

Default parameter strings are listed here.

The strings can be modified (personalized) through Barcode Manager so it is advised to print them out from Barcode Manager for any specific configurations.

These strings are also dynamic. They depend on the order in which the configuration is made. For example, each group of 2D or 1D Code parameters is associated with STRINGx, where x is a dynamically assigned number depending on the chronological order in which it was configured. For this reason, it is important to know the relative string assigned by Barcode Manager. See Conventions for details.

10.5.1 Image Setup

PARAMETER	TYPE	STRING	DATA
Image Polarity Inversion	2	IMAGE_POLARITYx	false = Disabled true = Enabled
Exposure Time (µs)	0	EXPOSURE_TIMEx	
ABR71xx-xxxx and ABR72L9-RDE2			
Disabled			10 to 50000
Always On			1000 to 50000
Very High-Power Strobed			1 to 500
High-Power Strobed			100 to 3300
ABR72L9-RPE2 and ABR72L16-xxxx			
Disabled			10 to 1,000,000
Continuous High-Power			1 to 250
Very High-Power Strobed			1to 300
High-Power Strobed			100 to 3800
ABR3009-xxxx			
Disabled			10 to 15000
Very High-Power Strobed			10 to 1000
ABR3106-xxxx			
Disabled			1 to 15000
Very High-Power Strobed			1 to 1000

Note: The Exposure Time parameter is measured in µs and the range of valid values depends on the ABR model and the Internal Lighting setting.

PARAMETER	TYPE	STRING	DATA
Gain	0	GAINx	Range: ABR3009 models: 4 to 32 ABR3106 models: 8 to 32 ABR71xx models: 4 to 48 ABR72xx models: 7 to 48

PARAMETER	TYPE	STRING	DATA
Gain Multiplier	1	GAINMULTIPLIERx	0 = X1 1 = X2 2 = X3 3 = X4 4 = X5 5 = X6 6 = X7 7 = X8 8 = X9 9 = X10
Reading Distance (mm)	0	READING_DISTANCEx	Range: ABR71L9 models: 22 to 452 mm ABR72L9 models: 30 to 525 mm ABR72L16 models: 50 to 960 mm

10.5.2 Code Setup

2D Codes

Data Matrix ECC200			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	2D_ID_CODES	false = Disabled true = Enabled
Image Mirroring	2	2D_IMAGE_MIRRORINGx	false = Disabled true = Enabled
Grey Level Calibration	2	2D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Color	1	2D_CODE_COLORx	0 = Black 1 = White 2 = Both Colors
Code Contrast	1	2D_CODE_CONTRASTx	0 = High 1 = Standard 2 = Low 3 = Very Low 4 = Automatic
Decoding Complexity	1	2D_DECODING_COMPLEXITYx	0 = Low 1 = Medium 2 = High 3 = Very High
Code Size	1	2D_CODE_SIZEx	1 = Free 2 = Squared 3 = Rectangular
Advanced Box Improvement	2	2D_ADVANCED_BOX_IMPROVEMENTx	false = Disabled true = Enabled

QR			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	2D_ID_CODESx	false = Disabled true = Enabled
Decoding Method	1	2D_DECODING_METHODx	0 = Standard 1 = Dot Peen Decoding
Image Mirroring	2	2D_IMAGE_MIRRORINGx	false = Disabled true = Enabled
Grey Level Calibration	2	2D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Color	1	2D_CODE_COLORx	0 = Black 1 = White 2 = Both Colors
Code Contrast	1	2D_CODE_CONTRASTx	0 = High 1 = Standard 2 = Low 3 = Very Low 4 = Automatic

Micro QR			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	2D_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	2D_IMAGE_MIRRORINGx	false = Disabled true = Enabled
Grey Level Calibration	2	2D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Color	1	2D_CODE_COLORx	0 = Black 1 = White 2 = Both colors
Code Contrast	1	2D_CODE_CONTRASTx	0 = High 1 = Standard 2 = Low 3 = Very Low 4 = Automatic

Aztec			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	2D_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	2D_IMAGE_MIRRORINGzx	false = Disabled true = Enabled
Grey Level Calibration	2	2D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Color	1	2D_CODE_COLORx	0 = Black 1 = White 2 = Both Colors

Aztec			
PARAMETER	**TYPE**	**STRING**	**DATA**
Code Contrast	1	2D_CODE_CONTRASTx	0 = High 1 = Standard 2 = Low 3 = Very Low 4 = Automatic

Maxicode			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical codes	2	2D_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	2D_IMAGE_MIRRORINGx	false = Disabled true = Enabled
Grey Level Calibration	2	2D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Color	1	2D_CODE_COLORx	0 = Black 1 = White 2 = Both Colors
Code Contrast	1	2D_CODE_CONTRASTx	0 = High 1 = Standard 2 = Low 3 = Very Low 4 = Automatic
Partial Code Tx	2	2D_PARTIAL_CODE_TXx	false = Disabled true = Enabled

Dotcode			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	2D_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	2D_IMAGE_MIRRORINGx	false = Disabled true = Enabled
Grey Level Calibration	2	2D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Color	1	2D_CODE_COLORx	0 = Black 1 = White 2 = Both Colors
Code Contrast	1	2D_CODE_CONTRASTx	0 = High 1 = Standard 2 = Low 3 = Very Low 4 = Automatic
Enable High Resolution	2	2D_HIGH_RESOLUTIONx	false = Disabled true = Enabled
Min Pixel per Module	0	2D_MIN_DOT_DIAMETERx	Range: 0 to 48
Max Pixel per Module	0	2D_MAX_DOT_DIAMETERx	Range: 0 to 48

Dotcode			
PARAMETER	**TYPE**	**STRING**	**DATA**
Fixed Code Position	2	2D_CONSTANT_POSITIONx	false = Disabled true = Enabled

2D Code Filter			
PARAMETER	**TYPE**	**STRING**	**DATA**
Wildcard	3	2D_WILDCARD_CHARx	Length: 1 character
Placeholder	3	2D_PLACEHOLDERx	Length: 1 character
Min. Number of Characters	0	2D_MIN_CHARx	Range 1 to 3116
Max. Number of Characters	0	2D_MAX_CHARx	Range 1 to 3116
Code Quality Standard [11]	1	2D_CODE_QUALITY_STx	0 = Overall 1 = Decode 2 = Contrast 3 = Modulation 4 = Decodability 5 = Minimum Edge Contrast 6 = Axial Non Uniformity 7 = Unused Error Correction 9 = Minimum Reflectance 10 = Defects 11 = Fixed Pattern Damage 12 = Grid Non Uniformity
Code Quality Index	1	2D_CODE_QUALITY_INDEXx	0 = Any 1 = Grade A 2 = Grade B 3 = Grade C 4 = Grade D

1D Codes

PDF417			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled

[11] Values depend on Code Type and Code Quality Standard. Not all parameters are valid for a given code.

MicroPDF			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled

Code 128			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled
Start/Stop Characters Tx	2	1D_STARTSTOP_CHAR_TXx	false = Disabled true = Enabled
Characters Tx	2	1D_CHAR_TXx	false = Disabled true = Enabled

GS1-128			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both

GS1-128			
PARAMETER	**TYPE**	**STRING**	**DATA**
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SIBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled
Start/Stop Characters	2	1D_STARTSTOP_CHAR_TXx	false = Disabled true = Enabled
Characters Tx	2	1D_CHAR_TXx	false = Disabled true = Enabled

Code 39			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled
Character Set	1	1D_CHARACTER_SETx	0 = Standard 1 = Full ASCII
Code 32 Decoding	2	1D_CODE32_DECODINGx	false = Disabled true = Enabled

Interleaved 2 of 5			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both

Interleaved 2 of 5			
PARAMETER	**TYPE**	**STRING**	**DATA**
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled
Min Accepted Code Length	0	1D_INT25_ENABLED_LENx	Range 2 to 60

Pharma			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Minimum Code Height (mm)	0	1D_MIN_CODE_HEIGHTx	Range 1 to 500
Code Aspect Ratio	1	1D_CODE_ASPECT_RATIOx	0 = Standard 1 = Low
Code Contrast	1	1D_CODE_CONTRASTx	0 = Standard 1 = Low
Small Codes Improvement	2	1D_SMALL_CODES_IMPROVEMENTx	false = Disabled true = Enabled

MSI			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled

Plessey			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled

Standard 2 of 5			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled

Matrix 2 of 5			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled

Matrix 2 of 5			
PARAMETER	**TYPE**	**STRING**	**DATA**
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled

Codabar			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled
Start/Stop Characters Tx	2	1D_STARTSTOP_CHAR_TXx	false = Disabled true = Enabled

GS1 Databar Expanded			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled

GS1 Databar Expanded Stacked			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled

GS1 Databar Expanded Stacked			
PARAMETER	**TYPE**	**STRING**	**DATA**
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled

GS1 Databar Limited			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled

GS1 Databar			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled

GS1 Databar Stacked			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled

Note: GS1 DataBar option also enables GS1 DataBar Truncated symboloy, while GS1 DataBar Stacked option also enables GS1 DataBar Stacked Omni-directional symbology.

Code 93			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled

EAN 13			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled

EAN 13			
PARAMETER	**TYPE**	**STRING**	**DATA**
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled

EAN 8			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled

UPC A / UPC E			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	1D_ID_CODESx	false = Disabled true = Enabled
Grey Level Calibration	2	1D_GREY_LEVEL_CALIBRATIONx	false = Disabled true = Enabled
Code Orientation	1	1D_CODE_ORIENTATIONx	0 = Horizontal 1 = Vertical 2 = Both
Advanced Box Improvement	2	1D_ADV_BOX_IMPROVEMENTx	false = Disabled true = Enabled
Subpixel Decoding Improvement	2	1D_SUBPIXEL_IMPROVEMENTx	false = Disabled true = Enabled
Narrow Quiet Zone	2	1D_NARROW_MARGINSx	false = Disabled true = Enabled

1D Code Filter			
PARAMETER	**TYPE**	**STRING**	**DATA**
Wildcard	3	1D_WILDCARD_CHARx	Length: 1 character
Placeholder	3	1D_PLACEHOLDERx	Length: 1 character

1D Code Filter			
PARAMETER	**TYPE**	**STRING**	**DATA**
Min. Number of Characters	0	1D_MIN_NUMBER_CHARx	Range 1 to 3116
Max. Number of Characters	0	1D_MAX_NUMBER_CHARx	Range 1 to 3116
Code Quality Standard [12]	1	1D_CODE_QUALITY_STx	0 = Overall 1 = Decode 2 = Contrast 3 = Modulation 4 = Decodability 5 = Minimum Edge Contrast 6 = Axial Non Uniformity 7 = Unused Error Correction 9 = Minimum Reflectance 10 = Defects 11 = Fixed Pattern Damage 12 = Grid Non Uniformity
Code Quality Index	1	1D_CODE_QUALITY_INDEXx	0 = Any 1 = Grade A 2 = Grade B 3 = Grade C 4 = Grade D

Postal Codes

Australia Post			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	POSTAL_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	POSTAL_IMAGE_MIRRORINGx	false = Disabled true = Enabled
Code Orientation	1	POSTAL_CUSTOMER_FIELD_DECODINGx	0 = N Decoding Table 1 = C Decoding Table 2 = Bars Value (Decimal)

Royal Mail 45			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Code	2	POSTAL_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	POSTAL_IMAGE_MIRRORINGx	false = Disabled true = Enabled

[12] Values depend on Code Type and Code Quality Standard. Not all parameters are valid for a given code.

KIX Code			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	POSTAL_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	POSTAL_IMAGE_MIRRORINGx	false = Disabled true = Enabled

Japan Post			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	POSTAL_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	POSTAL_IMAGE_MIRRORINGx	false = Disabled true = Enabled

PLANET			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	POSTAL_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	POSTAL_IMAGE_MIRRORINGx	false = Disabled true = Enabled

POSTNET			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	POSTAL_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	POSTAL_IMAGE_MIRRORINGx	false = Disabled true = Enabled
POSTNET BB	2	POSTAL_POSTNETBBx	false - Disabled true = Enabled

Intelligent Mail			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	POSTAL_ID_CODESx	false = Disabled true = Enabled
Image Mirroring	2	POSTAL_IMAGE_MIRRORINGx	false = Disabled true = Enabled

Swedish Postal			
PARAMETER	**TYPE**	**STRING**	**DATA**
Identical Codes	2	POSTAL_ID_CODESx	false = Disabled true = Enabled

Swedish Postal			
PARAMETER	**TYPE**	**STRING**	**DATA**
Image Mirroring	2	POSTAL_IMAGE_MIRRORINGx	false = Disabled true = Enabled

Postal Code Filter			
PARAMETER	**TYPE**	**STRING**	**DATA**
Wildcard	3	POSTAL_WILDCARD_CHARx	Length: 1 character
Placeholder	3	POSTAL_PLACEHOLDERx	Length: 1 character
Min. Number Of Characters	0	POSTAL_MIN_NUM_CHARx	Range 1 to 3116
Max. Number of Characters	0	POSTAL_MAX_NUM_CHARx	Range 1 to 3116

Outputs

Outputs			
PARAMETER	**TYPE**	**STRING**	**DATA**
Active State	1	LOCOUT_ACTIVE_STATE_x	0 = Active Closed 1 = Active Open
Line Function	1	LOCOUT_MODE_x	0 = General Purpose 2 = External III – Always On 3 = External III – Strobed
Number of Events	0	LOCOUT_NUM_OF_EVENTS_x	Range 1 to 255
Line Type	1	LOCOUT_LINE_TYPE_x	0 = NPN 1 = PNP 2 = Push-Pull

LEDs

Green Spot			
PARAMETER	**TYPE**	**STRING**	**DATA**
Active State	1	OUT_ACTIVE_STATE_1	0 = Active Closed 1 = Active Open
Number of Events	0	OUT_NUM_OF_EVENTS_1	Range 1 to 255

Red Spot			
PARAMETER	**TYPE**	**STRING**	**DATA**
Active State	1	OUT_ACTIVE_STATE_2	0 = Active Closed 1 = Active Open
Number of Events	0	OUT_NUM_OF_EVENTS_2	Range 1 to 255

11 Web Monitor

Web Monitor is a remote monitoring tool provided to visualize the ABR reader in its run-time environment, even simultaneously from multiple devices at once.

Access Web Monitor from the Barcode Manager Task area or directly from your browser by inputting the IP address of the reader. The reader must available on the LAN.

Web Monitor is compatible with major browsers, smartphones, and tablets, which support HTML5 (see the following table).

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)	

The ABR Web Monitor interface provides the following features:

Figure 95. Information (Info) Page

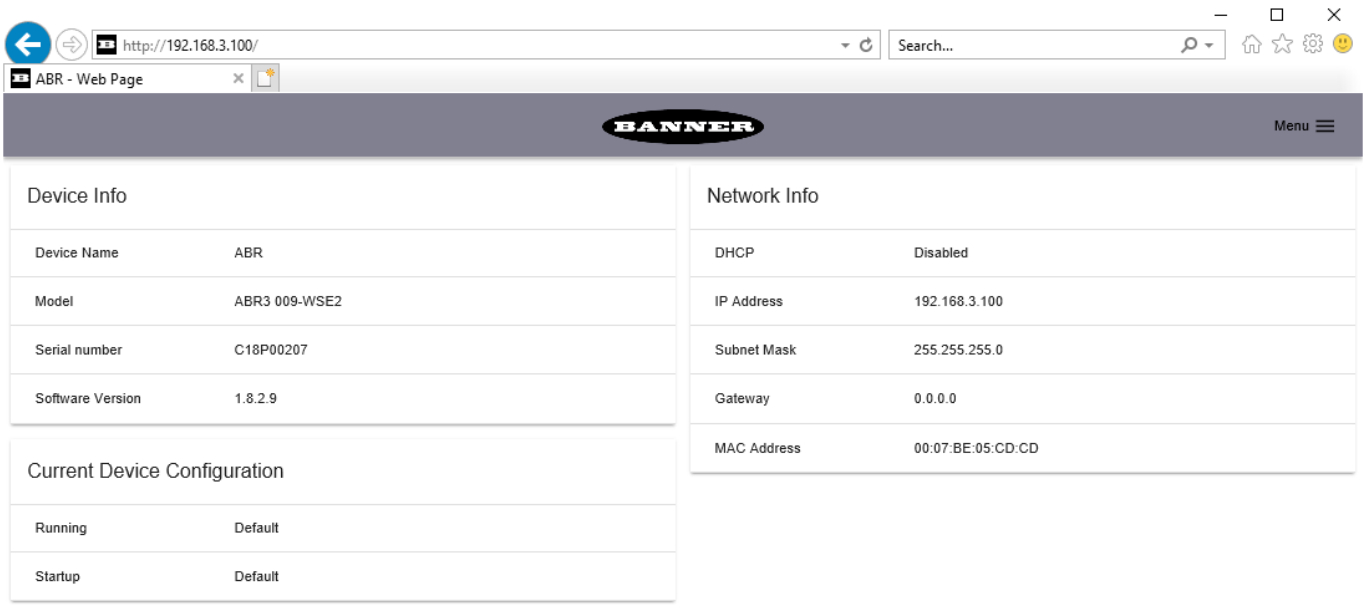
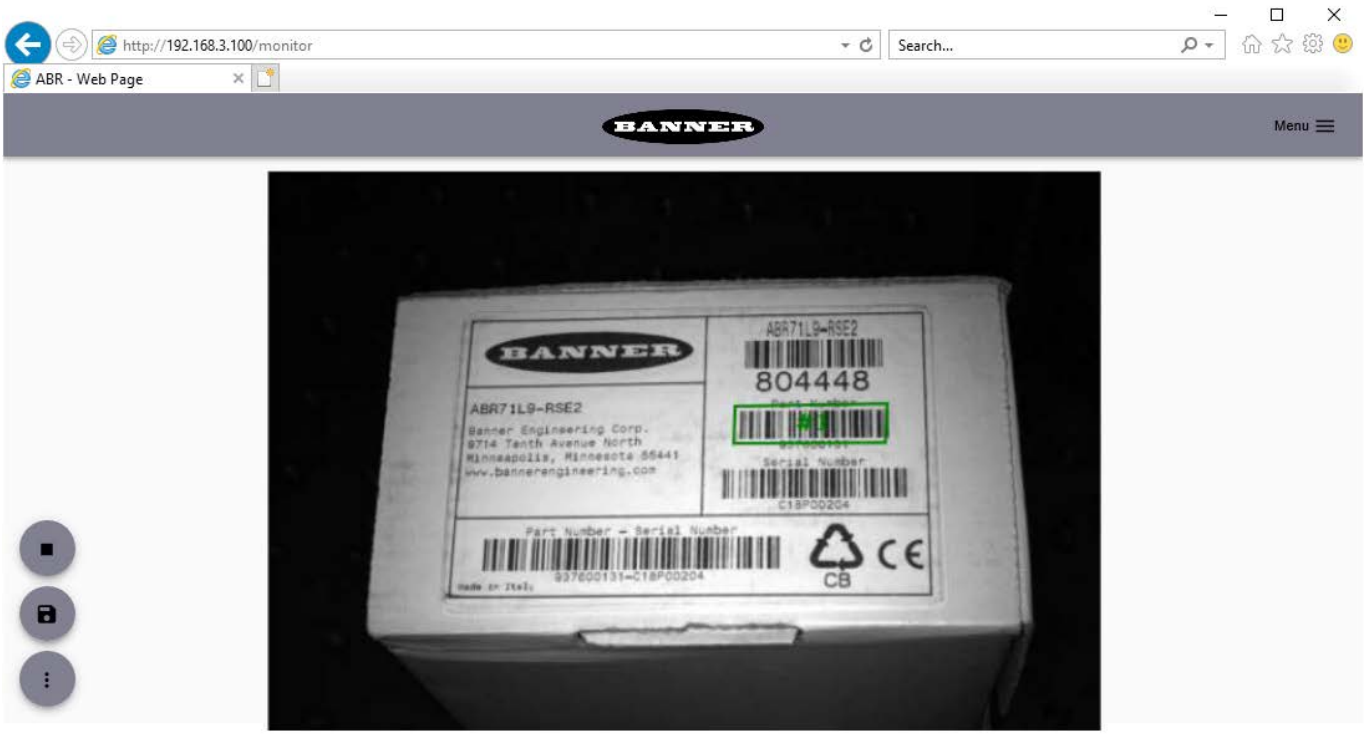


Figure 96. Monitoring Page



	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 (for example, image_Fri Sep 29 2017.jpg)
	The options button opens an options panel on the Monitoring window which allows setting or adding the following: • The quality of the image to monitor, (higher quality images reduce frame rate) • Show only positive results (good read), only negative results, or all results • Add Framerate reporting to the Monitoring window • Add 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.

