

DataMan[®] Fixed-Mount WebUI

Reference Manual

2026 July 30

Revision: 26.2.0.44

Legal Notices

The software described in this documentation is provided "as is" and "as available," without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and non-infringement. In no event shall Cognex Corporation and its global affiliates (collectively, "Cognex") be liable for any claims, damages, or other liability arising from, out of, or in connection with the software or your use of the software.

Your use of this software is governed by the full terms and conditions of the Cognex Software License Agreement. For complete licensing details and legal terms, please visit <https://www.cognex.com/terms-and-conditions>.

The information in this document is subject to change without notice and should not be construed as a commitment by Cognex. Cognex is not responsible for any errors that may be present in either this document or the associated software.

Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, nor transferred to any other media or language without the written permission of Cognex Corporation.

Portions of the hardware and software provided by Cognex may be covered by one or more U.S. and foreign patents, as well as pending U.S. and foreign patents listed on the Cognex web site at: cognex.com/patents.

In the event of any inconsistency between the English language version of this documentation and any other language version, the English language version shall take precedence.

Copyright © 2026. Cognex Corporation. All Rights Reserved.

The following are registered U.S. trademarks of Cognex Corporation:

2DMAX, 3D LIGHTBURST TECHNOLOGY, ADVANTAGE, ALIGNPLUS, ASSEMBLYPLUS, COGNEX, COGNEX VISION FOR INDUSTRY, CVL, DATAMAN, DVT, EASYBUILDER, EDGE INTELLIGENCE, HOTBARS, IDMAX, IN-SIGHT, LIGHTBURST, MX, OMNIVIEW, PATFIND, PATFLEX, PATINSPECT, PATMAX, PATMAX REDLINE, PATQUICK, POWERGRID, QUICKSORT, ULTRALIGHT, VISION SOLUTIONS, VISIONPRO, VISIONVIEW

The following are trademarks of the Cognex Corporation:

1DMAX, 3D SYMBOLIC LIGHT, 3D-LOCATE, 3DMAX, ONEVISION, PINPOINT, RANGEMAX, SLX, VIDI, OCRMAX, INSPECTEDGE, VISION SOLUTIONS

Portions copyright © Microsoft Corporation. All rights reserved.

Portions copyright © MadCap Software, Inc. All rights reserved.

Other product and company trademarks identified herein are the trademarks of their respective owners.

Symbols

The following symbols indicate safety precautions and supplemental information:

 **WARNING:** This symbol indicates a hazard that could cause death, serious personal injury or electrical shock.

 **CAUTION:** This symbol indicates a hazard that could result in property damage.

 **Note:** This symbol indicates additional information about a subject.

 **Tip:** This symbol indicates suggestions and shortcuts that might not otherwise be apparent.

Table of Contents

Legal Notices	2
Symbols	3
Table of Contents	4
About the DataMan WebUI	6
Connecting to WebUI	7
Troubleshooting	8
Setting your IP Address to DHCP	8
Using Static IP addresses	8
Connecting on a Managed Network	8
Connecting through USB-C	8
Application Steps	9
Home	9
Code Setup	11
Positioning	11
Tuning	12
Test	17
Review	17
Application Details	20
Trigger Settings	20
Code Quality	21
Data Validation	22
Data Formatting	24
Communications	28
Communications	28
Inputs	54
Outputs	56
File Transfer	57
Runtime	62
Diagnostics	64
User Interface	64
Signals	65
Display Modes	65
User Actions	66
Examples	66
TruCheck™ Grading	68
Configuring Grading	68
Setting up TruCheck™	69
Application Standards	70
Calibration Process	71
Transferring Verification Reports	71
Examining TruCheck™ Results	72
ISO/IEC 15416 Quality Parameters	73
ISO/IEC 15415 Quality Parameters	75

ISO/IEC TR 29158 AIM-DPM-1-2006 Quality Parameters	79
Settings	81
Overview	81
Configure Codepage	81
Licenses	84
Date & Time	85
Password Protection	86
Turning on Password Protection	86
Locking the User Interface	86
Forgotten Password	87
Removing the Password	87
Program Management	87
Add New Program	88
Program Management with DMCC	89
Network Settings	90
Ethernet 0 Settings	90
Ethernet 1 Settings	91
Firmware Update	93
Services & Ports	93
Backup Device	95
Restore Device	96
Factory Reset	96
Device Log	97
Field Diagnostics	98
Reboot	98
Certificates	98
Prerequisites	99
Install a TLS Certificate	99
Install a Trusted Certificate	100
Image Viewer	101