Figure 97. Statistics Page

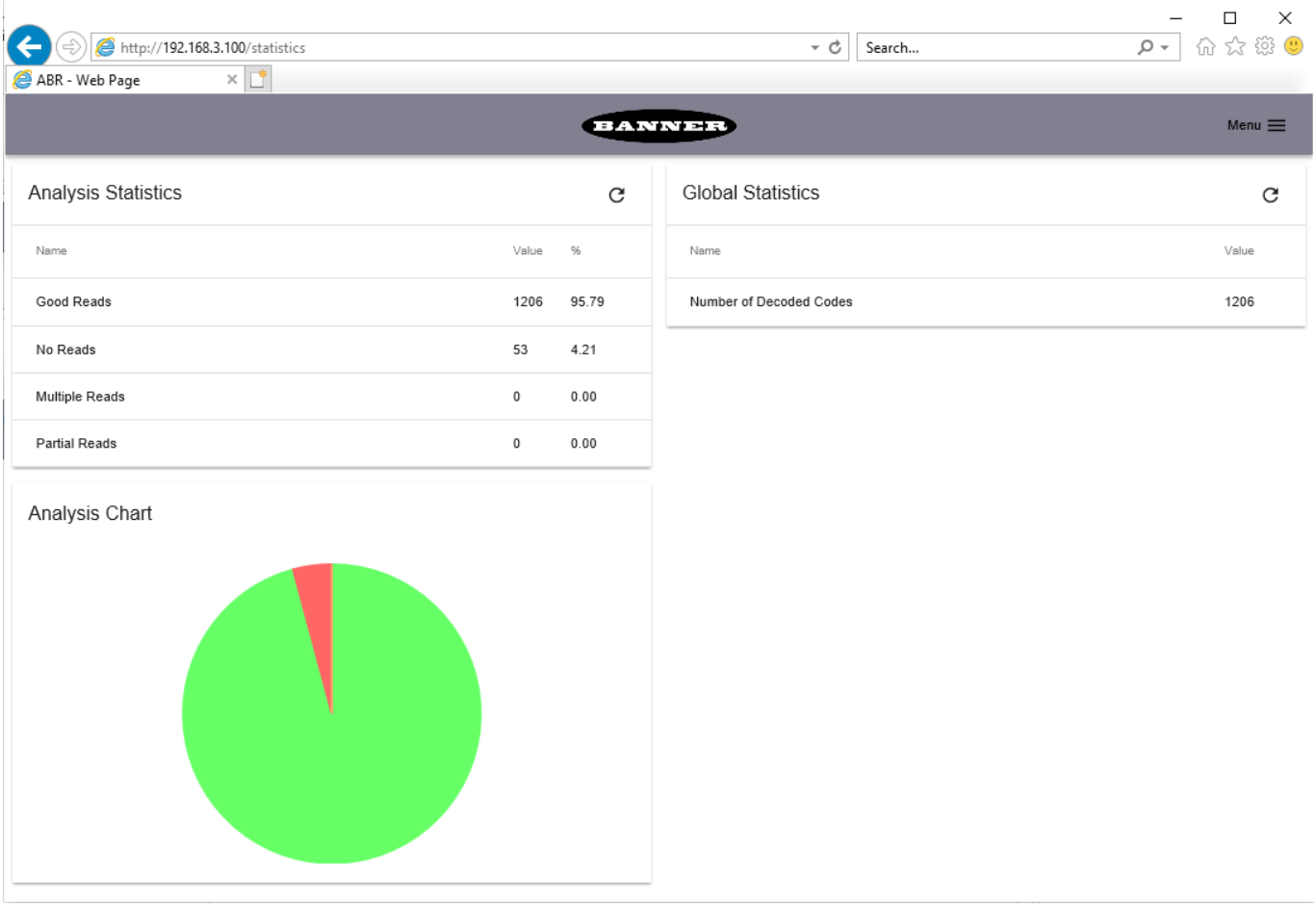
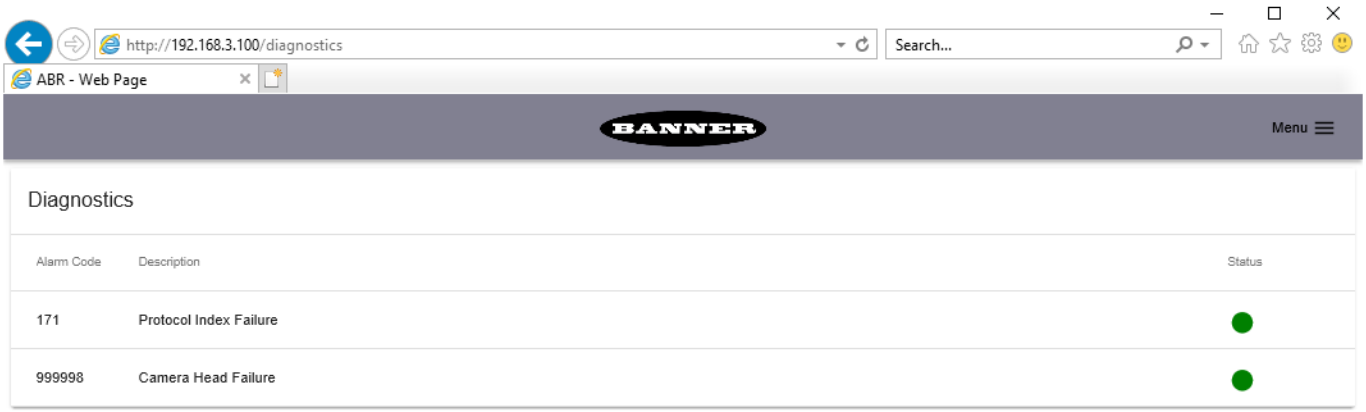


Figure 98. Diagnostics Page



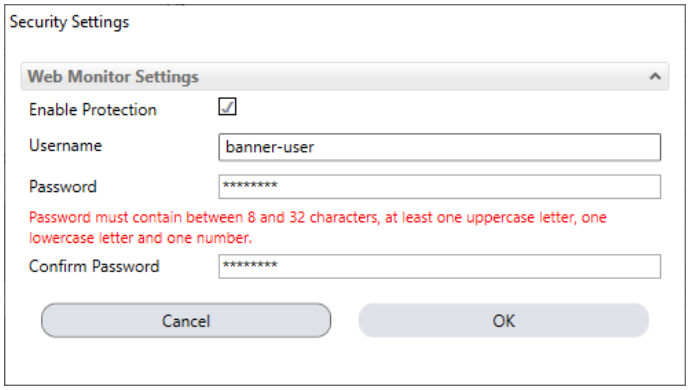
11.1 Access Reader Settings Though Web Monitor

The Web Monitor tool provides a page for **Code Grading Monitoring** and a **Configurator** to edit the ABR reader settings. A user name and password are required to access these features.

1. From Barcode Manager, go to **Device** > **Settings** > **Security Settings**.
2. Set the password under the **Web Monitor Settings** tab.

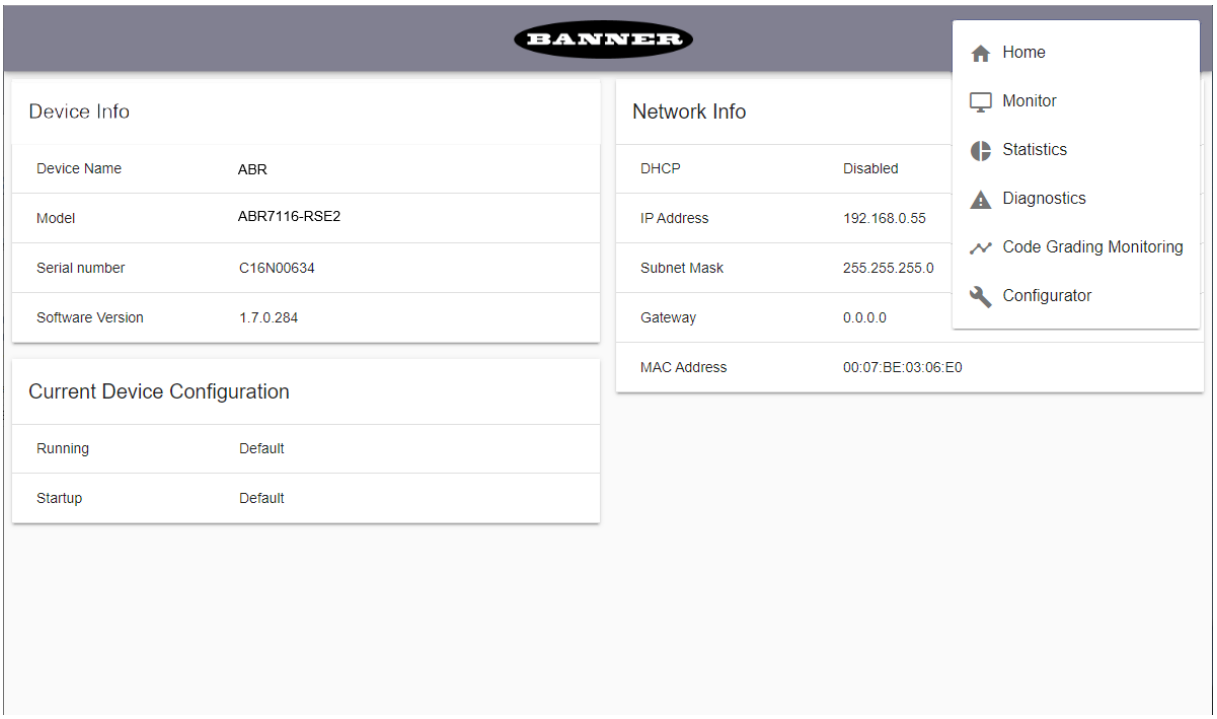
Note: A default user name is provided, however, for security reasons no password is provided.

*Figure 99. **Security Settings** Screen*



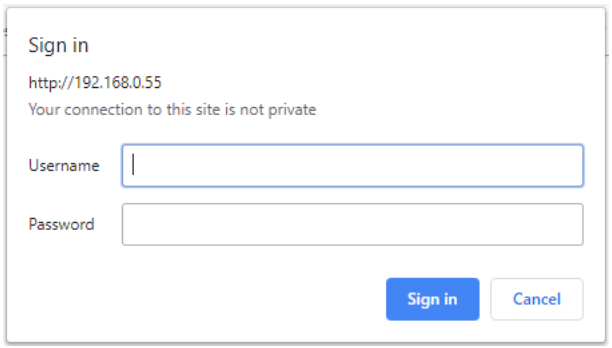
3. Click **Web Monitor** in the **Task Area**.
   The **Web Monitor** tool opens.
4. From the menu in the upper right corner of the **Web Monitor** tool, select **Code Grading Monitoring** or **Configurator**.

*Figure 100. **Web Monitor** Menu*



5. Enter your user name and password.

*Figure 101. **Sign in** Screen*



11.1.1 Code Grading Monitoring Page

On the **Code Grading Monitoring Page**, view live results formatted in graphs. Results can be viewed as **Grades** or **Values**
To view code grading live results, make sure that code grading is enabled. See Code Grading on p. 144.

Figure 102. Code Grading - Grades View

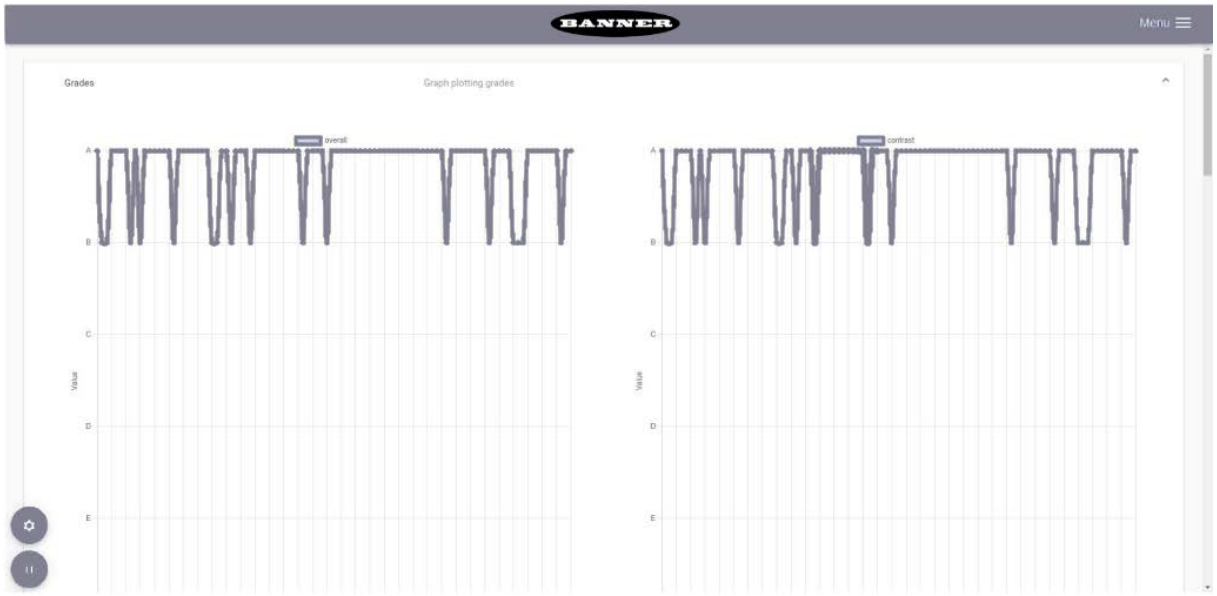
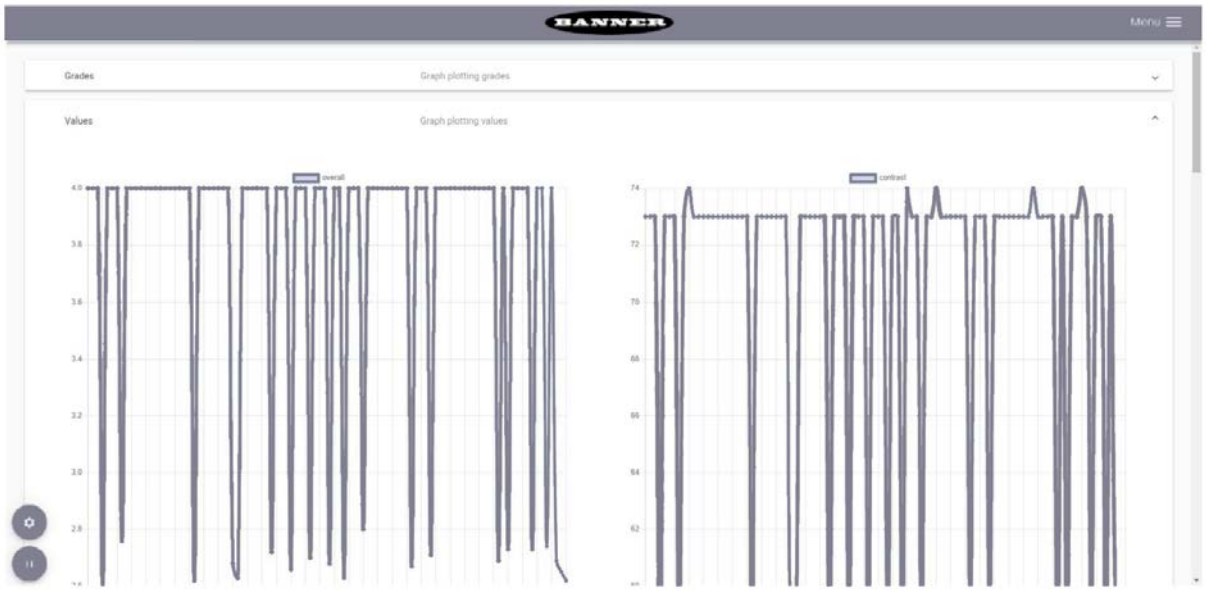


Figure 103. Code Grading - Values View



Each graph represents results for a code grading parameter. Click **Settings** to select

- The graphs to display
- The number of samples
- The refresh period (ms)

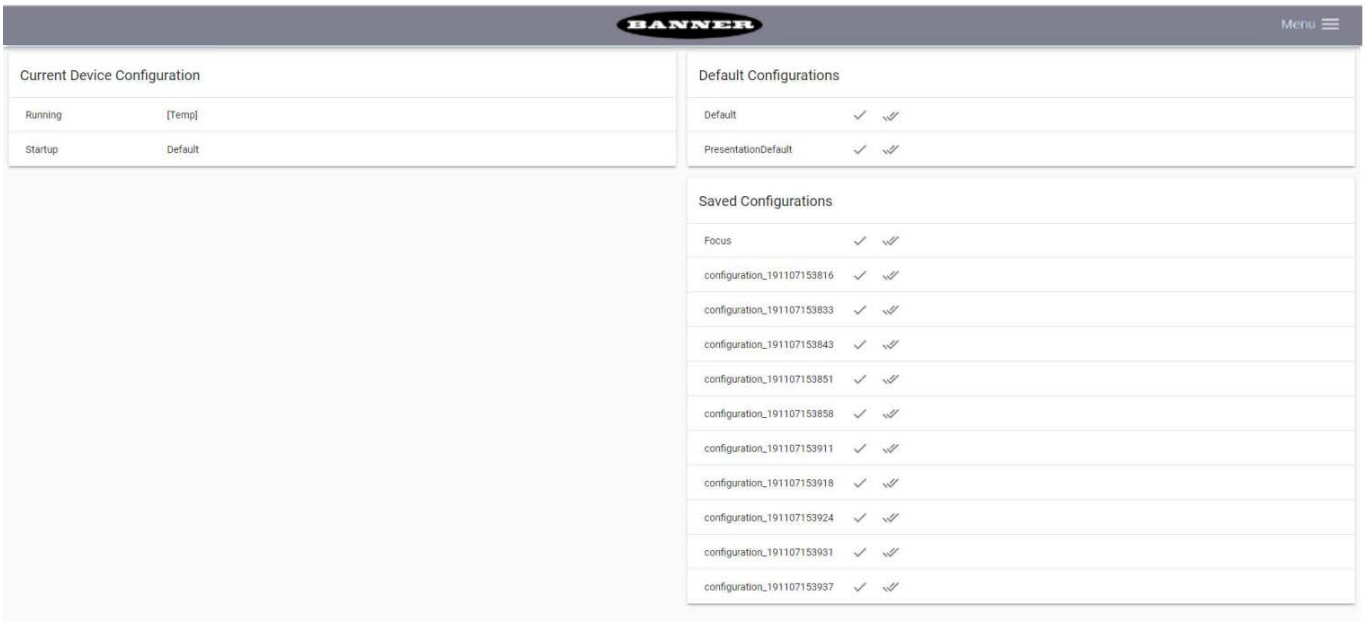
Figure 104. Graph Settings and Options



11.1.2 Configurator Page

From the Configurator page, view the configuration that is currently running on the device and run another one by selecting it from the saved configurations.

*Figure 105. **Configurator** Page*



- Click on the single checkmark to load the configuration
- Click on the double checkmark to load the configuration and set it as the startup configuration

The loaded configuration displays a yellow single checkmark. The startup configuration displays a yellow double checkmark.

12 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 Barcode Manager overhead and therefore reading results are much closer to real-time performance, but not as close as when using only the Web Monitor.

Monitor also provides diagnostic alarm feedback.

12.1 Accessing the Monitor

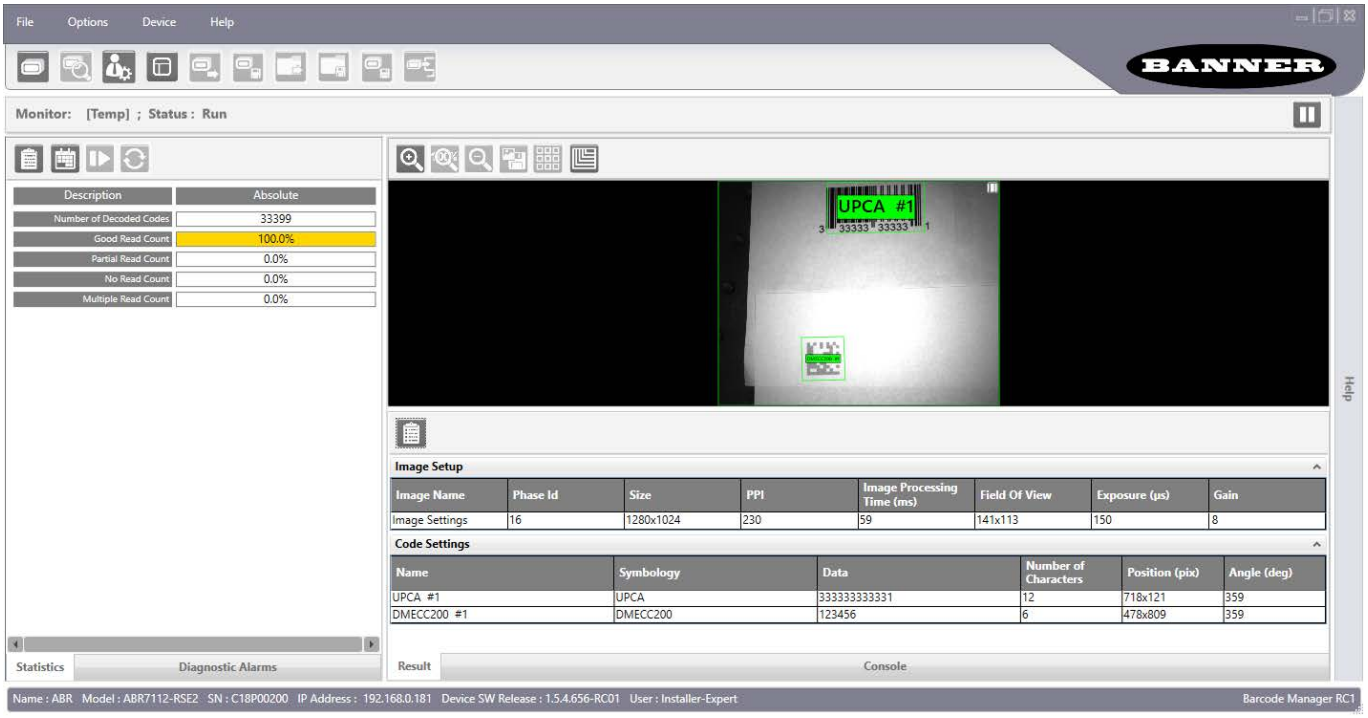
Access the **Monitor** page through **File** > **Monitor** or the **Switch to Configuration/Monitor** icon.

To exit the **Monitor** page, click **Pause**, then click **Switch to Configuration/Monitor**.

12.2 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 shows image acquisition. The Results area below the image gives feedback on decoding results.

Figure 106. Statistics and Results



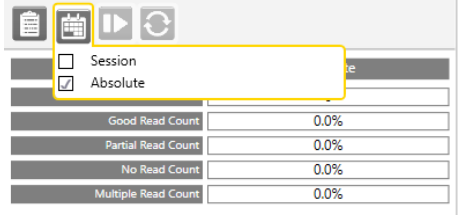
The Statistics page is collapsed to the left side of the Barcode Manager window and can be opened by clicking on the **Statistics** tab.

Resize the various windows by dragging the edges.

Toggle between table view and chart view statistics by clicking on the or icon.

Switch between Session and Absolute statistics using the icon.

Figure 107. Switch Between Session and Absolute Statistics



Session Statistics

Show only the fields selected that are represented as a percentage, rate, or average (that is, 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 Statistics Settings on p. 115). A session ends if the device is connected to the Barcode Manager configuration environment.

Absolute Statistics

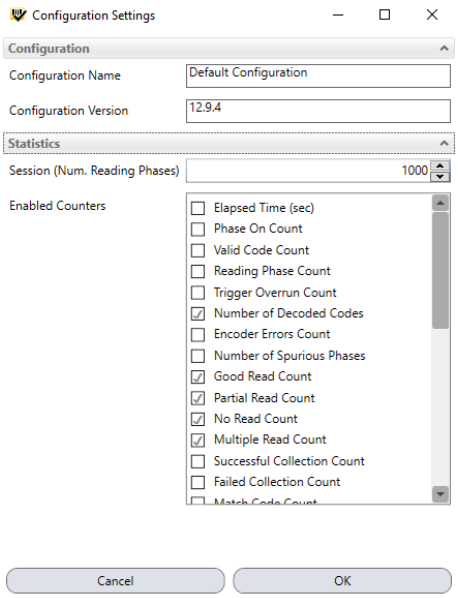
Show all the fields selected in the **Device menu** > **Settings** > **Configuration Settings** > **Statistics** > **Enabled Counters** list from the last device power on or the last time the Statistics were reset.

12.2.1 Statistics Settings

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

Choose which fields to visualize in the **Monitor Statistics** page.

Figure 108. Configuration Settings - Statistics Screen



To reset all the statistics (both Session and Absolute), click the **Reset Statistics** icon or go to **Device menu** > **Settings** > **Reset Statistics**.

12.3 Monitoring Diagnostic Alarms

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

Figure 109. Diagnostic Alarms



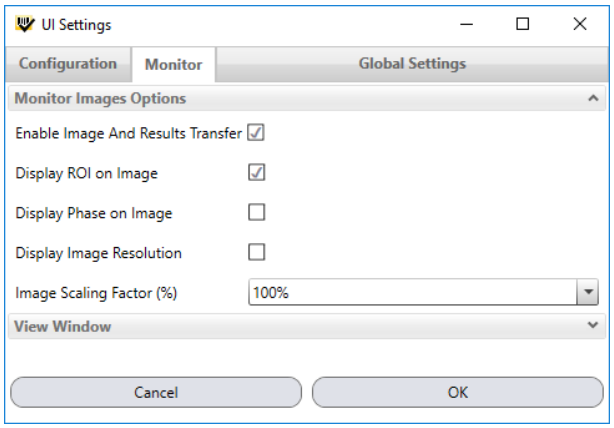
Alarm Code	Description	Meaning	Applicable Models
1	Slave No Reply	No response to master from slave (shown in slave column).	7000
64	Slave Address Duplication	Two or more slaves have the same address.	7000
171	Protocol Index Failure	The expected Protocol Index is not received.	3000, 7000
185	Backup Memory Communication Failure	There is a communication error between the reader and the external memory (TCNM-ACMK-100 module inside the TCNM-ACBB1 connection box).	7000
187	Wrong Rotary Switch Selection Failure	One or more of the selected TCNM-ACMK-100 Rotary Switch settings inside the TCNM-ACBB1 doesn't match the reader configuration memory.	7000
999998	Camera Head Failure	There is a communication error between the reader and its internal camera module.	3000, 7000

12.4 Monitor Settings

12.4.1 Monitor Images Options

Several options can be set for the Monitor window. Go to **Options** > **UI Settings** menu on the **Monitor** tab.

Figure 110. UI Settings - Monitor Images Options



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.

Display Phase on Image

Shows the Phase identification number in white letters in the upper left-hand side of the image.

Display Image Resolution

Shows the Image resolution in white letters in the upper left-hand side of the image.

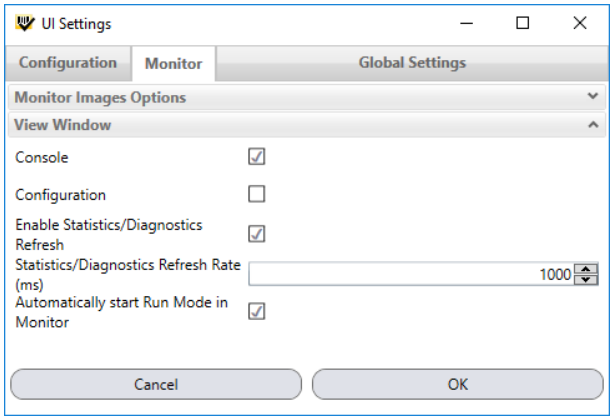
Image Scaling Factor (%)

Allows downsizing the image subsampling to increase the visualization performance.

12.4.2 View Window

Select several options to view in the Monitor window.

Figure 111. UI Settings - View 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.**

Enable Statistics/Diagnostics Refresh

Shows the Statistics and Diagnostic Alarms panels.

Statistics/Diagnostics Refresh Rate (ms)

Set the refresh rate, in milliseconds, for statics and diagnostics.

Automatically start Run Mode in Monitor

When launching Monitor the window opens with the device in Run Mode. This is the default setting.

13 Image Saving

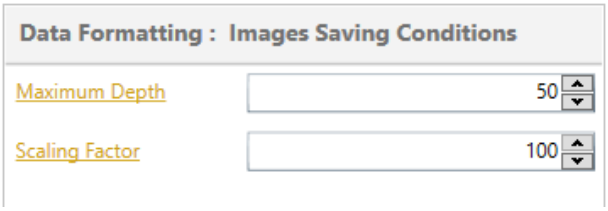
13.1 Device Image Buffer

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

Important: The internal device memory has a circular buffer to avoid overrunning the available memory. To save images correctly to the internal buffer, calculate the number of images the buffer can hold based on the image resolution for your device model.

More images can be stored by scaling down the image size. Go to **Communications** > **Images Saving Conditions**.

Figure 112. Images Saving Conditions



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 Important Note.

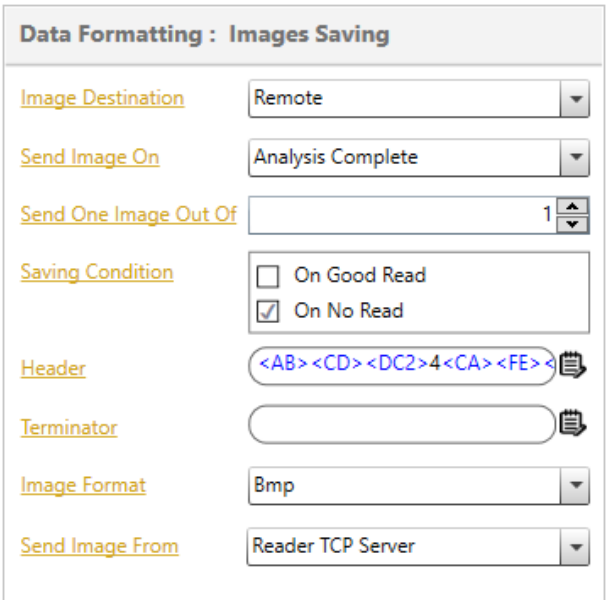
Scaling Factor

Allows reducing the size of images to be saved so that a greater number of images can be saved.

13.1.1 Image Saving Using Reader TCP Server/Client

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

Figure 113. Images Saving Condition



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

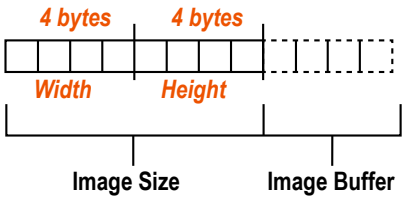
According to the selected Saving Condition parameter, after data collection analysis is complete, images can be sent at run time directly from the Device Image Buffer to a remote or local PC.

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.

Figure 114. Image Saving Format

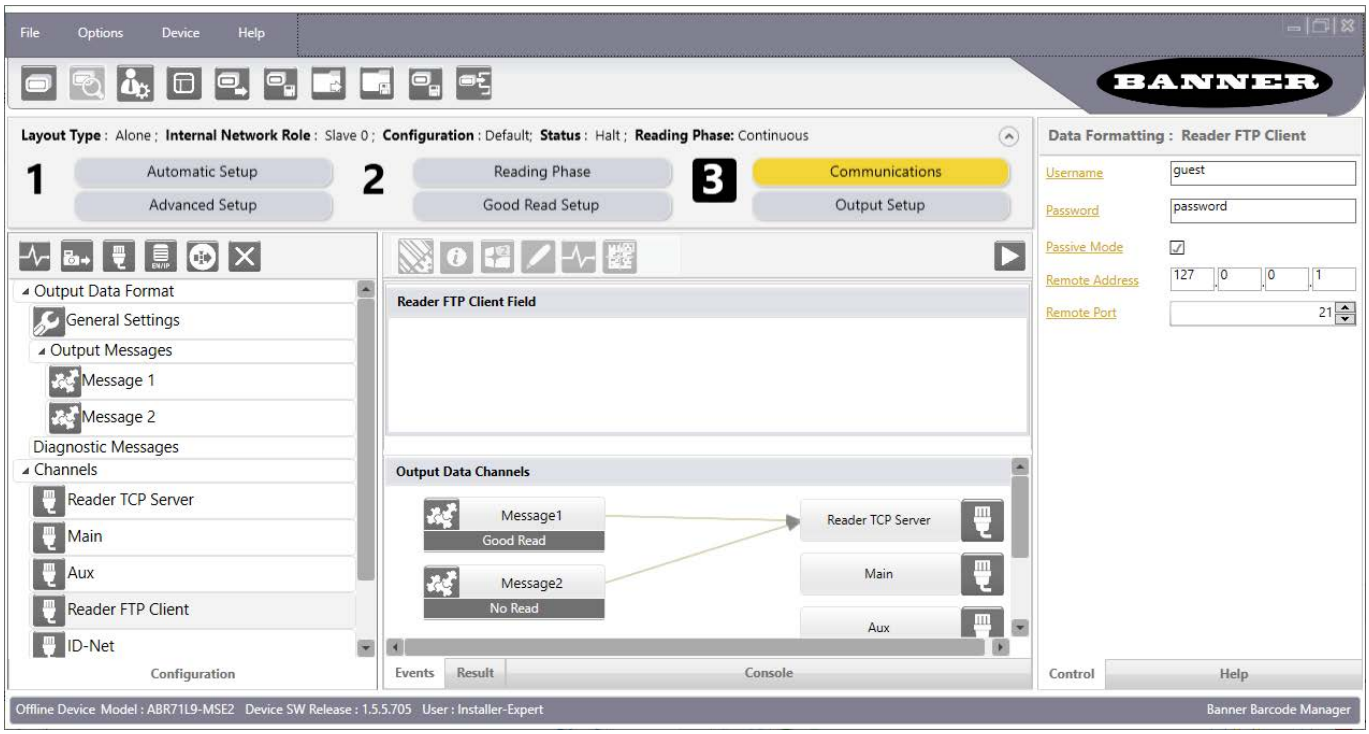


13.1.2 Image Saving Using Reader FTP Client

Images can be sent at run time directly from the device image buffer to an external FTP Server through a Reader FTP Client channel.

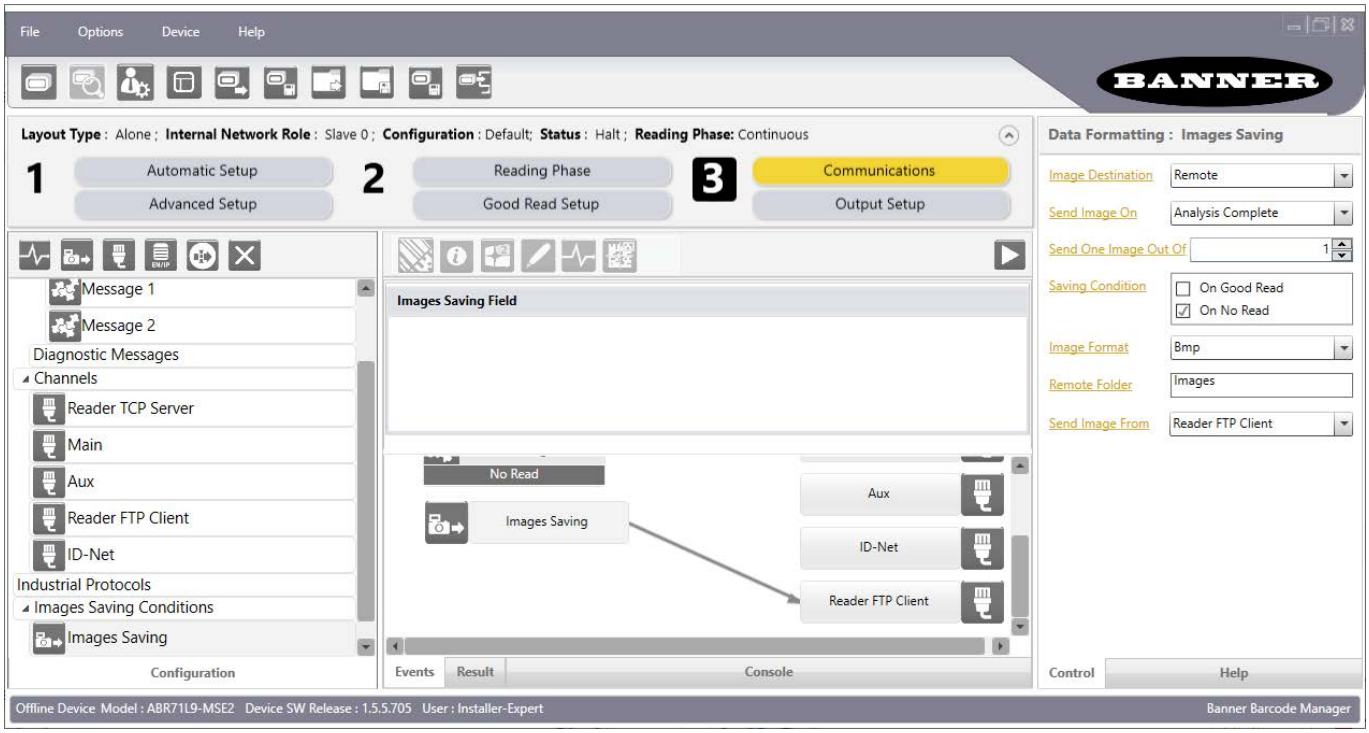
1. Select the **Add New Channel** icon to add the Reader FTP Client. It must be added as a new communication channel because it is disabled by default.

Figure 115. Communications: Reader FTP Client



2. Verify the following:
   - The Server Address matches the external FTP Server PC
   - The Username and Password match the ones required by the external FTP Server PC
3. Select **Images Saving**.
4. Set the **Send Image From** parameter to **Reader FTP Client**.
5. Input the name of the Remote Folder on the FTP Server to which the image files must be saved.

Figure 116. Images Saving



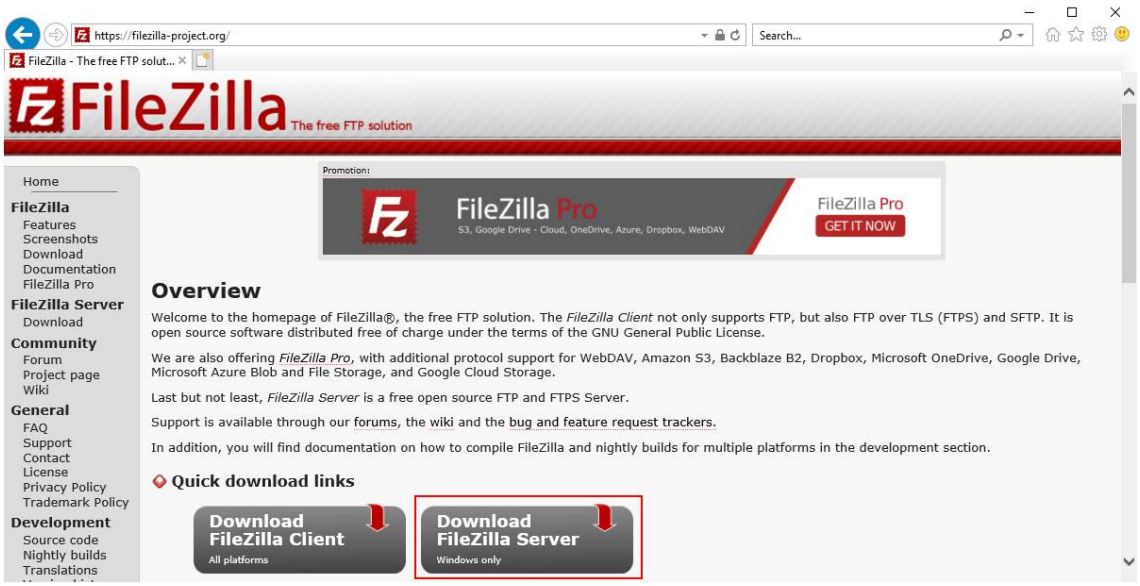
13.1.3 Image Export to FileZilla FTP Server

Use these instructions to export images from an Ethernet-equipped ABR to the FileZilla FTP server. Use these instructions as a basis for using other FTP servers. These example instructions use a Windows® 10 [14] system.

1. Download the free FileZilla FTP server for Windows.

Note: FileZilla has both an FTP server and an FTP client. Use the FTP server.

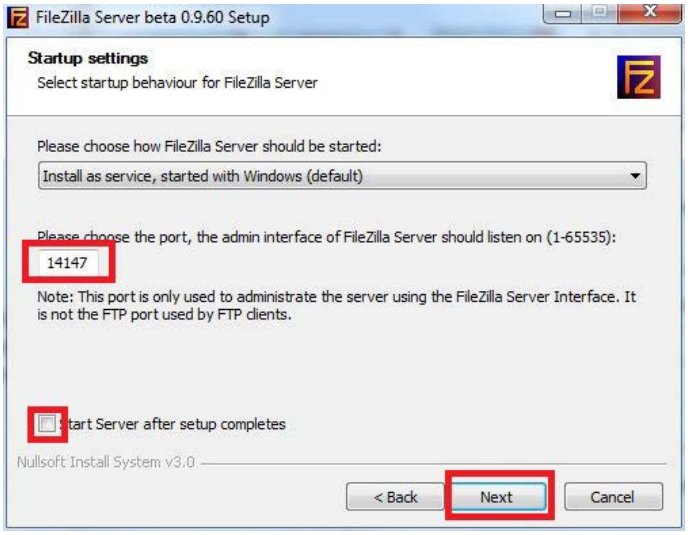
a) Navigate to the FileZilla website: filezilla-project.org.
b) Download the FileZilla Server executable file.



2. Install FileZilla.

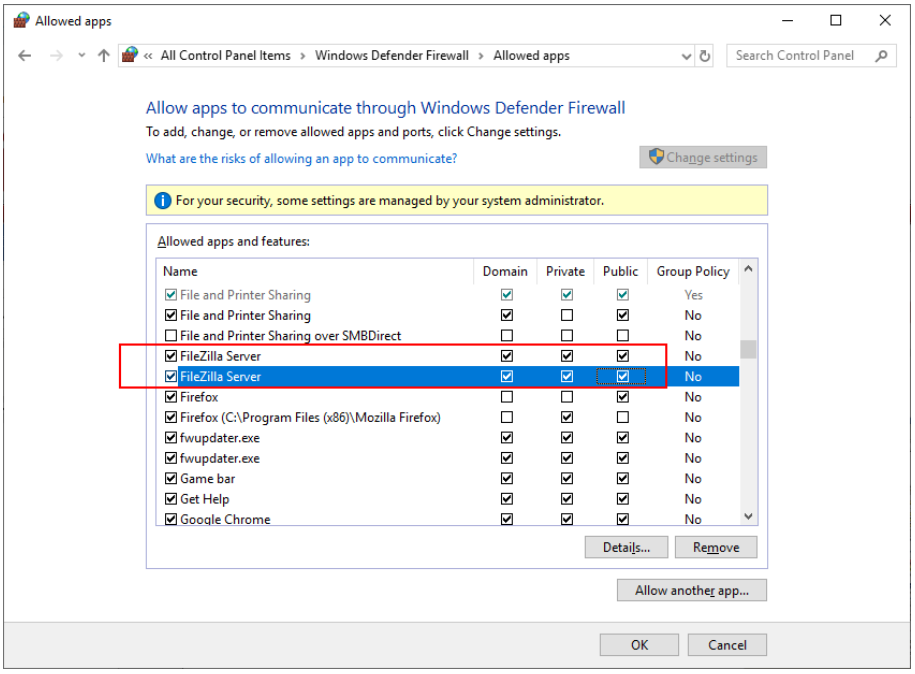
[14] Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries.

a) Run the executable file to start the installation process.
b) Click **I Agree** on the **License Agreement** screen.
c) On the **Choose Components** screen, select the components to install, and click **Next**.
For this example, use the standard default options.
d) On the **Choose Install Location** screen, select the destination folder, and click **Next**.
For this example, use the default location.
e) During installation of the FileZilla FTP Server, make note of the Admin port setting provided on the **Startup settings** screen.
You will need this number to log in later.

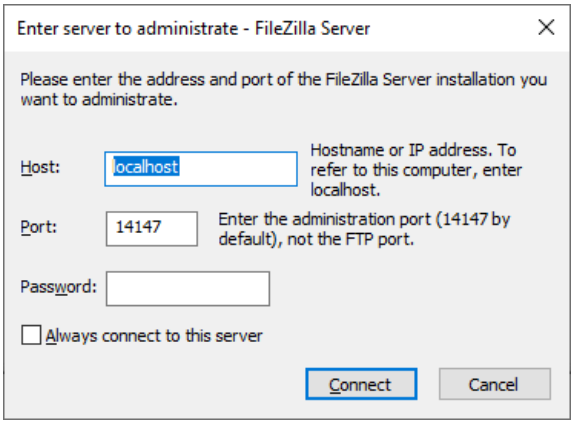


f) Clear the **Start Server after setup completes** checkbox.
g) Click **Next**.
h) Clear the **Start Interface after setup completes** checkbox.
i) Click **Install**.
The FileZilla Server installs.
j) When the installation is complete, click **Close**.

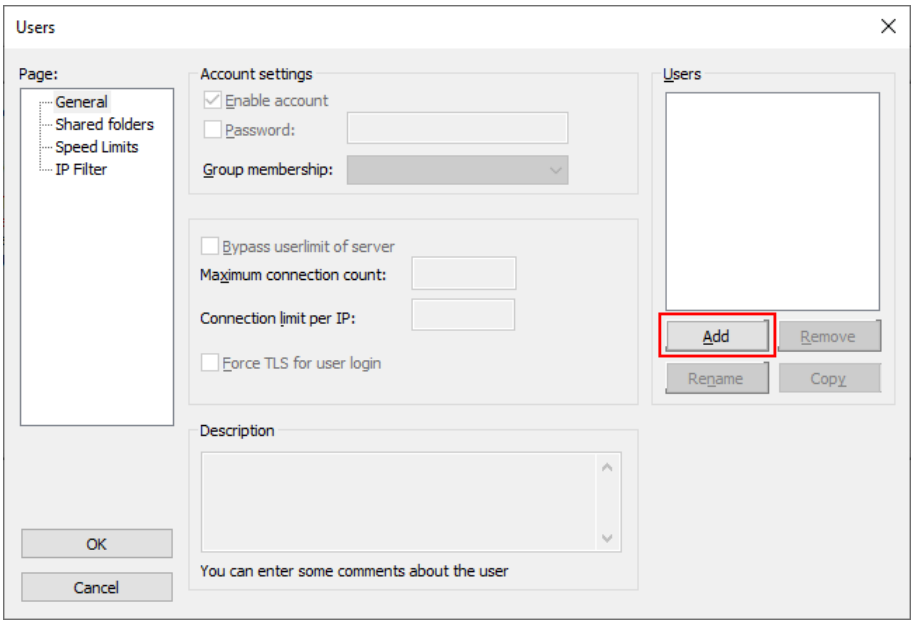
3. Make sure that the Windows Firewall does not interfere with the FileZilla software.
a) Click **Start**, type `control panel`, and click **Control Panel**.
b) From the **View By** list, select **Small Icons**.
c) Click **Windows Defender Firewall**.
d) Click **Allow an app or feature through Windows Defender Firewall** on the left.
e) Click **Change Settings** on the right.
f) If the programs are listed, select the **FileZilla Server Interface** and **FileZilla_Server** checkboxes in the **Allow programs and features** list.
If you do not see the programs, go to step 4.
g) Select the **Domain**, **Private**, and **Public** checkboxes, and go to step 5.



4. If you do not see **FileZilla Server Interface** and **FileZilla_Server** on the list initially, follow these steps:
   a) Click **Allow another app...**
   b) From the **Add an app** screen, click **Browse...**
   c) Navigate to the location of the FileZilla server.
      The default location is `C:\Program Files (x86)\FileZilla Server`.
   d) Choose the executable file `FileZilla Server.exe` and click **Add**.
      The program is added to the **Allowed programs and features** list.
   e) Select the **Domain**, **Private**, and **Public** checkboxes.
   f) Repeat steps b. through e. for `FileZilla Server Interface.exe`.
5. Set up the FileZilla FTP server.
   a) Click **Start**, then click **FileZilla Server**.
   b) Right-click on **FileZilla Server Interface**, click **More**, and select **Run as administrator**.
   c) Click **Connect** to administrate the FileZilla FTP Server.



   d) Ignore the red text in the window.
   e) Click **Edit** at the top of the program, then click **Users**.
   f) Click **Add** to create a new user.



g) Enter the user account name, then click **OK**.
For this example, the user name is: `Test`.

h) On the **General** tab, select the **Enable account** checkbox.

i) Select the **Password** checkbox and enter a desired password for the user account.
For this example, the password used is: `password`.

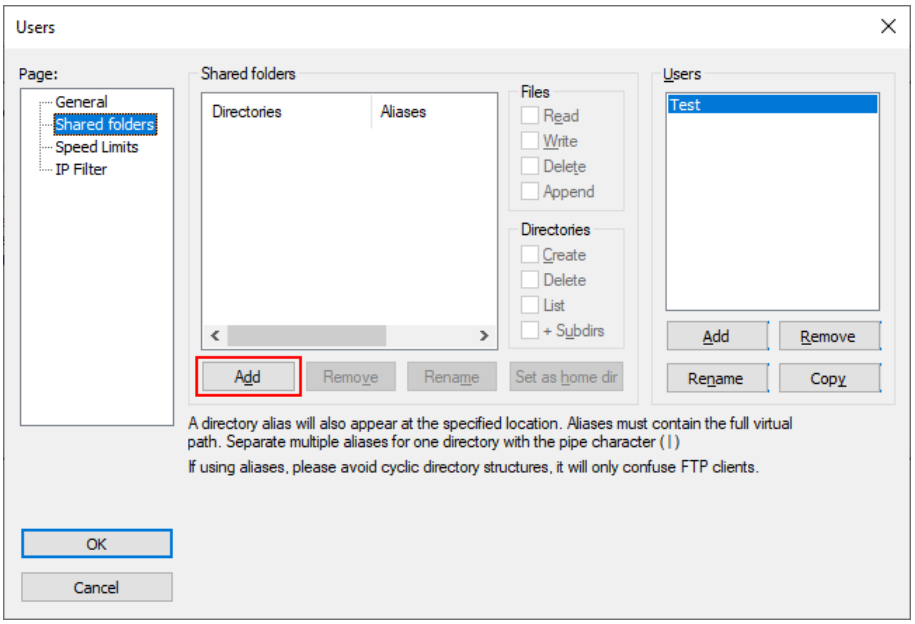
The ABR uses this User Name and Password to log into this FTP server.

j) On the message `You need to share at least one directory and set it as home directory`, click **OK**.

6. Share a directory with the FTP user account.

a) Create a new folder.
For this example, add a new folder called `FTP Server` to the PC C drive.

b) From FileZilla, click **Shared Folders**, then click **Add**.