About the DataMan WebUI

The DataMan WebUI provides an intuitive interface for managing your DataMan readers. The WebUI streamlines configuration and monitoring tasks. Key features include:

- Live code review: View real-time images of the codes being read by your DataMan reader directly within the WebUI.
- Unified platform: The DataMan WebUI is consistent across all models, simplifying deployment by consolidating common controls into a single page.
- Real-time impact: Explore how different options affect the reader in real time.
- No additional downloads required: You no longer need to download and install additional software.
- Fast setup: Save time by not having to download and install software. Systems take just a few minutes to deploy.
- Programs: Programs provide a convenient method of grouping and storing multiple read setups and device-level parameters, enabling you to easily switch your reader between different applications.
- The WebUI offers the same functionality as the DataMan Setup Tool, with the following differences:
 - Settings changes are automatically saved in WebUI, while DataMan Setup Tool tool requires you to manually save settings.
 - The DataMan Setup Tool tool only provides a program-level view.

You can access the WebUI from the following browsers:

- Google Chrome
- Microsoft Edge
- Apple Safari
- Mozilla Firefox
- Opera

The following DataMan reader models are compatible with WebUI:

- DataMan 80 series
- DataMan 280 series
- DataMan 290 series
- DataMan 390 series

Connecting to WebUI

Perform the steps described to connect your DataMan reader to WebUI. These steps apply to both Ethernet and USB readers.

Note: USB readers always receive a random link local IPv4 address automatically to connect with.

Power on your reader and connect it to a PC. There are two ways of connecting your reader to the WebUI: using MAC address, serial number, host name, or IP address.

MAC address, Serial Number, or Host Name

1. Locate the MAC address and serial number on the sticker of your reader:



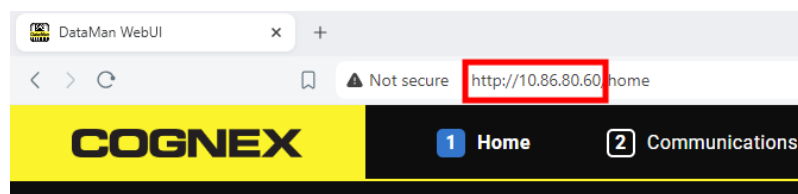
Alternatively, you can use the host name of the reader. By default, the host name of your reader is the model of your reader followed by the last six digits of the MAC address, using the following format: DM290-3BF53C.

2. Type in the MAC address, serial number, or host name into the web browser followed by `.local`. For example: `00D0243BF53C.local`.

Note: Make sure that your computer and your reader are on the same subnet, otherwise you cannot use the MAC address or serial number to connect to the reader from your computer. This restriction does not apply if you are using the host name of the reader to establish the connection.

IP address

1. Find the IP address of the reader in Station Manager:
 - a. Go to <https://support.cognex.com/en/products/station-manager> and download Station Manager.
 - b. Use Station Manager to view and find the IP address of all Cognex devices on the network. To connect to a specific reader, select it and click **Connect**. Your browser automatically opens the WebUI.
2. Type your IP address into the web browser: For example: `10.86.80.60`



Note:

i The WebUI uses a regular HTTP connection by default. For more information on how to set up a secure HTTPS connection, see [Certificates on page 98](#).

Troubleshooting

Setting your IP Address to DHCP

You must ensure that the Ethernet port of your computer is set to DHCP when connecting to a reader for the first time.

In Windows, go to **Network and Sharing Center > Your Ethernet Connection > Properties > Internet Protocol Version 4 (TCP/IPv4) > Obtain an IP Address Automatically**.

Using Static IP addresses

You can use the mDNS features of the WebUI on a static network.

Ensure that the reader and your computer are on the same subnetwork, and you can connect using the MAC address, serial number or host name to connect to the reader.

Connecting on a Managed Network

Managed networks with DNS servers behave differently than connecting the reader directly into your computer. Most internal corporate networks define different domains than `.local`, and have an internal DNS server that connects computers on the internal network through hostname.

.local Addresses do not Work

A managed network might define a different domain than `.local`. For example, an internal network may use `.computer.corporate.com` instead of `.local`. In this case you need to replace `.local` in all examples above with `.computer.corporate.com`.

To check the domain of your network:

1. Type in *About* in the search bar of your computer
2. Find Domain or Workgroup settings.
3. Verify that your local domain is listed.

Connecting through Device Name Works, but MAC and Serial do not

You can only connect using the Device Name on a managed network with a DNS server. This is because managed networks typically support a single non-IP address name for each device.