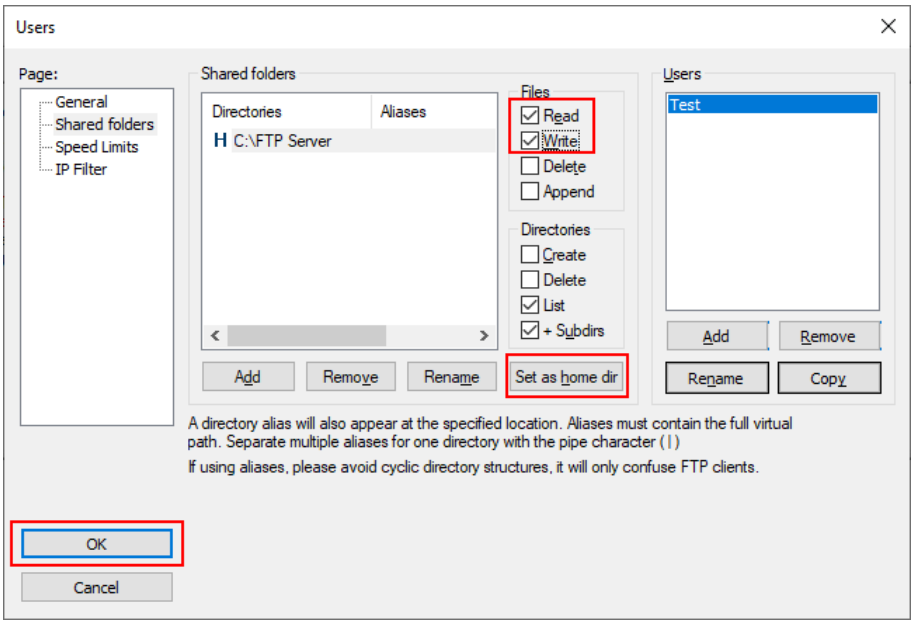
c) Select the folder to send the exported images to.

These are the images that are exported from the ABR. This folder will be linked to the user account and password created in step 5.

d) Make sure that the user account can write to the shared folder.

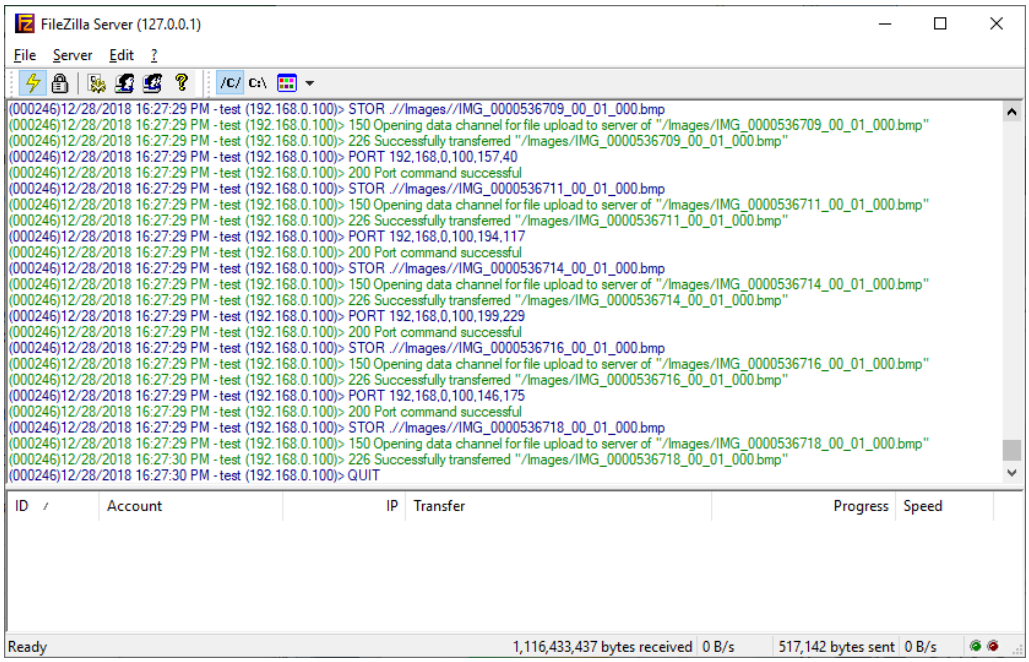
e) Select the folder, enable **Write**, and click **Set as home dir**.

f) Click **OK** to finish setting up the FTP Server.

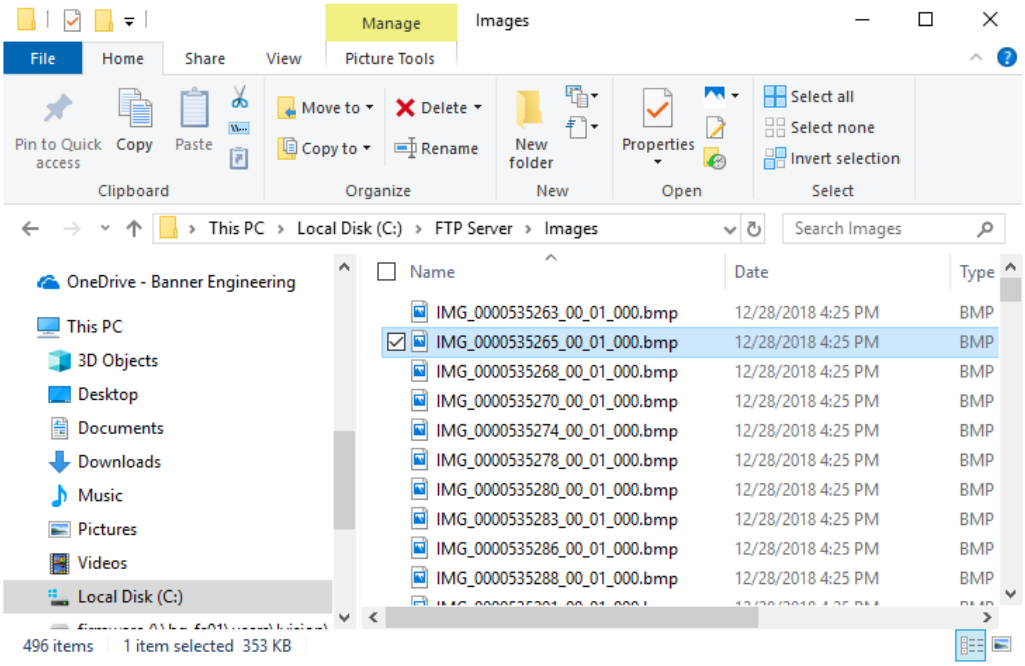


7. In Barcode Manager, save your settings, if desired, and then click  to disconnect and put the reader in run mode.

The FTP Server receives a file every time the reader gets a No Read result. The **FileZilla FTP Server Admin** window looks like the following as the image files are received:



The shared folder accumulates .bmp image files as the server runs.



To store Good Read images in a separate folder, create a second Image Saving Condition, select **On Good Read** instead of **On No Read**, and change the Remote Folder to another folder name.

13.1.4 Image Saving On Demand Using Send Images HPM Command

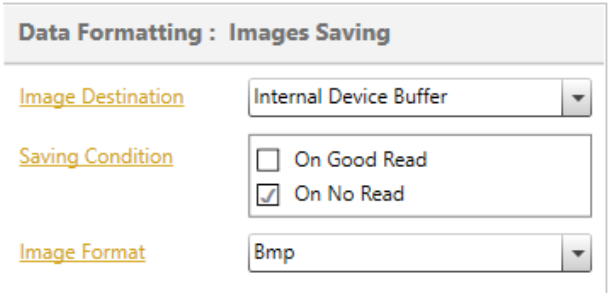
Images can also be sent to a remote destination through a Reader communication channel (typically to an FTP Server through a Reader FTP Client channel), upon receiving a Send Images command from the Host. For details on this command, see Host Mode Programming on p. 79.

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

13.1.5 Image Saving Using Internal Buffer

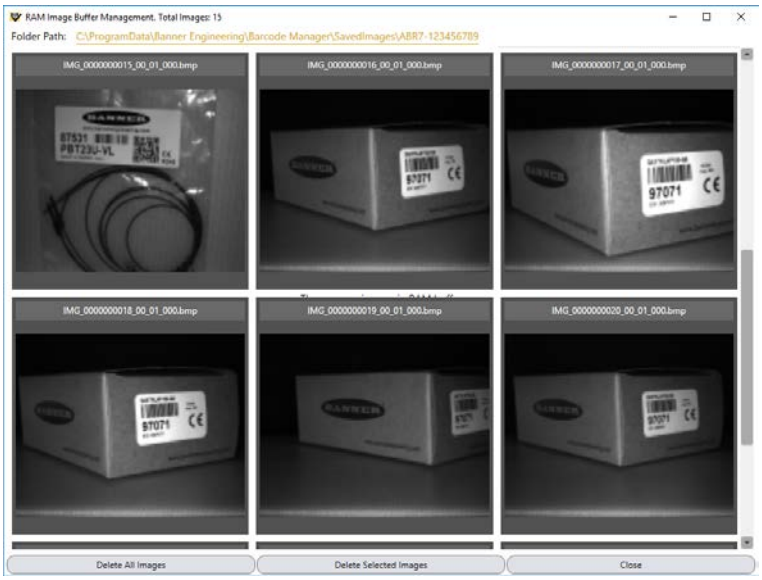
Images can be saved to the Device Image Buffer and be downloaded to a PC when the device is offline.

Figure 117. Device Image Buffer



To download the saved images, the device must be in offline mode. Go to **Device menu** > **RAM Image Buffer Settings** to access the **Image Buffer Management** window.

Figure 118. RAM Image Buffer Management



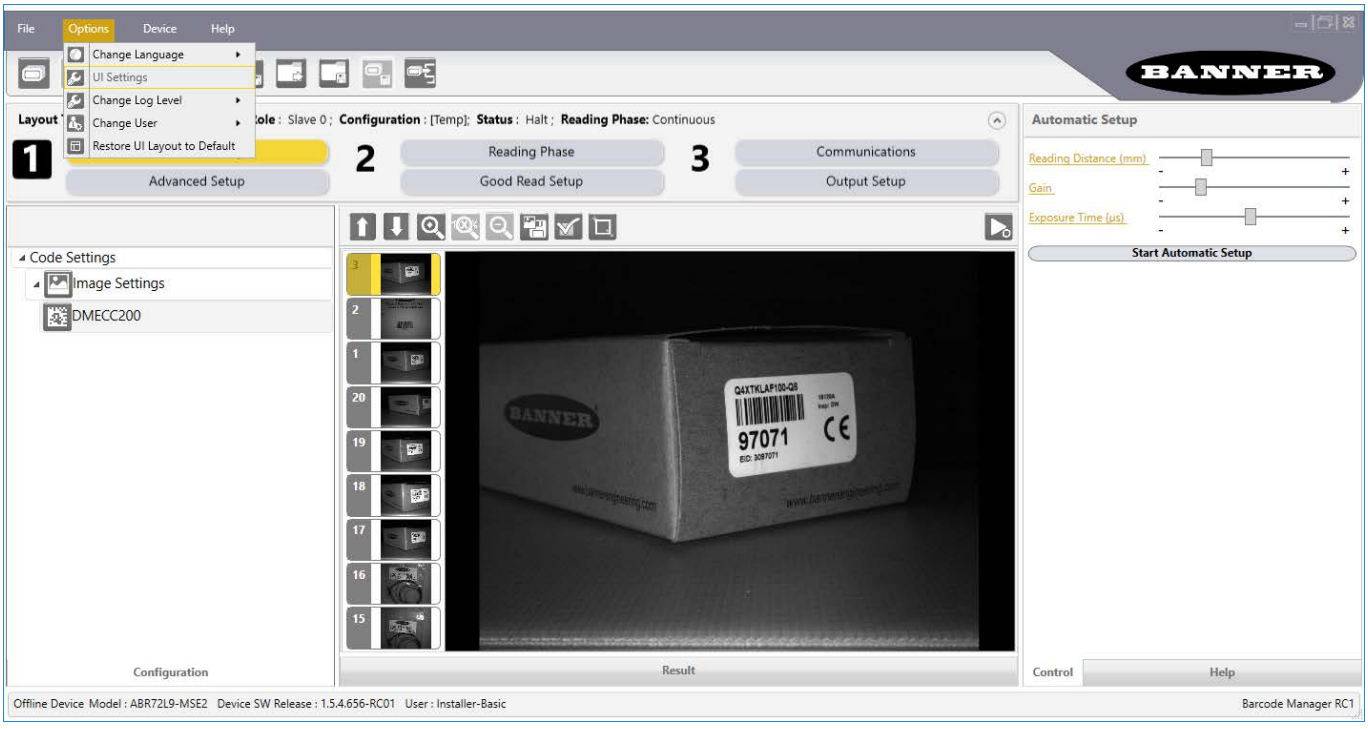
Upon opening this window, the images in the Device Image Buffer (RAM) are transferred to the Barcode Manager Device Image Buffer Management window, as well as the PC hard drive folder listed at the top of the window, and the images are deleted from the device RAM.

Deleting these images using the **Delete all Images** or **Delete Selected Images** buttons deletes the images from both this window, as well as from the PC hard drive. Any images remaining when clicking **Close** remain on the PC hard drive.

13.2 UI Image Buffer

The UI Image Buffer manages the images captured through the Barcode Manager Play and Capture features. Go to **Options** > **UI Settings** > **Buffer Size** to change the number of images available in the filmstrip to the left of the full image.

Figure 119. UI Settings - Image Buffer

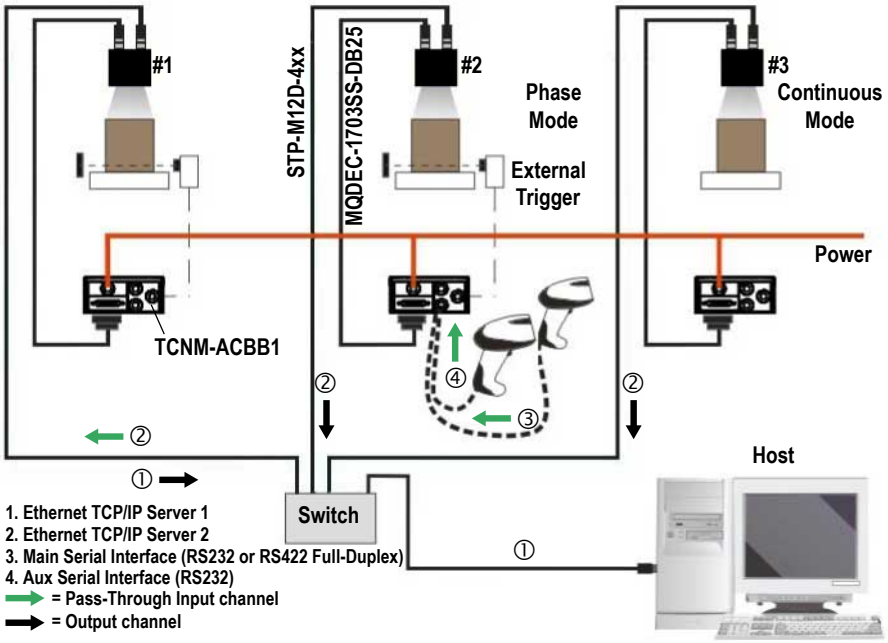


14 Multi Device Configuration Options

14.1 Pass-Through 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 120. Example Pass-Through Layout



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

Figure 121. Reader 1 Configuration Settings



Figure 122. Reader 2 Configuration Settings

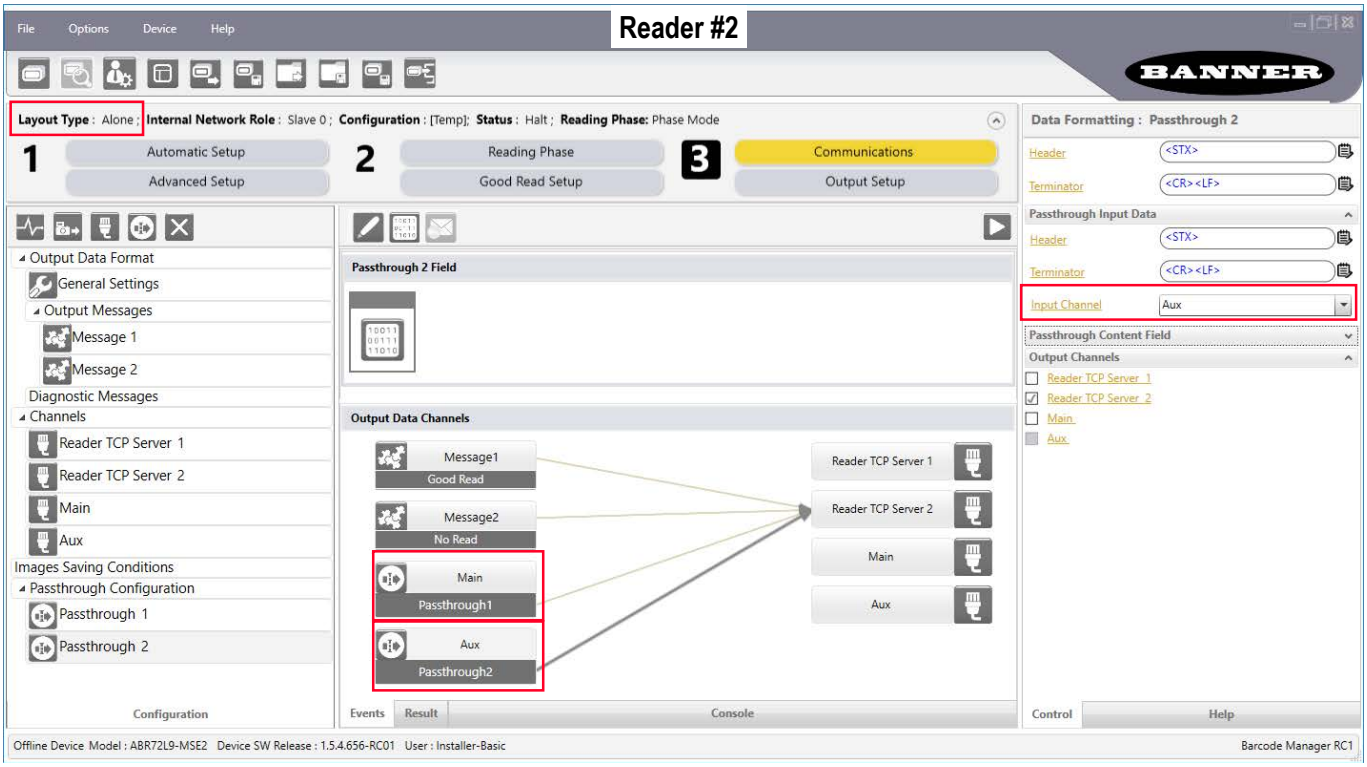


Figure 123. Reader 3 Configuration Settings



14.2 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.

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

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:

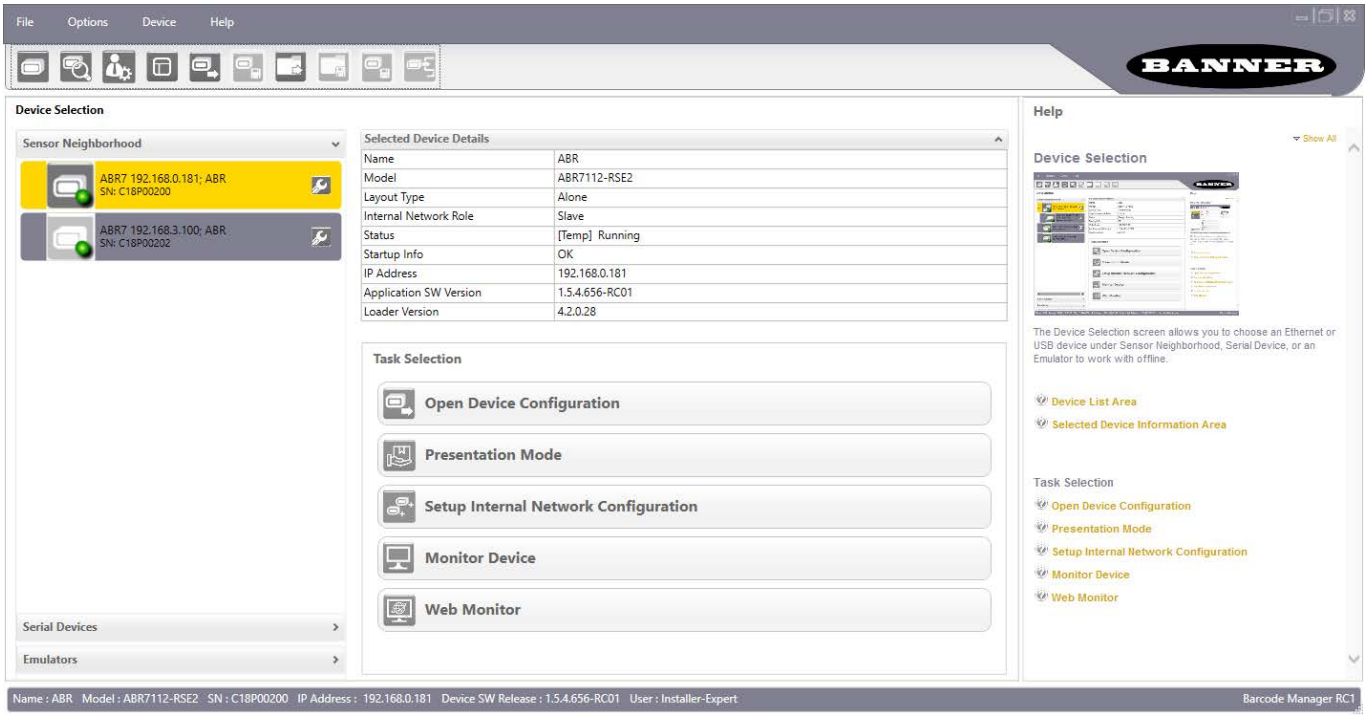
1. Mount all the readers (mechanical and electrical installation) with factory default settings (Layout Type = **Alone**, Internal Network Role = **Slave**).
2. Connect to the designated Master device in Barcode Manager and open the **Setup Internal Network Configuration**. You will be prompted to change the device to Master.
3. Click **OK**. The Slave units will automatically be recognized.
4. Depending on the application, select **Multidata** or **Synchronized Phase Mode**.

14.2.1 Master Configuration

Start with the desired device to assign as ID-NET Master (current default setting is Slave).

1. From the **Task** area, click **Setup Internal Network Configuration**.

Figure 124. Task Selection Area



   You will be advised that the device role will be changed to Master.
2. Click **OK**.
   The Net Autoset feature automatically starts to find Slave devices connected to the ID-NET network of the Master.
   When finished, all of the Slaves should have been correctly recognized. If not, verify all device connections and power, and then click **Start Net Autoset** to repeat the operation.

14.2.2 Internal Network Icons

	Open Device Configuration	Go directly to the Master device configuration without setting one of the default network configurations.

	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 Slave	Delete the Slave device from the list.

Note: The internal network can be pre-configured by right-clicking (and waiting) **Add a Device** 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 or Synchronized Phase Mode.

Figure 125. Internal Network Task Selection



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

14.2.3 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.

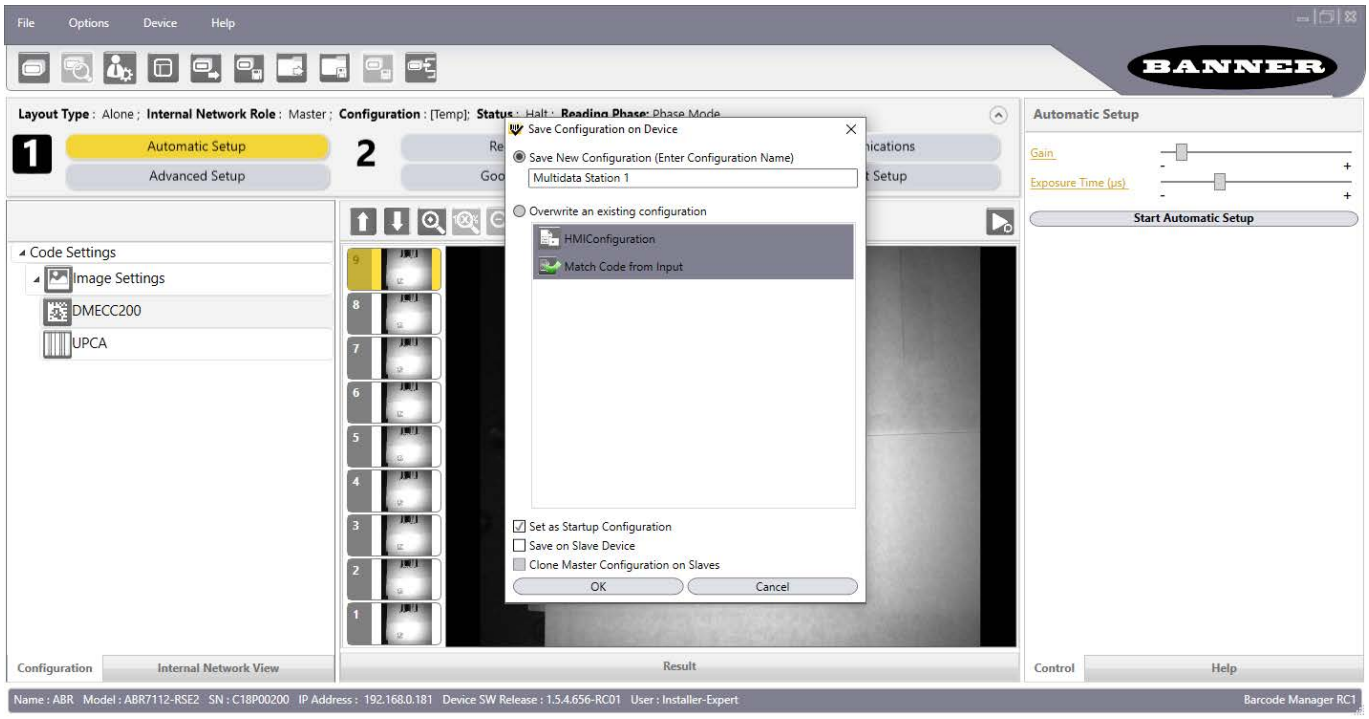
A pre-configured job is loaded with the correct pass-through settings for both the Master and Slaves when the Default Multidata Configuration is selected from the Internal Network Setting feature.

1. Complete the configuration of all the application parameters (including **Image Settings**) and save them to the Master with an application specific name.

   Optionally, selecting the **Save on Slave Device** checkbox 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 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 selected**.

   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.

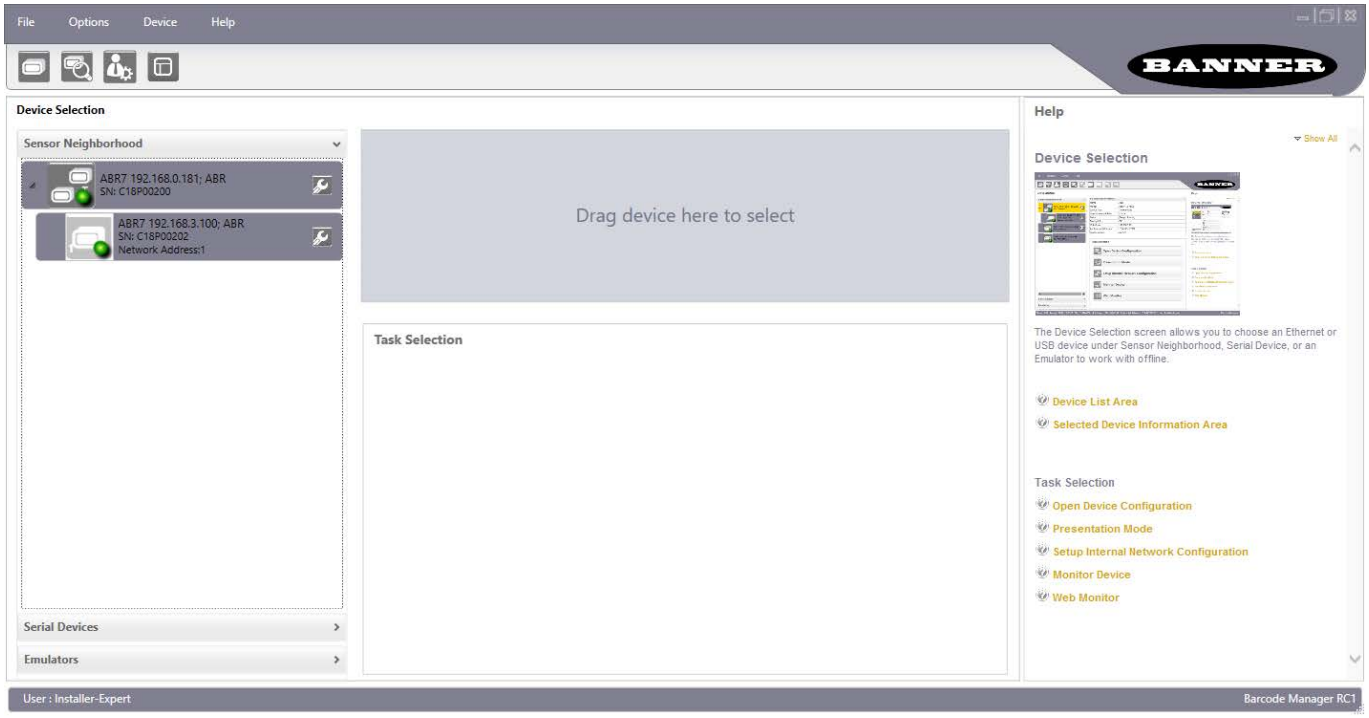
Figure 126. Saving Multidata Configuration to the Master



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

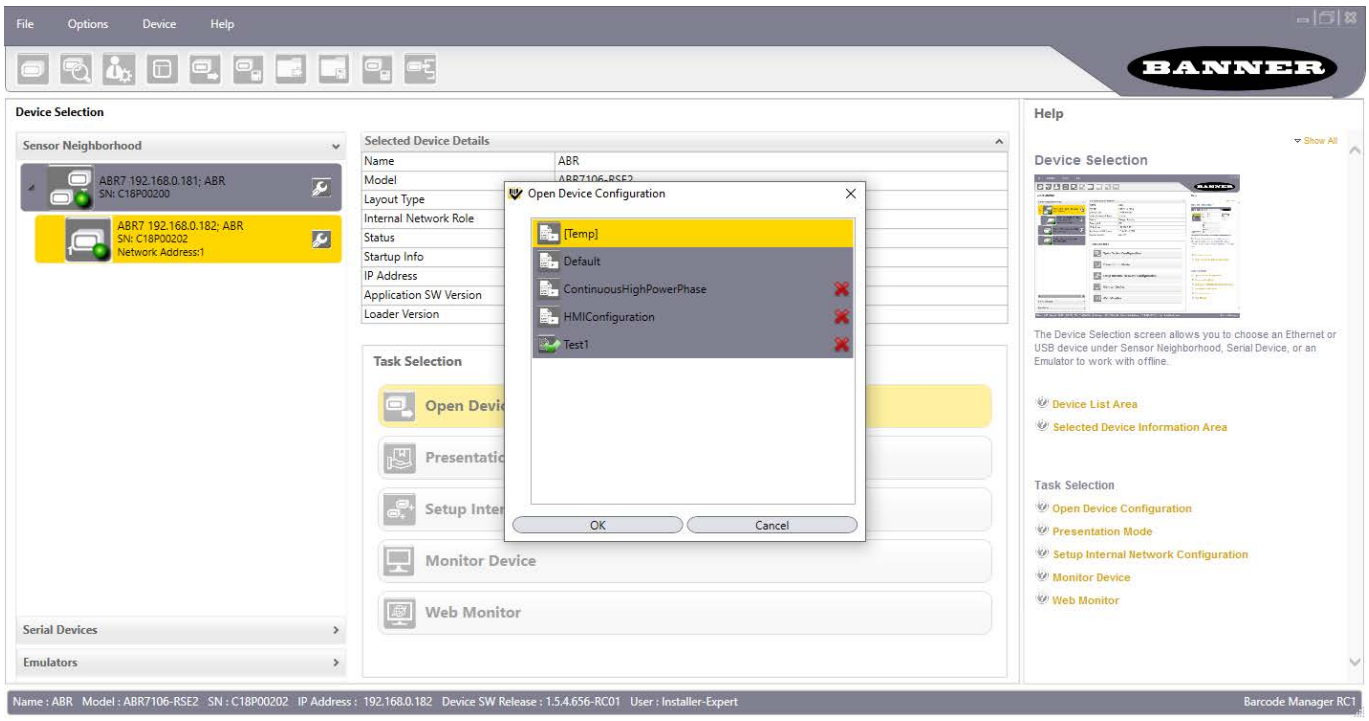
Note: If necessary, Slave device photometric (Image Settings) parameters must be configured separately through Barcode Manager. This is preferably done through each device's Ethernet TCP/IP channel. If Slave devices are not connected to Ethernet you must temporarily (manually) connect them one by one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

Figure 127. Home Screen with Save Device



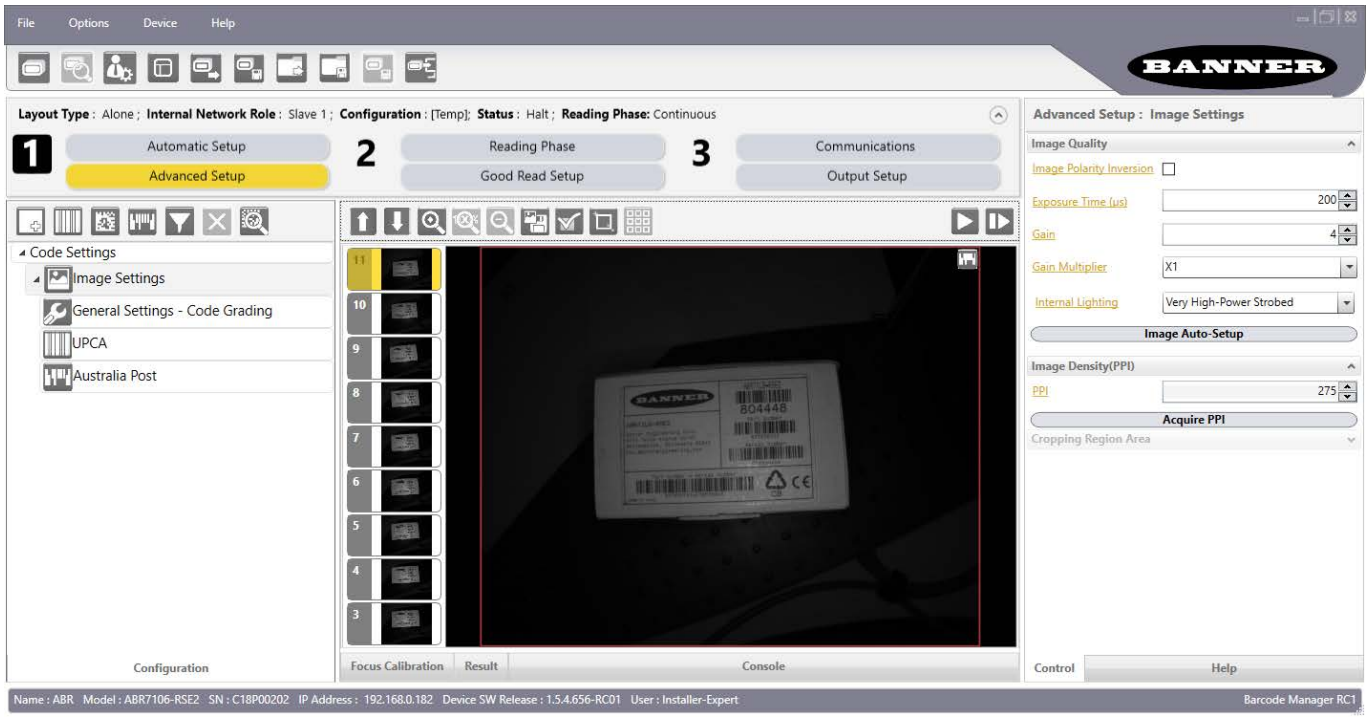
3. Open the Slave specific application job. It will either have the new name saved from the Master or Temp depending on the Save on Slave Device selection.

Figure 128. Save Configuration



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

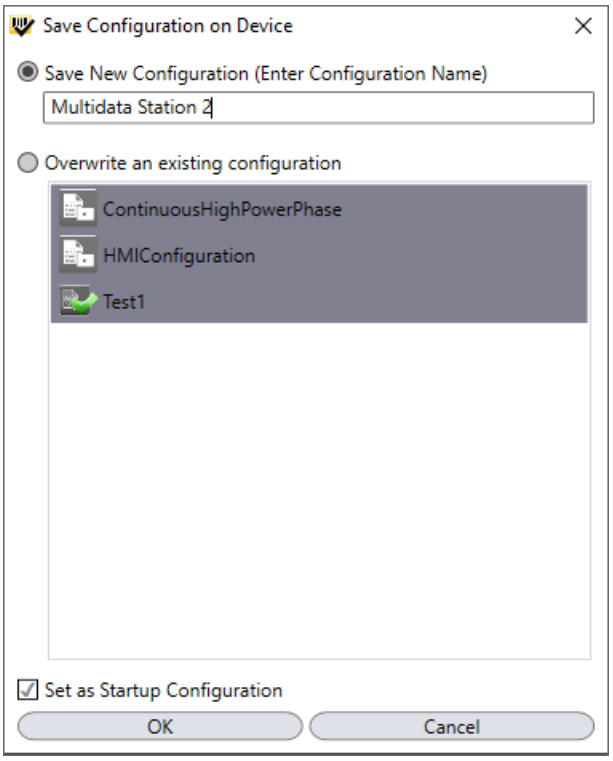
Figure 129. Slave 1 Configuration



5. Click **Capture** so the image updates with the latest settings.
6. Verify the focus and decoding using the capture image.
7. Save the parameters to a new Slave-specific application job name [15].

[15] 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.

Figure 130. Save Slave Configuration



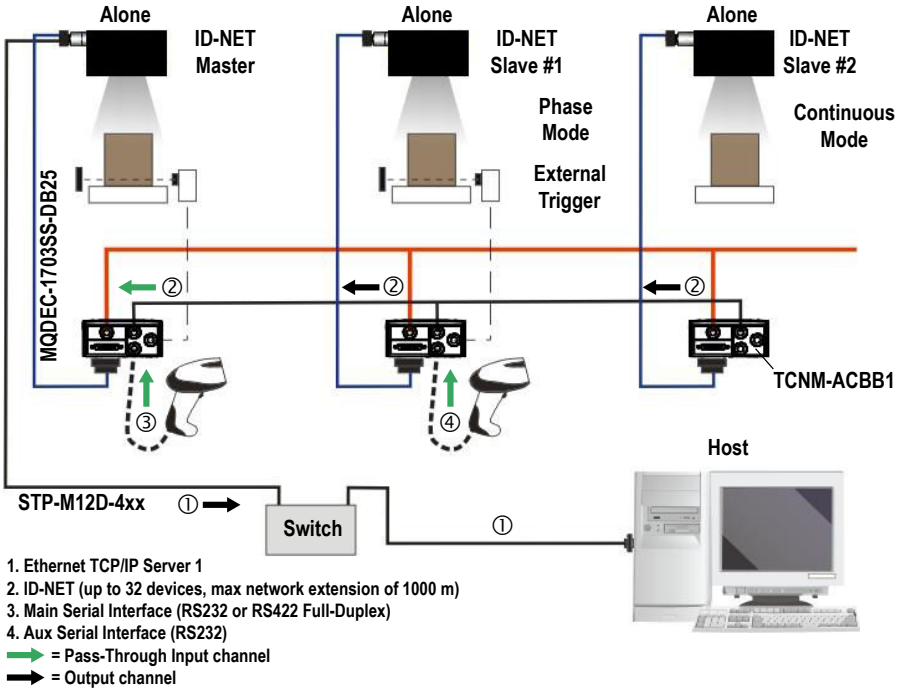
8. Repeat this procedure for each Slave device until the entire network is configured.

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 following figure as an example.

Figure 131. Example: ID-NET Multidata Layout (Pass-through)



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

Figure 132. ID-NET Master

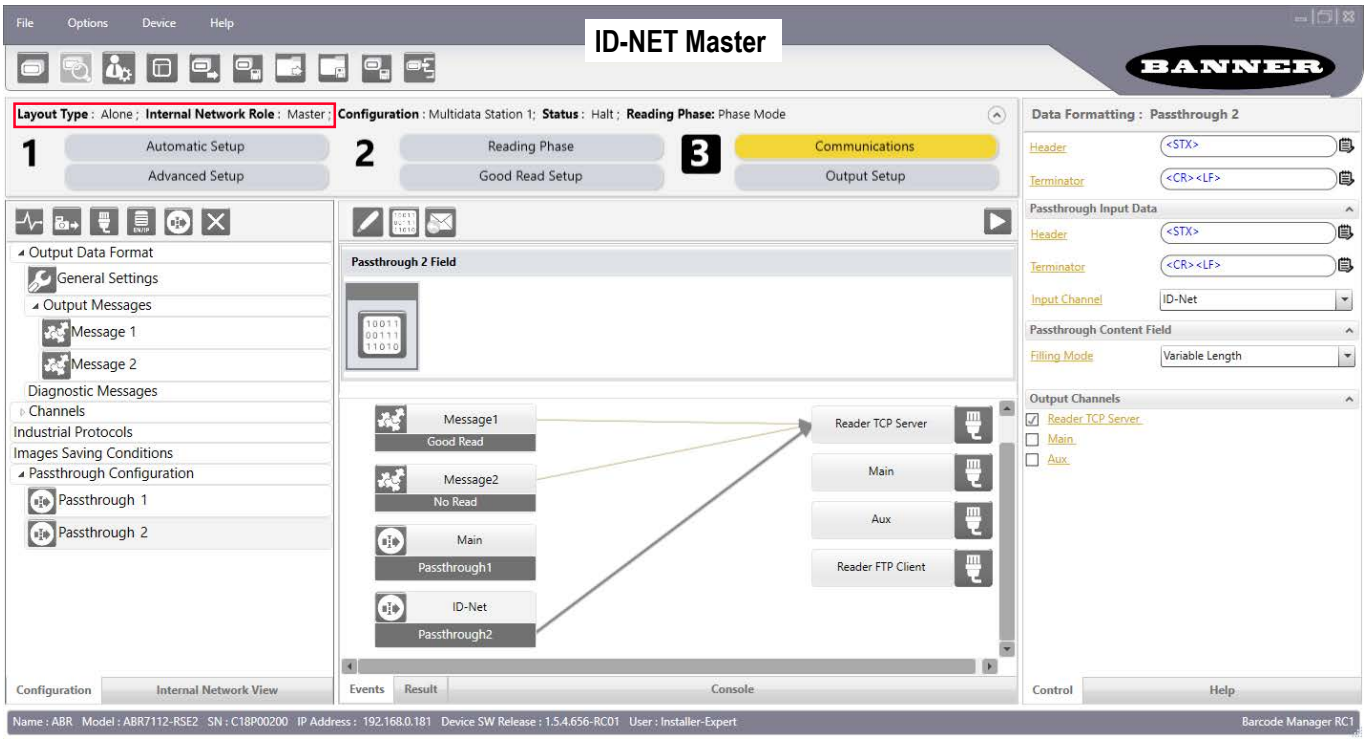


Figure 133. ID-NET Slave 1

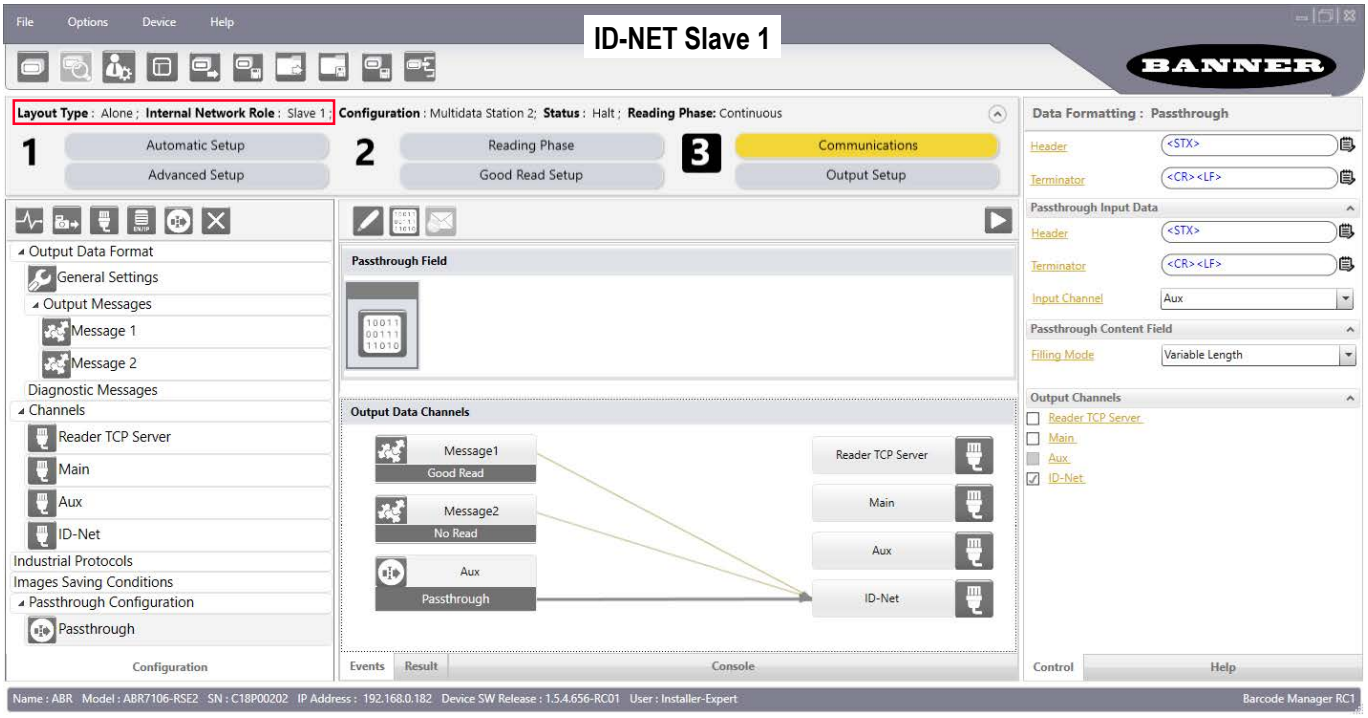
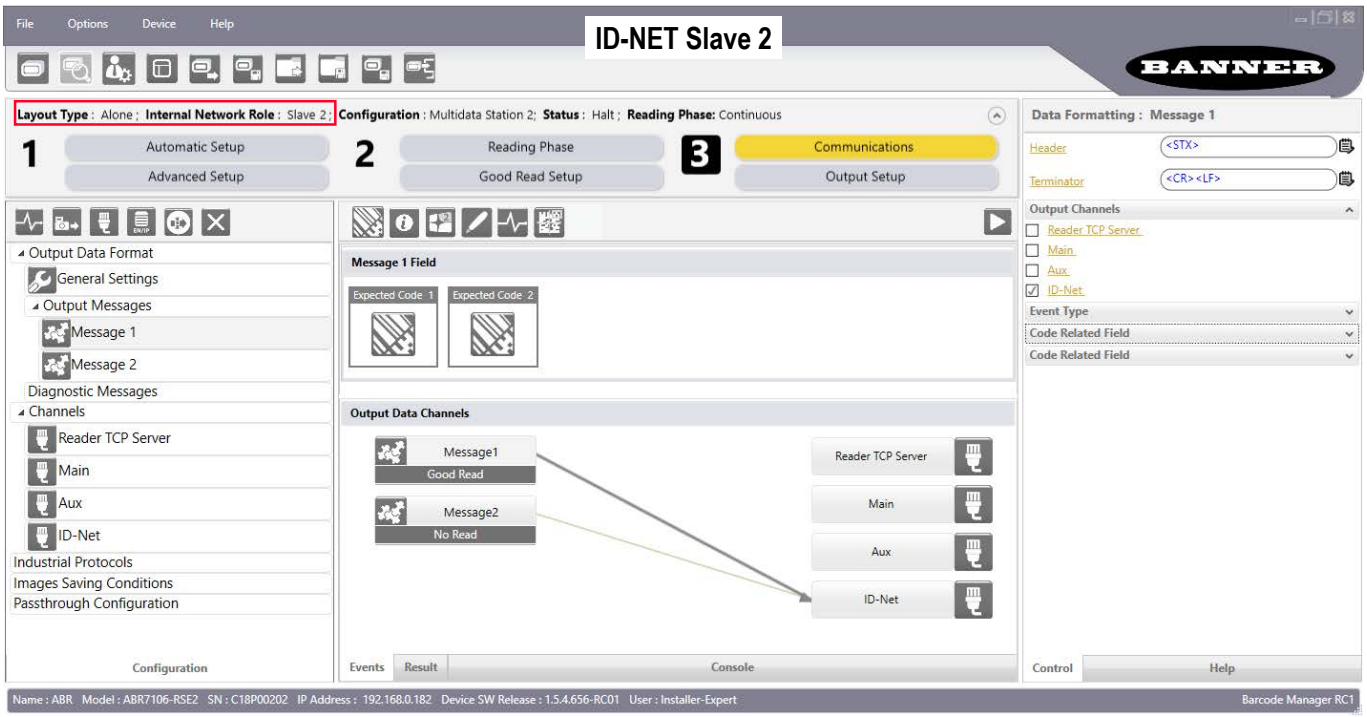


Figure 134. ID-NET Slave 2



Multidata Slave Commands

The following commands **are not** Host Mode Programming commands. They **do not** require the `Enter/Exit Host Mode` commands and therefore **do not** take place during a programming session. They are sent directly to the reader from the Host to perform their specific functions.

Communication Channels	
Allowed	**Not Allowed**
Serial	MODBUS TCP
Client TCP, port 1023	EtherNet/IP, SLMP, and PROFINET

The following command allows sending an encapsulated command string to a (Multidata) Slave #N through the Master (Multidata) device in an ID-NET reading system layout.

This command can be used to send Reading Phase ON, Acquisition Trigger, and Reading Phase OFF strings (for One Shot and Phase Mode operating modes) to a Multidata Slave device in an ID-NET reading system layout.

The reader `Send String To Slave Device #N` command must have the following format:

```
<ESC> ( <B0H> ADDR STRING ) <ESC>
```

Where:

- `ADDR`: Device Address
- `STRING`: String To Send (Length: 1 to 32)

ADDR is a character indicating address of the device in an ID-NET Master/Slave Multidata reading system layout:

ADDR = `<30H> + <Device Address>` where:

- **Device Address = 0:** Standalone device or Master Multidata ID-NET device
- **Device Address = 1 to 31:** Slave Multidata ID-NET device
- **Device Address = 32:** ID-NET broadcast address

This means:

- **ADDR = <30H>:** Standalone device or Master Multidata ID-NET device
- **ADDR = <31H> to <4FH>:** Slave Multidata ID-NET device
- **ADDR = <50H>:** ID-NET broadcast address

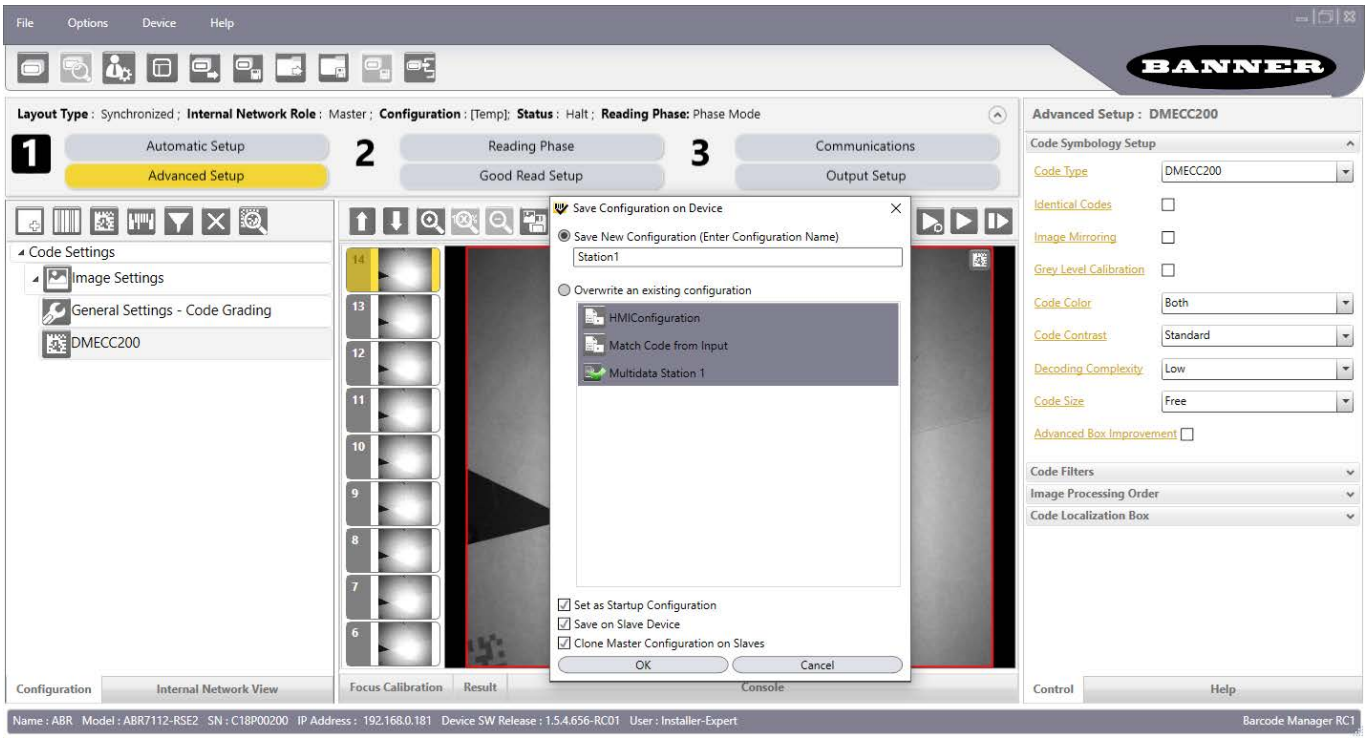
14.2.4 Synchronized ID-NET Network Configurations

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 the Phase Mode 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 select the option to **Clone Master configuration on Slaves**.

Figure 135. Saving Synchronized Phase Mode Configuration to Master

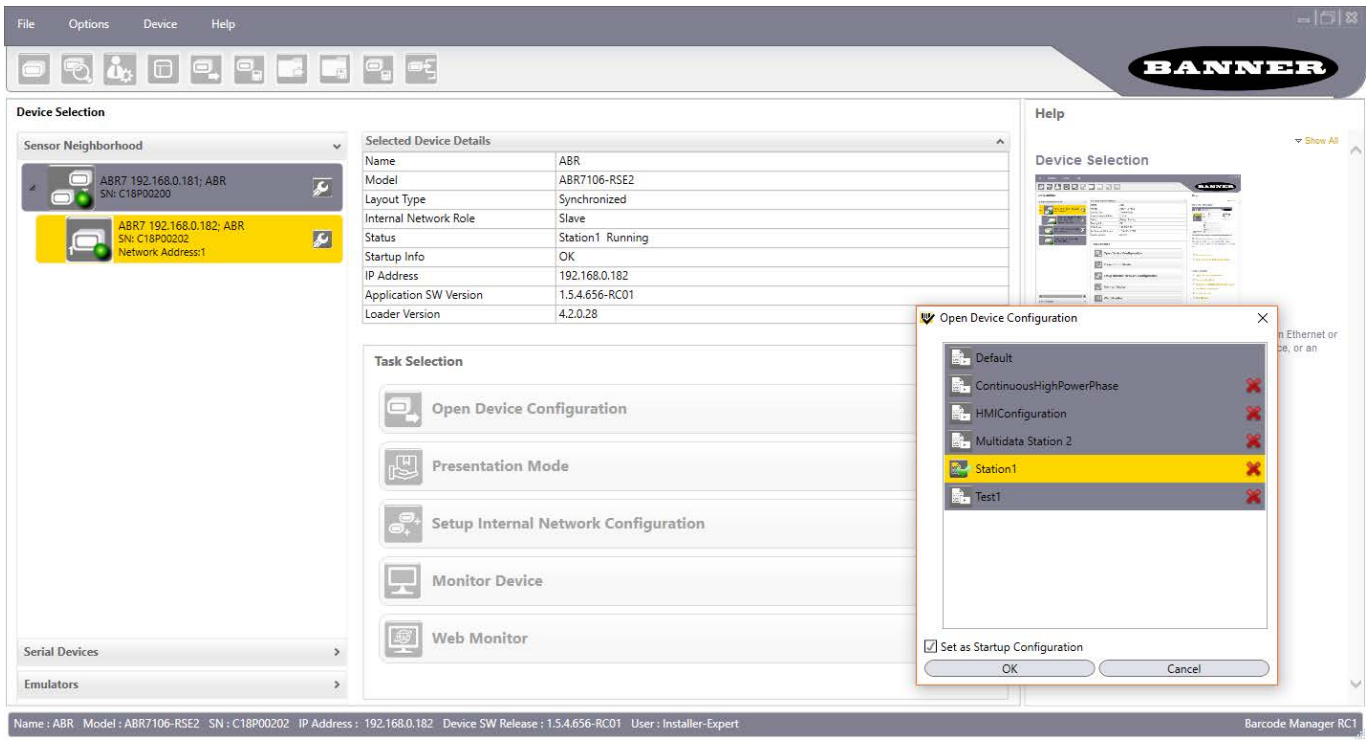


2. Connect to each Slave reader via Ethernet (see note below).

Note: If necessary, Slave device photometric (Image Settings) parameters must be configured separately through Barcode Manager. 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.

3. Set the Slave-specific parameters.
4. Open the cloned application job.

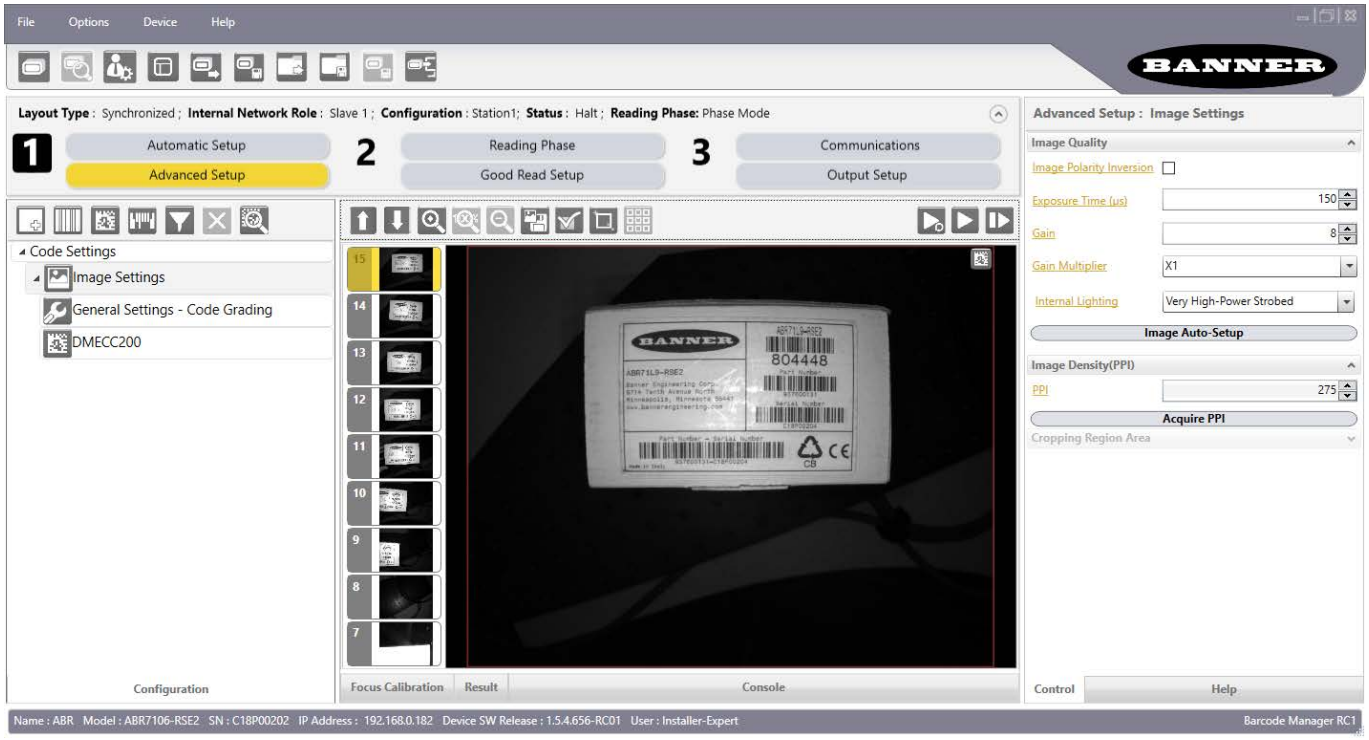
Figure 136. Opening the Cloned Synchronized Phase Mode Configuration on the Slave



5. After the job opens, pause run mode and configure the Slave-specific parameters.

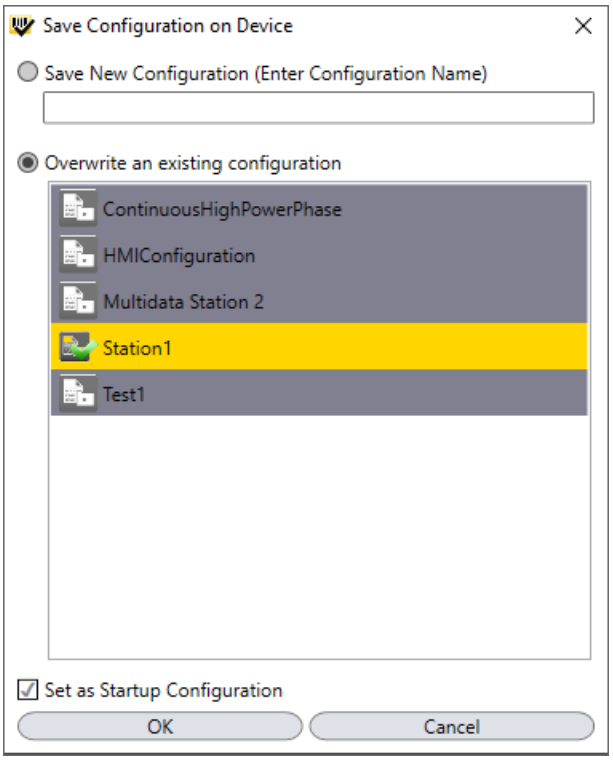
   These parameters depend on the application and include the following:
   - Photometric parameters (Image Auto-Setup feature in the Advanced Setup - Image Settings step)
   - Acquisition Trigger Delays necessary to avoid lighting interference between adjacent or oppositely positioned readers (Reading Phase step)
   - Images Saving, if used (Communications step)
   - Encoder Sensor, if used (for all Slaves, the Encoder Type must be set to **Internal**)

Figure 137. Accessing Image Settings on the Slave



6. Click **Capture** so the image updates with the latest settings.
7. Verify the focus and decoding using the capture image.
8. Save the the parameters, overwriting the cloned application job [16].

Figure 138. Saving Synchronized Phase Mode Configuration to Slave



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

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 Ethernet interface is used.

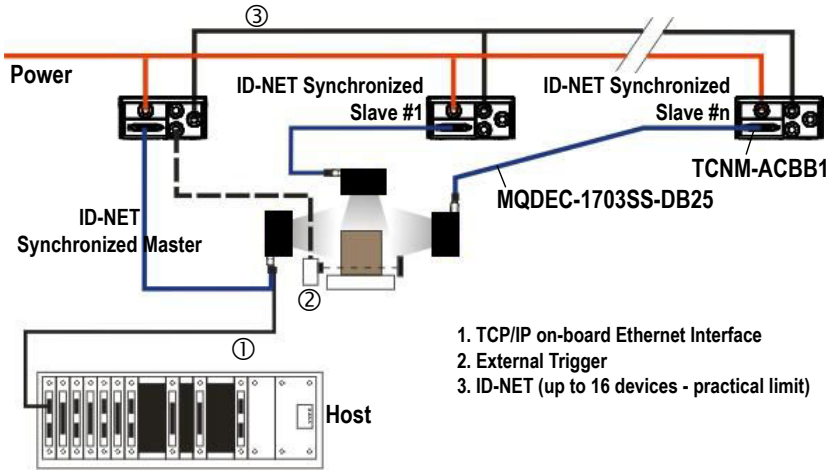
For a Master/Slave Synchronized layout the External Trigger signal is unique to the system; there is a single reading phase and a single message from the Master reader to the Host computer. It is not necessary to bring the External Trigger signal to all the readers.

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

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

[16] 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.

Figure 139. Example ID-NET Synchronized Layout with Master TCP/IP Ethernet Interface to Host



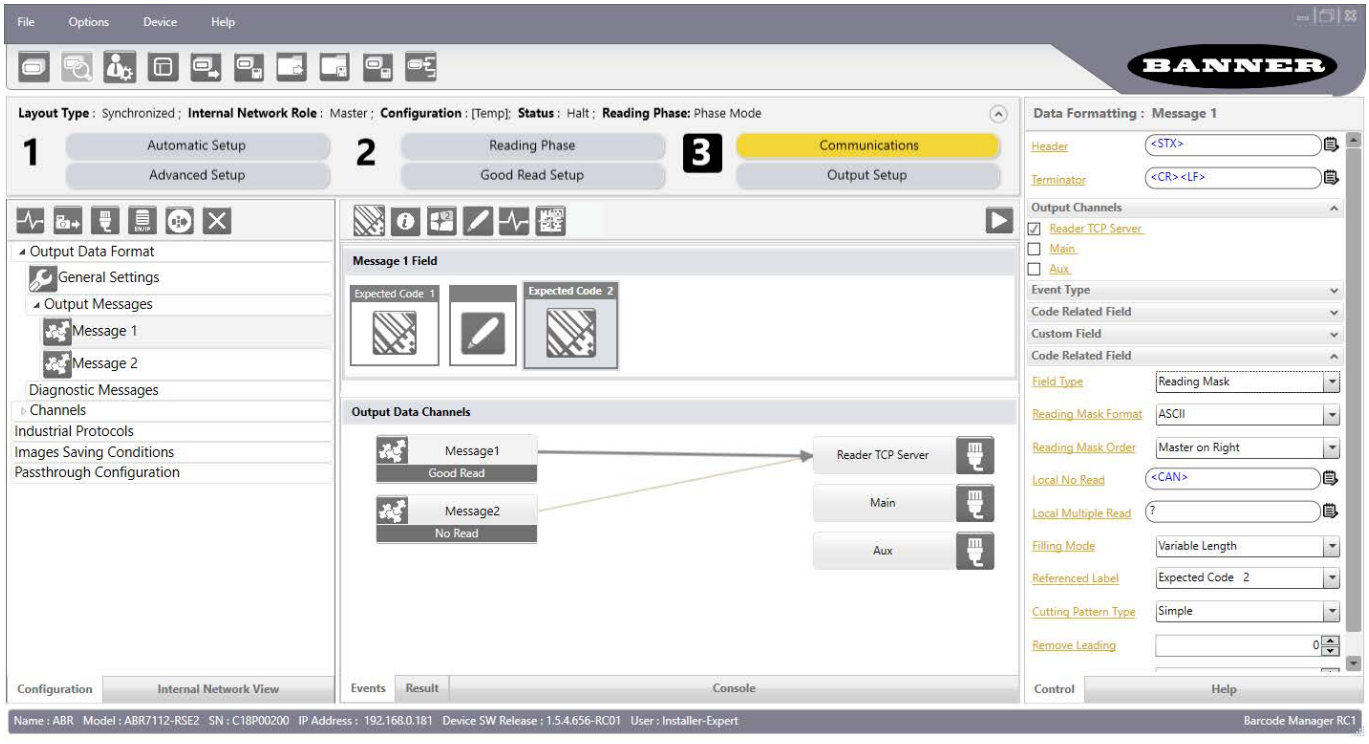
The Master reader can be connected to the TCNM-ACBB1 connection box with the advantage of the Backup and Restore configuration function (TCNM-ACBB1 + TCNM-ACMK-100 module).

All devices always support multiple output channels (that is, for data monitoring).

14.2.5 Verify Master/Slave Synchronization Configuration

1. From the Master configuration, run the application.
2. Monitor the output data from the Barcode Manager Console or a configured channel terminal.
3. If necessary, as a troubleshooting tip, follow steps 2–9 to temporarily apply the **Reading Mask** field in place of each **Code Content** field to verify if all devices are reading.
4. Connect to the Master device via Ethernet and from the Communications step.
5. Change each Expected Code **Field Type** from **Code Content** to **Reading Mask**.

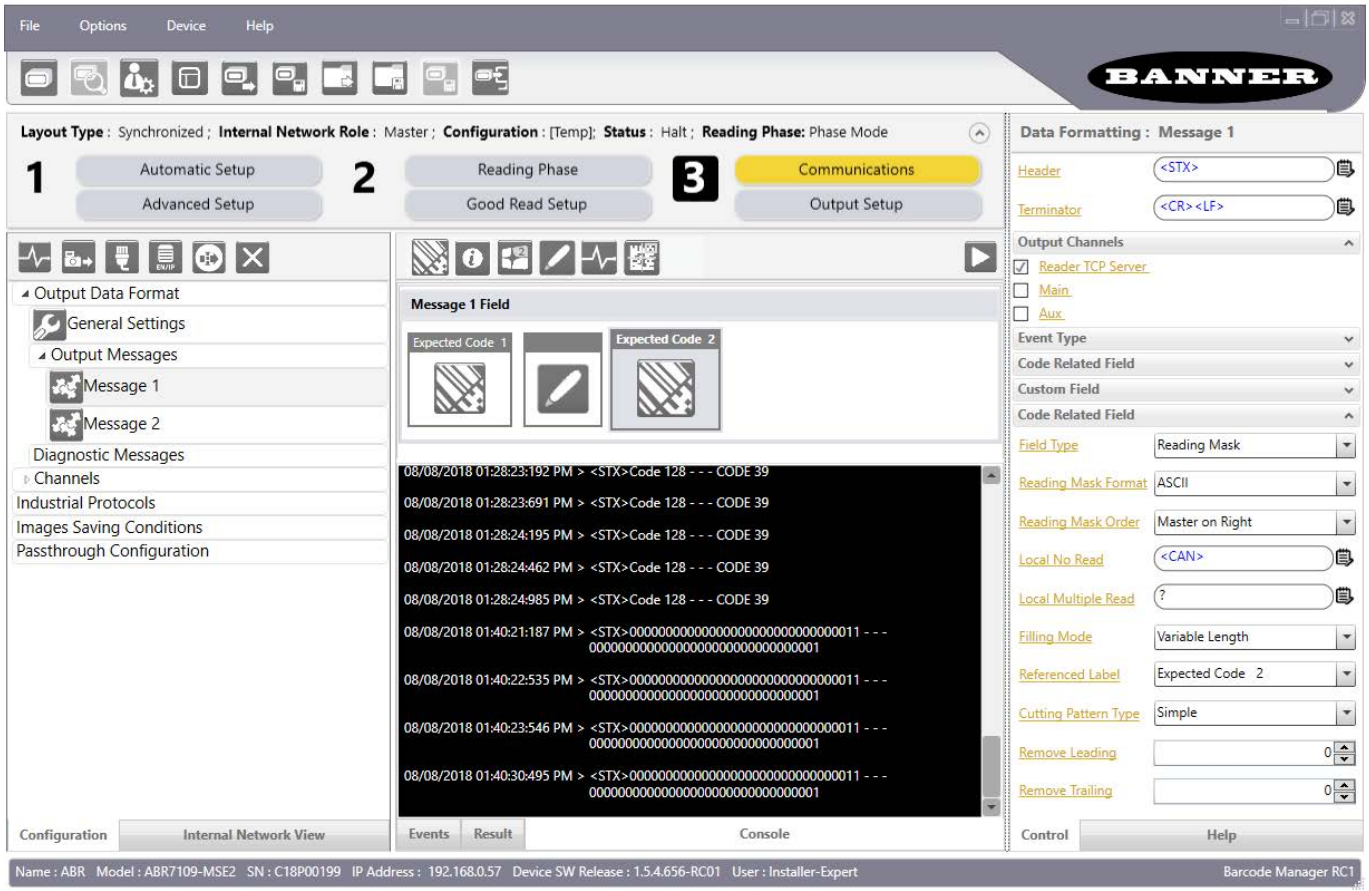
Figure 140. Apply a Reading Mask



6. Run the application.

7. Monitor the output data from the Barcode Manager Console or a configured channel terminal.

Figure 141. Reading Mask Example



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.

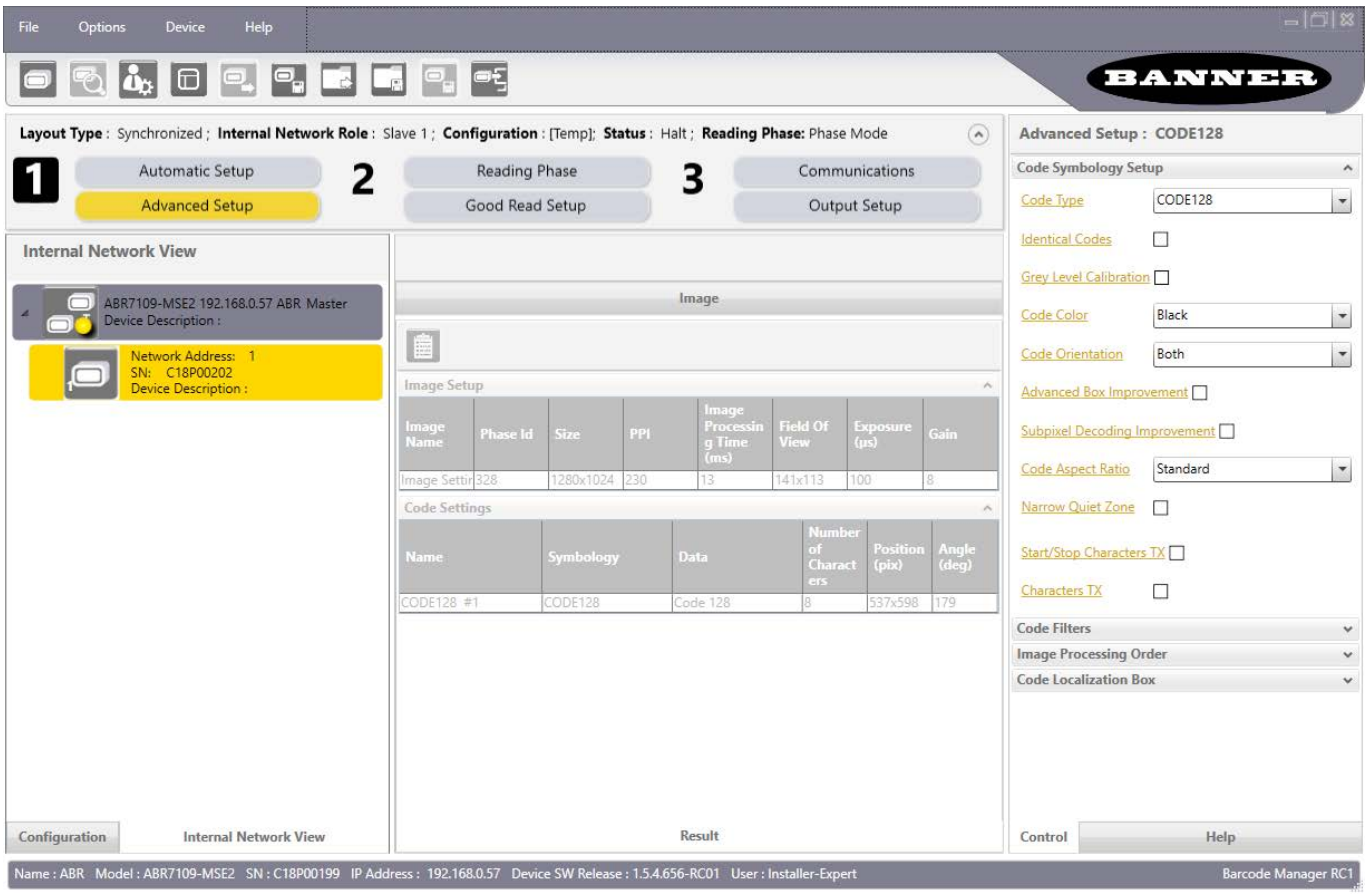
8. After verifying correct functioning of the reading devices, return the Expected Code fields from **Reading Mask** to **Code Content**.
9. If you haven't made any other changes, exit without saving. Otherwise, save the Master device configuration overwriting its previous one. Make sure to save without selecting **Clone Master Configuration on Slaves**, otherwise the Slave configurations will be overwritten.

View Connected Slave Configurations

To view the connected Slave configurations, complete the following steps.

1. Click 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.
4. Wait for the configuration to load.
5. Click the **Configuration** tab at the bottom of the screen.
   By selecting the various configuration steps above you can visualize the slave configuration.

Figure 142. View Connected Slave Configurations



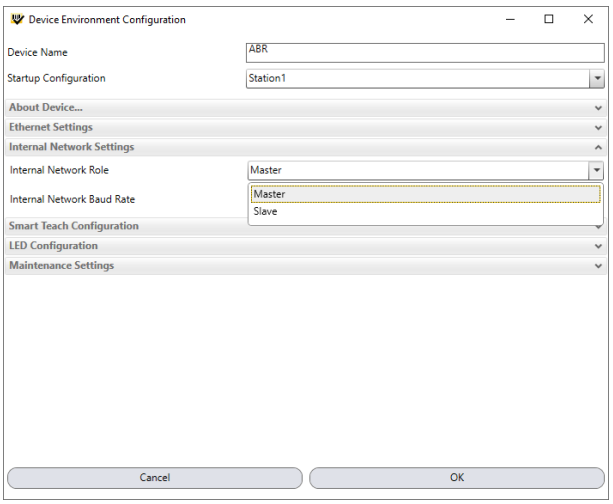
Note: You can modify some Slave Synchronized parameters from this view but you cannot save them here. To save changed slave parameters here, you must click on the Master and Save the configuration overwriting it, making sure to select **Save on Slave Device** but without **Clone Master Configuration on Slaves**, otherwise all the Slave configuration parameters will be overwritten by the Master configuration.

14.2.6 Alternative Device Role Selection

Use the following instructions as an alternate method to set up a Master/Slave Internal Network Role.

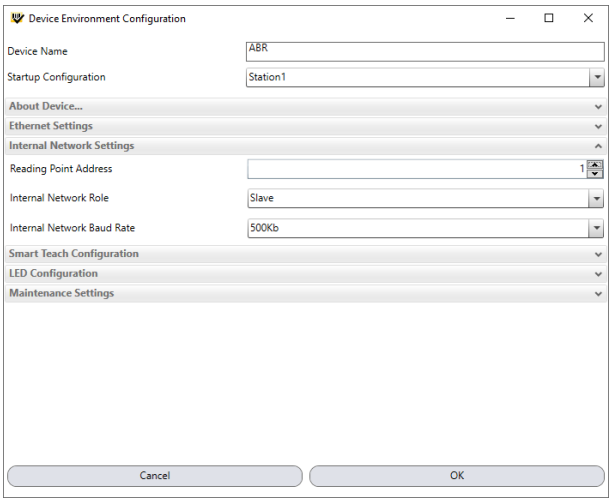
1. Connect to the reader selected to be the Master.
2. Go to **Device menu** > **Settings** > **Device Environment Settings** and open the **Internal Network Settings** group.
3. Change the Internal Network Role to Master.

Figure 143. Change Internal Network Role



4. 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 ( ).
5. Connect to a Slave device and use the same Device Settings menu to set the Internal Network Role and Reading Point Address (ID-NET address).

Figure 144. Set Slave Reading Point Address



   a) Complete the Slave configuration and save it to the device.
   b) Complete all the ID-NET Slaves in the same way.
6. Set the Master to expect the same number of slaves as planned in the application.
   Now by simply making the electrical connections the network will be automatically recognized upon power up. See the Note in Internal Network Icons on p. 130.

15 Code Grading

15.1 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 (that is, 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 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 ABR family of readers 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 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 Barcode Manager you can also configure the ABR 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 p. 153.

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

15.2 International Standards Applied to Code Grading

ABR 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.

15.3 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	x	x	Non Graded	x	Non Graded
Axial Non Uniformity	x	x	x	x	
Unused Error Correction	x	x	x	x	
Symbol Contrast	x	x		x	x
Cell Contrast			x		
Cell Modulation			x		
Decode			x	x	x
Fixed Pattern Damage			x	x	
Grid Non Uniformity			x	x	
Minimum Reflectance			x		x
Minimum Edge Contrast					x
Decodability					x
Modulation				x	x
Defects					x

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

Note: This can be done by an **Installer-Expert** user.

1. Go to **Advanced Setup** > **General Settings - Code Grading**.
2. Set the **Processing Mode** parameter to **Standard – Code Grading**.
3. 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 displays that allows you to select which parameters to use to determine the overall grading.

Figure 145. Example Code Grading Parameters



15.4 ISO/IEC 16022 and ISO/IEC 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.

15.4.1 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 elements dimension from the expected element dimension due to printing problems (that is, overprint or underprint).

Figure 146. Underprinting



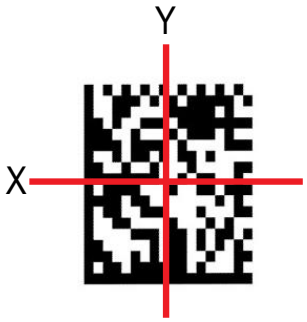
Figure 147. Overprinting



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.

Figure 148. Axial Non-Uniformity



Unused Error Correction (UEC)

Tests and grades 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.

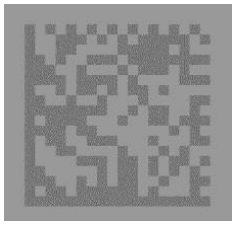
Figure 149. Unused Error Correction (UEC)



Symbol Contrast (SC)

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

Figure 150. Symbol Contrast (SC)



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

15.5 ISO/IEC TR 29158 (AIM DPM 2006) Quality Guideline

The ISO/IEC TR 29158 (AIM DPM 2006) 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.

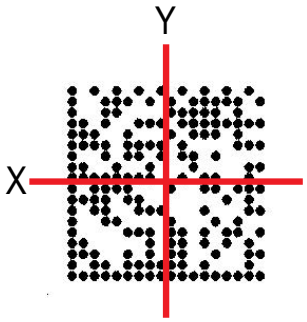
15.5.1 Code Quality Scan Grade Parameters

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.

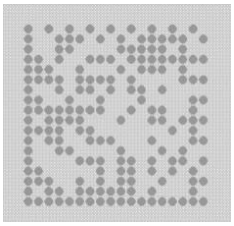
Figure 151. Axial Non-Uniformity (ANU)



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).

Figure 152. Cell Contrast (CC)



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.

Figure 153. Cell Modulation (CM)



Decode

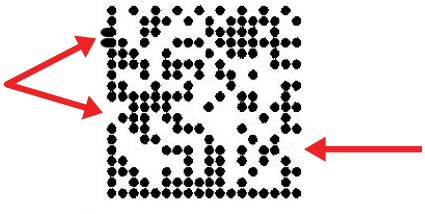
Tests, on a Pass/Fail basis, whether the symbol has all its features sufficiently correct to be readable. If the image cannot be decoded using the symbology reference decode algorithm, then it shall receive the failing grade 0. Otherwise, it shall receive the grade 4.

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.

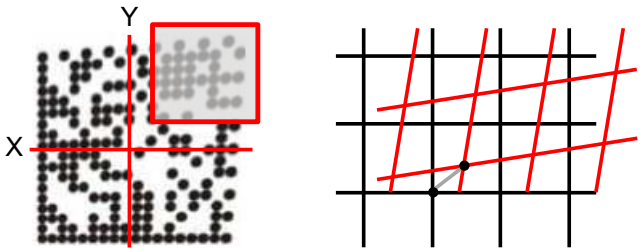
Figure 154. Fixed Pattern Damage (FPD)



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.

Figure 155. Grid Non-Uniformity (GNU)



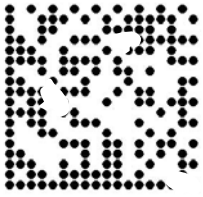
Minimum Reflectance (MR)

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)

Tests and grades 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.

Figure 156. Unused Error Correction (UEC)



Non-Graded Parameters

Print Growth

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

Figure 157. Underprinting

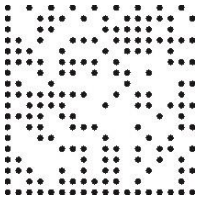
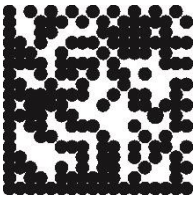


Figure 158. Overprinting



15.6 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.

15.6.1 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 (that is, overprint or underprint).

Figure 159. Underprinting

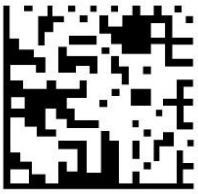


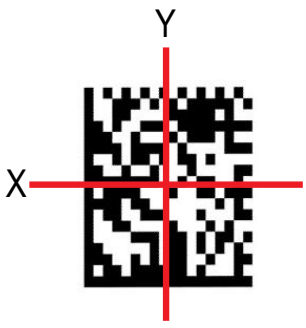
Figure 160. Overprinting



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.

Figure 161. Axial Non-Uniformity



Unused Error Correction (UEC)

Tests and grades 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.

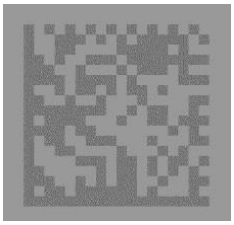
Figure 162. Unused Error Correction (UEC)



Symbol Contrast (SC)

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

Figure 163. Symbol Contrast (SC)



Decode

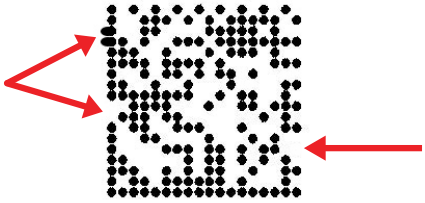
Tests, on a Pass/Fail basis, whether the symbol has all its features sufficiently correct to be readable. If the image cannot be decoded using the symbology reference decode algorithm, then it shall receive the failing grade 0. Otherwise, it shall receive the grade 4.

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.

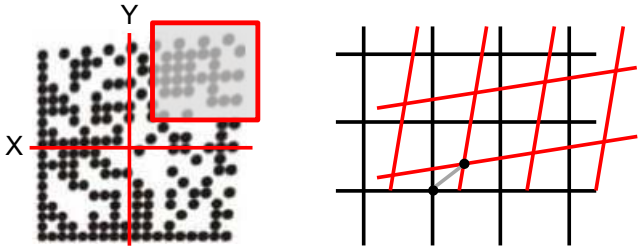
Figure 164. Fixed Pattern Damage (FPD)



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.

Figure 165. Grid Non-Uniformity (GNU)



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.

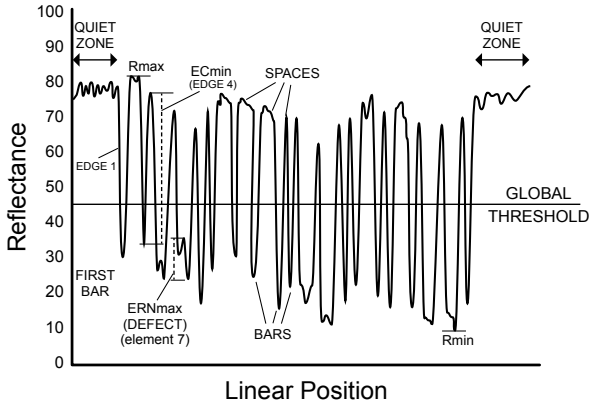
The Overall Code Grade is determined by the lowest resulting Scan Grade within the evaluated set of individual Scan Grade parameters.

15.7 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 166. 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 integers from 4 to 0 shall be allocated to each of the parameters evaluated

15.7.1 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.

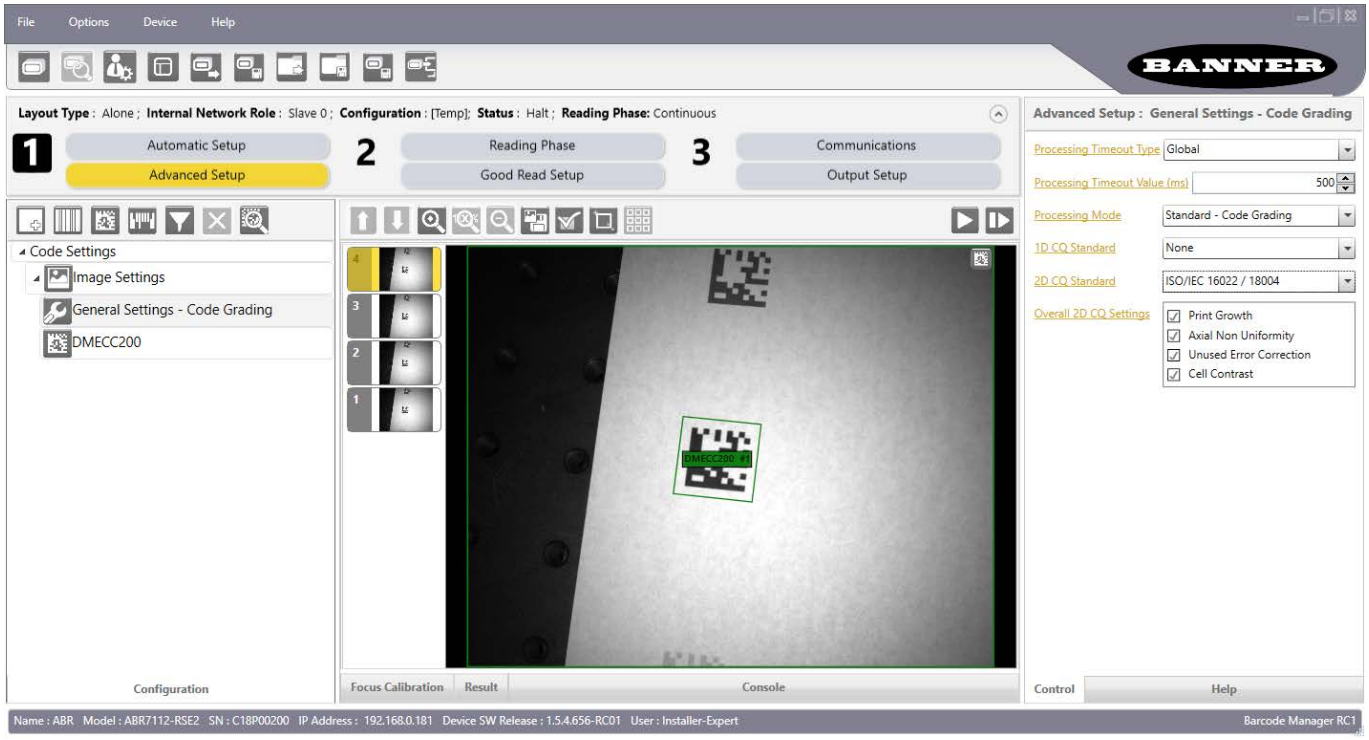
Symbol Contrast (SC)

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

15.8 Code Grading Example Using ISO/IEC 16022 and ISO/IEC 18004 Standards

1. Click **General Settings - Code Grading**.
2. Select **ISO/IEC 16022 / 18004** from the **2D CQ Standard** dropdown list.

Figure 167. Grading Parameters Set for ISO/IEC 16022 / 18004



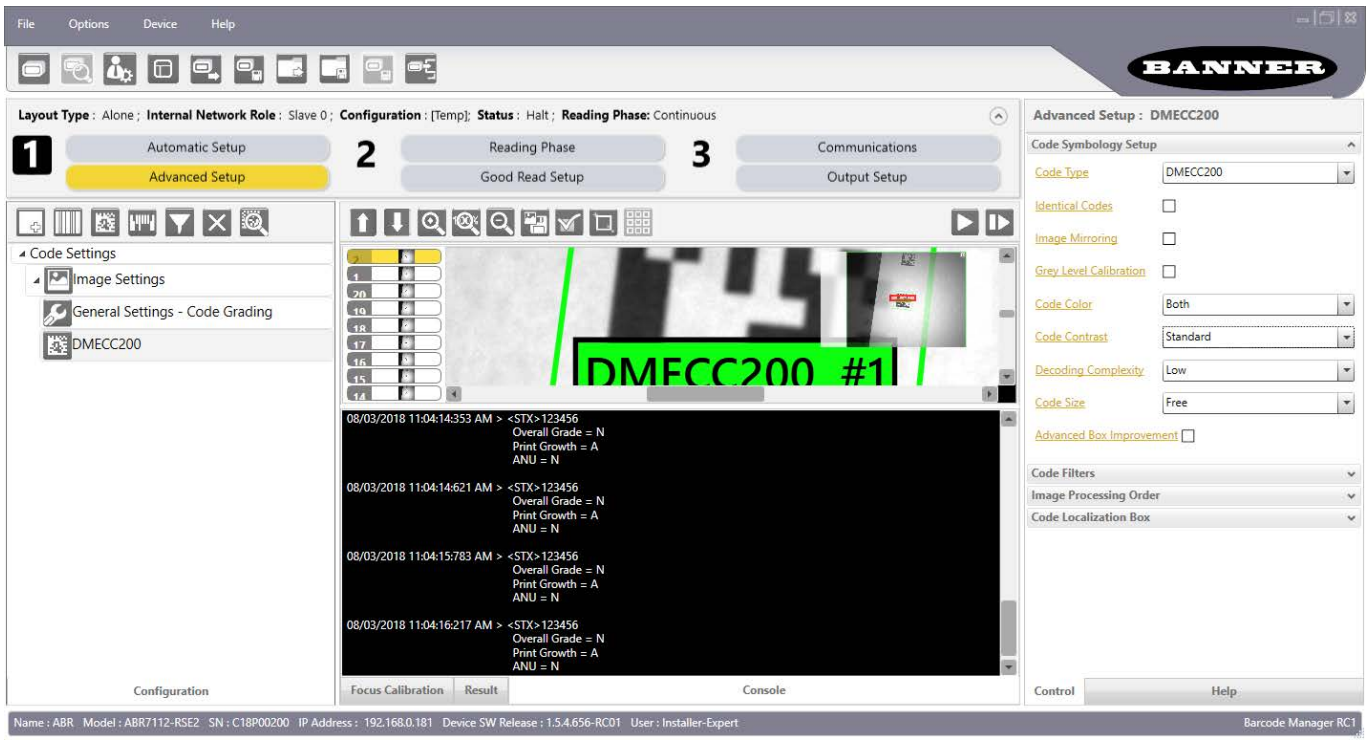
3. Deselect any **Overall 2D Code Quality** grade parameters that do not apply to the application.
4. Click **DMECC200** from the **Configuration** panel.
5. Click the **Result** tab at the bottom of the screen.
6. Click the **Single Result View** icon, and scroll down to view the code quality grade parameters.

Figure 168. Select a Code Type on the Code Settings Page to View Code Grades from the Result Tab



7. If necessary, define the output message to include the scan grade parameters.

Figure 169. Output Message

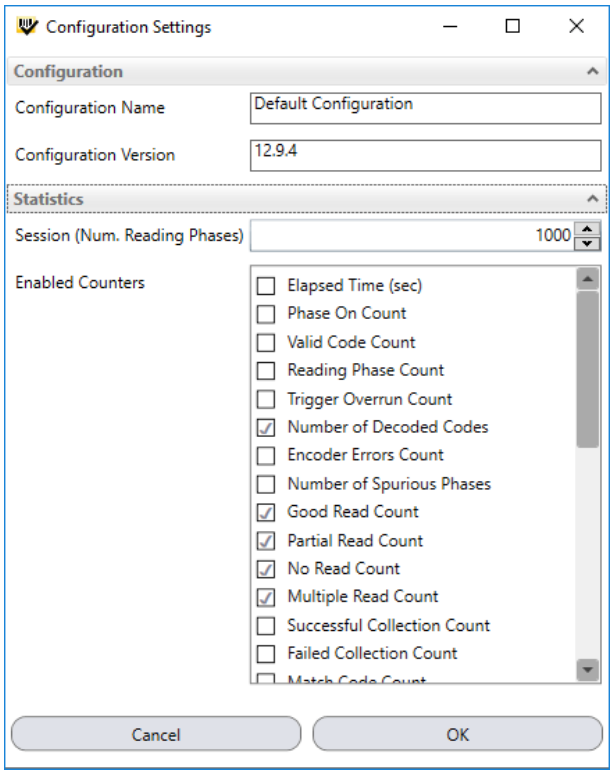


16 Device Configuration Settings

The Statistics behavior for the Barcode Manager Monitor is managed through the **Configuration Settings** window.

To access the device configuration settings, the desired device must be connected to Barcode Manager. Go to **Device** > **Settings** > **Configuration Settings** to open the **Configuration Settings** window.

Figure 170. Configuration Settings



This window includes the following fields: Configuration and Statistics.

The **Configuration** group includes **Configuration Name** and **Configuration Version** of the Default Configuration (read-only).

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

This **Statistics** group manages the statistics reporting behavior in the Monitor Statistics page. See also Monitoring Statistics on p. 114.

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.

17 Device Environment Settings

17.1 Accessing Device Environment Settings

To access the device environment settings, the device must be connected to Barcode Manager. From the **Device** > **Settings** > **Settings** menu open the **Device Environment Configuration** window.

This window presents the following fields:

Device Name

(Default “ABR”) Personalized by typing a new name in this field. This can be used to distinguish this device from others in the network (that is, a name to indicate the device position in the network).

This name also shows up in the Device List Area.

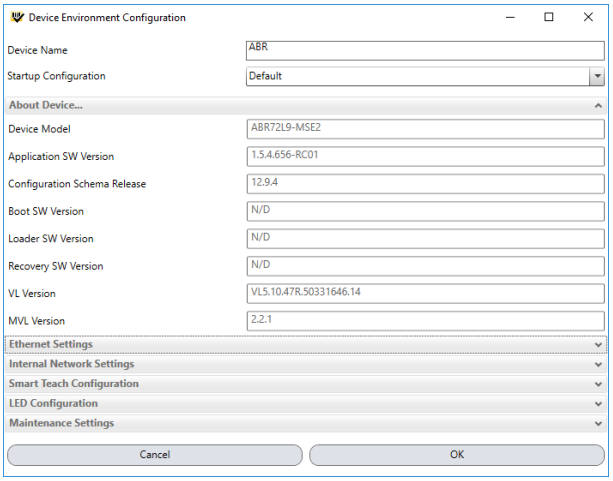
Startup Configuration

Change by selecting a different configuration from the dropdown list (if any).

About Device

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

*Figure 171. **Device Environment Configuration—About Device** Screen*



17.2 Ethernet Settings

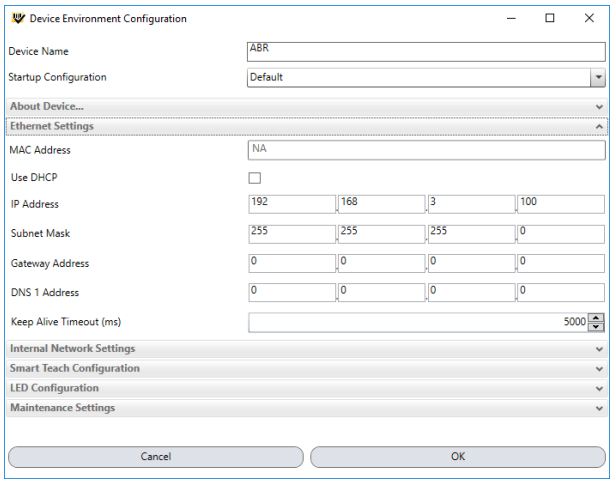
This group presents all the Ethernet Settings for correct device connection to the LAN.

It can also be accessed directly by clicking on the wrench icon in the Device List area as shown in Ethernet Device Discovery on p. 9.

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 Barcode Manager UI configuration environment.

*Figure 172. **Device Environment Configuration—Ethernet Settings** Screen*



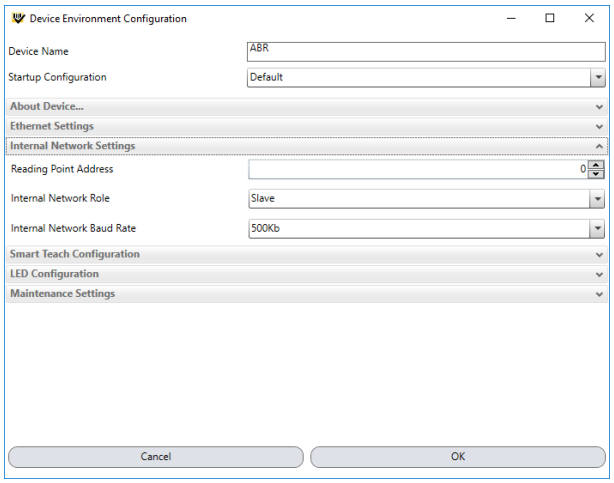
Note: When using Dynamic Host Configuration Protocol (DHCP), if there is a communication loss (that is, cable disconnect), the reader takes about 40 seconds to boot.

17.3 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).

*Figure 173. **Device Environment Configuration—Internal Network Settings** Screen*



The **Reading Point Address** indicates the address of the slave reader. This address is set automatically through the Setup Internal Network Configuration procedure (Internal Network Configurations on p. 130) but it can also be set here manually (1 to 31 for ID-NET Slaves).

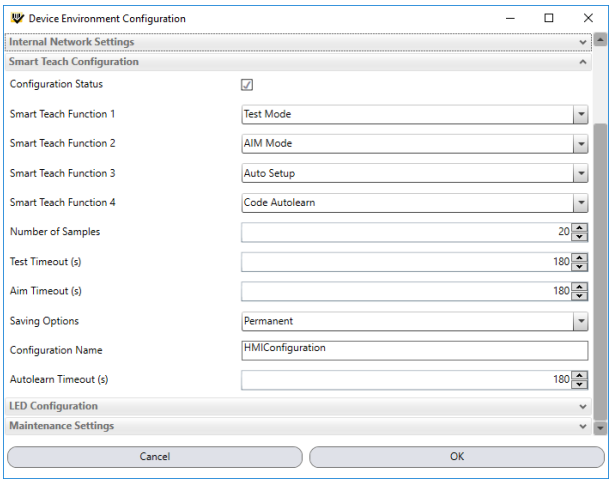
The ID-NET Baud Rate is selected here and must be common to all devices in the network.

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

17.4 Smart Teach Configuration

This group allows managing the Smart Teach behavior. See the device Instruction Manual for more details on the Smart Teach features.

*Figure 174. **Device Environment Configuration**—Smart Teach Configuration Screen*



Configuration Status

Enables or disables the Smart Teach features available through the multifunction key on the device.

Smart Teach Function (1-4)

Assigns a Smart Teach function selected from the list to each one of the Smart Teach 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 Smart Teach 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

The name given to the configuration saved to permanent memory by the Smart Teach Setup and Learn procedures.

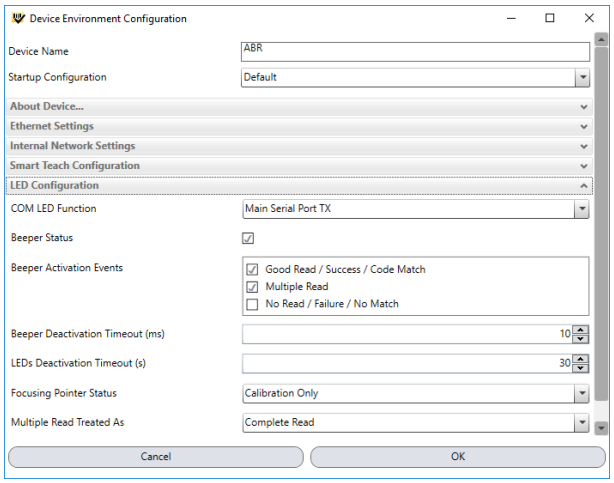
Autolearn Timeout

Sets when the expired timeout causes the Learn feature to exit.

17.5 LED Configuration

This group manages the device LEDs and Beeper behavior.

*Figure 175. **Device Environment Configuration—LED Configuration** Screen*



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 or 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

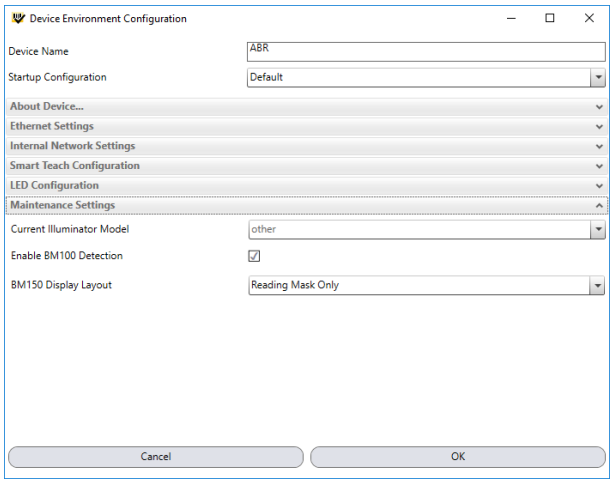
Multiple Read Treated As

Determines which device LED will be activated on a Multiple Read event:

- Complete Read—The Good Read LED is activated
- No Read—The Status LED is activated

17.6 Maintenance Settings

*Figure 176. **Device Environment Configuration—Maintenance Settings** Screen*



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.

PPI

This 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 TCNM-ACMK-100 Detection

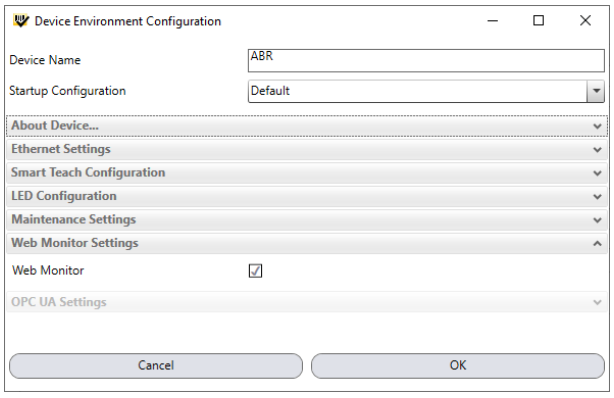
If enabled, at startup, the reader sends a message to recognize the presence of, and communicate with, the External Backup Memory (TCNM-ACMK-100 Backup Module).

If using the Backup Memory, this parameter must be enabled.

17.7 Web Monitor Settings

This setting enables the **Web Monitor** function for the reader.

*Figure 177. **Device Environment Configuration—Web Monitor** Screen*



For more information, see Web Monitor on p. 108.

18 PPI (Pixels Per Inch) Setup Chart

Print and use the Setup Chart on the following page to aid in aiming and focusing the reader (ABR 7000 liquid lens autofocus models), or in the Acquire PPI (Pixels Per Inch) procedure in the Advanced Setup of the reader (ABR3 000 and ABR 7000 manual focus models). Using this chart when running the AutoSetup procedure or the Focus Autolearn procedure on a liquid lens autofocus model typically results in a more accurate focus/reading distance. This in turn results in a better image and a more accurate PPI value. On any model, using this chart to accurately configure the PPI value allows the ABR to provide accurate module size measurements of barcodes for a given distance.

PPI (Pixels Per Inch) Setup Chart

Code 128

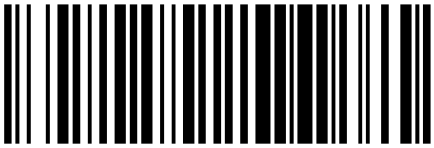
Resolution mm (mils)

0.30 (12)



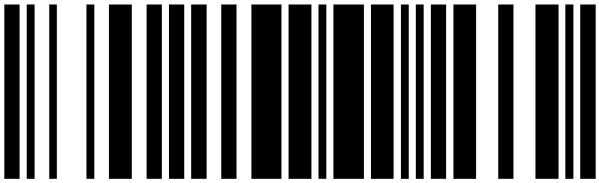
0.30 mm

0.50 (20)

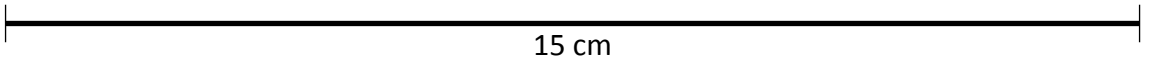


0.50 mm

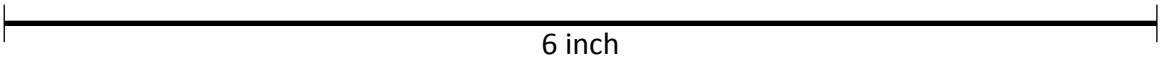
1.00 (40)



1 mm



15 cm



6 inch

Do not use these Barcodes for Smart Teach Autolearn

Do not scale this page

19 Troubleshooting

Problem	Solution
Online device is not displayed in the **Device Selection** area	To be found by Barcode Manager, Online devices must be powered on and connected to the Local Area Network. If you don't see the desired device within this list, verify the connections to the LAN and assure it is powered on. Then, click the **Discovery** icon to run a new device search.
Serial device is not displayed in the **Device Selection** area	To be found by Barcode Manager, Serial devices must be powered on, connected to 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 the **Discovery** icon to run a new device search. If the **Enable Serial Devices Discovery** checkbox is not selected (see **Options** > **UI Settings** > **Global Settings** > **Find Devices**), the Serial Devices list/button is not displayed in the Device Selection Area. See Serial Device Discovery on p. 10.
Cannot open a device configuration Barcode Manager **Selected Device Information** area shows a yellow background	Exit Barcode Manager and cycle the power to the device. Run Barcode Manager 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 gray)	The device is in run mode. Click **Pause** button to exit run mode.
Connection problem between Barcode Manager and Online Device	If more than one LAN card is present on the local PC and an Emulator (offline device) is enabled in Barcode Manager (disabled by default), the program can freeze up. Either disable the emulators in Barcode Manager 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.
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 (that is, 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.

20 Product Support and Maintenance

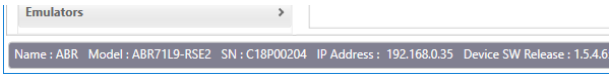
20.1 Update the Software and Firmware

The current version of Barcode Manager software and the device firmware is available for download from www.bannerengineering.com.

20.1.1 Update the Firmware

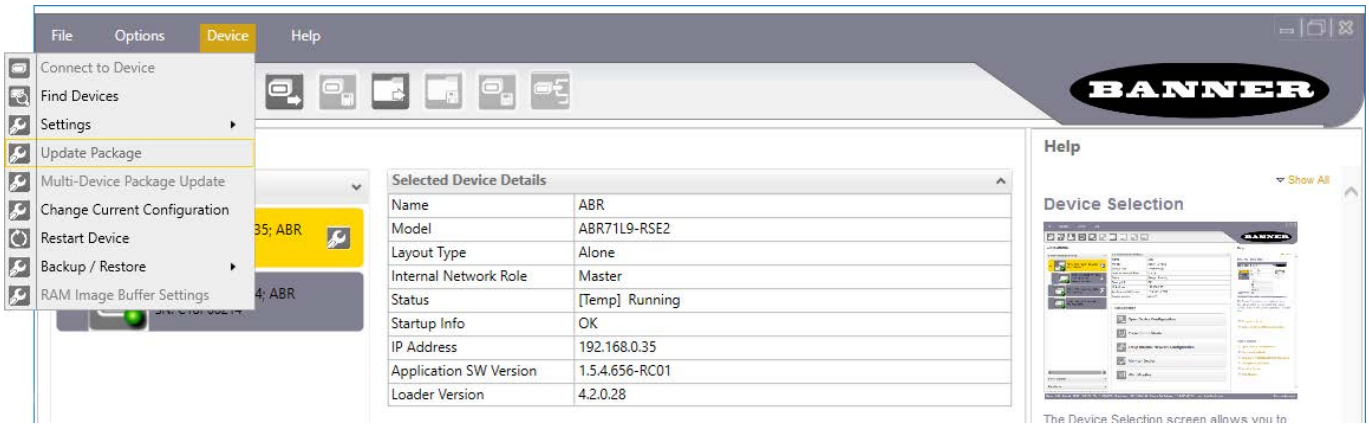
1. Make sure the device to be updated is selected and is shown in the status bar.

Figure 178. Status Bar



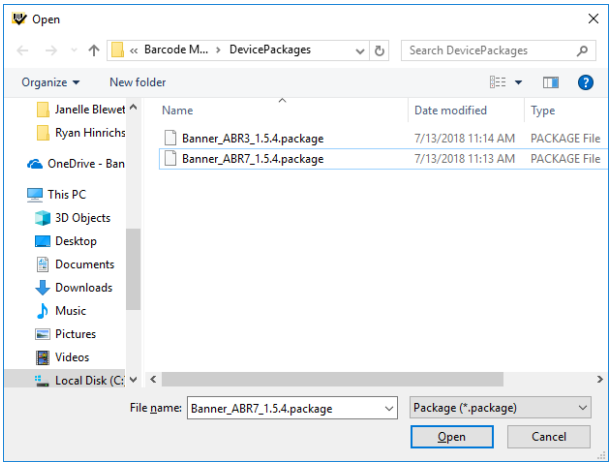
2. Go to **Device** > **Update Package**.

*Figure 179. **Update Package***



3. Select the firmware package file for your device family.

Figure 180. Select the Firmware File



The **Device Package Update** window displays.

4. Click **Yes** to start the update.
   The **Device Restart Information** window displays.
5. Click **OK** to restart the device.
   The update finishes, the reader restarts, and Barcode Manager returns to the **Home/Device Selection** page.
6. If the reader is not listed in the **Device List Area**, click **Find Devices** to search for the reader.

20.2 Contact Us

Banner Engineering Corp. headquarters is located at:

9714 Tenth Avenue North
Minneapolis, MN 55441, USA
Phone: + 1 888 373 6767

For worldwide locations and local representatives, visit www.bannerengineering.com.

20.3 Banner Engineering Corp Limited Warranty

Banner Engineering Corp. warrants its products to be free from defects in material and workmanship for one year following the date of shipment. Banner Engineering Corp. will repair or replace, free of charge, any product of its manufacture which, at the time it is returned to the factory, is found to have been defective during the warranty period. This warranty does not cover damage or liability for misuse, abuse, or the improper application or installation of the Banner product.

THIS LIMITED WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES WHETHER EXPRESS OR IMPLIED (INCLUDING, WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE), AND WHETHER ARISING UNDER COURSE OF PERFORMANCE, COURSE OF DEALING OR TRADE USAGE.

This Warranty is exclusive and limited to repair or, at the discretion of Banner Engineering Corp., replacement. **IN NO EVENT SHALL BANNER ENGINEERING CORP. BE LIABLE TO BUYER OR ANY OTHER PERSON OR ENTITY FOR ANY EXTRA COSTS, EXPENSES, LOSSES, LOSS OF PROFITS, OR ANY INCIDENTAL, CONSEQUENTIAL OR SPECIAL DAMAGES RESULTING FROM ANY PRODUCT DEFECT OR FROM THE USE OR INABILITY TO USE THE PRODUCT, WHETHER ARISING IN CONTRACT OR WARRANTY, STATUTE, TORT, STRICT LIABILITY, NEGLIGENCE, OR OTHERWISE.**

Banner Engineering Corp. reserves the right to change, modify or improve the design of the product without assuming any obligations or liabilities relating to any product previously manufactured by Banner Engineering Corp. Any misuse, abuse, or improper application or installation of this product or use of the product for personal protection applications when the product is identified as not intended for such purposes will void the product warranty. Any modifications to this product without prior express approval by Banner Engineering Corp will void the product warranties. All specifications published in this document are subject to change; Banner reserves the right to modify product specifications or update documentation at any time. Specifications and product information in English supersede that which is provided in any other language. For the most recent version of any documentation, refer to: www.bannerengineering.com.

For patent information, see www.bannerengineering.com/patents.

Index

A

ABR
  barcode manager 71
  reboot 71
advanced setup
  adjustable focus 16
  manual adjustable focus 13
ASCII
  escape sequence 70
  Hexadecimal value 70
autolearn
  code 32
automatic image settings 29

B

backup 43, 44
Barcode Manager
  install 8
  serial device discovery 10
  USB device discovery 10

C

code
  collection 52
  combination 53
  match 55, 56
  presentation 54
code autolearn 32
code collection 20
code combination 20
code grading 144
code presentation 20
Command
  setMessage 71
Comment 70
communications 21, 71
configuration
  LED 158
  LED configuration 158
  multi device 128, 130, 131, 134, 136, 137, 139–142
  multi device configuration 128, 130, 131, 134, 136, 137, 139–142
  pass-through 128
continuous reading phase 19

D

data format 21
defaults 44, 45
device environment settings 156
device list 23
DPM
  algorithms 34

E

ECMAScript 73
environment settings
  device 156
Ethernet settings 156
external image settings 29

F

filtering 33, 34, 38, 40
focus
  manual adjustment 13
Function
  onResult 70, 71

G

good read 20

H

Host Mode Programming
  command 76

I

image cropping 31
image filtering 33, 34, 38, 40
image filters 34
image saving 119–121, 126, 127
image settings
  automatic 29
  external 29
Index value 73

J

JavaScript 73

L

Line Feed 70

M

manual adjustable focus 13
match code 20
Message 71
message formatting 64–67, 69–78
monitor 108, 110, 112, 113

O

Object
  output 71
  Output 70
  result 70
output 22
Output
  discrete 71
  Setup 71

P

pass-through configuration 128
PI
  , See Protocol Index
Ports
  active 71
  TCP Server 71
Property
  nested 74–78
  value 74–78
Protocol Index 60

R

reading phase
  continuous 19
  one shot 19
  phase mode 19
Reading Phase 71
reset software 46
reset to defaults 44, 45
restore 43, 44

S

save
  image 119–121, 126, 127
Script
  event 71
serial device discovery 10
setup
  adjustable focus models 16
  advanced 13, 16
  automatic 11
  liquid lens autofocus models 16
software reset 46
Special characters 73
String
  count string 73
  lowercase 73
  pattern 73
  slice 73
  split 73
  subset 73
  substring 73
  uppercase 73
symbol verification 144

U

USB device discovery 10
user level 28

V

Variable
    custom 76
    matchcode 76

W

Web Monitor 108, 110, 112, 113