Connecting through USB-C

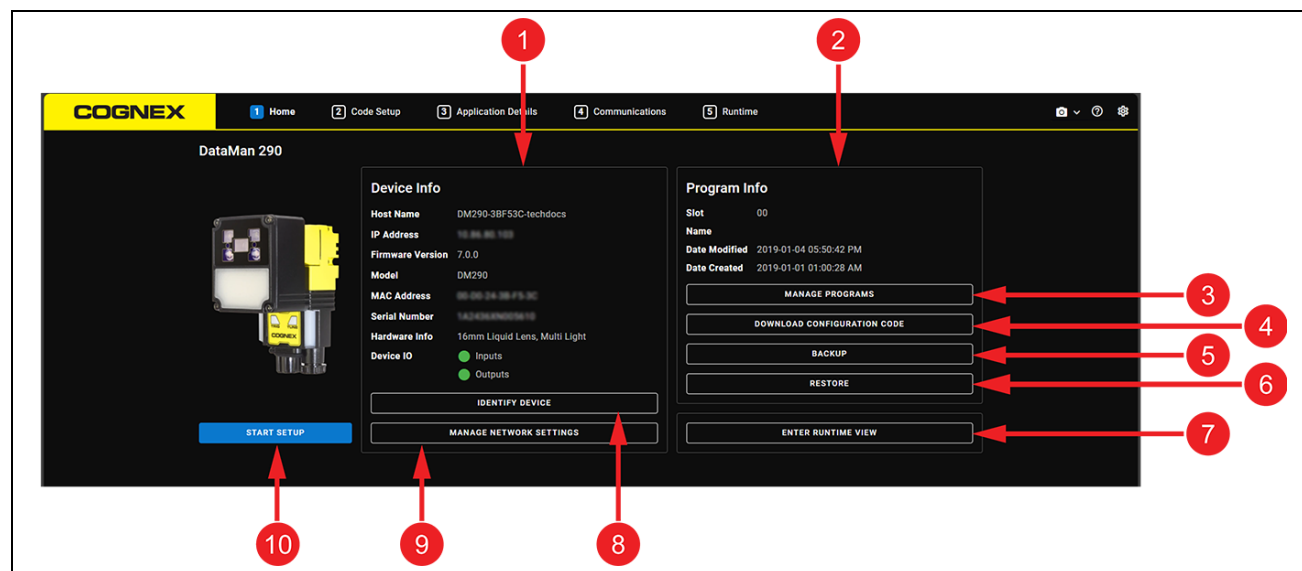
The WebUI does not support mDNS connections through USB-C.

Application Steps

WebUI provides a workflow that allows you to set up your reader through a series of guided steps.

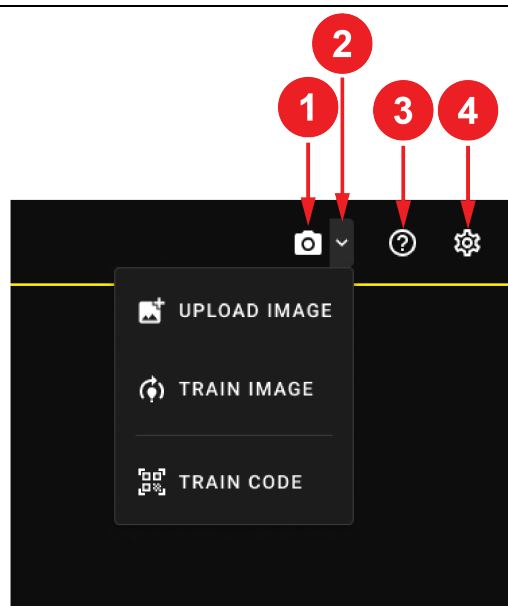
Home

The **Home** step provides an overview of your reader and serves as a central hub for reader management and configuration.



Number	Element
1	Information about the reader.
2	Information about the currently active program on the reader.
3	The Manage Programs button opens the window where you can manage the programs on the reader. For more information, see Program Management on page 87 .
4	The Download Configuration Code button downloads the current reader configuration as a series of 2D codes.
5	The Backup button opens the window to save a backup file.
6	The Restore button opens the window to upload the selected backup file.
7	The Enter Runtime View button takes you to the Runtime step, where you can see the reader reading barcodes in real time.
8	The Identify Device button activates the ring light indicator of the reader.
9	The Manage Network Settings button opens the connectivity settings panel.
10	The Start Setup button takes you to the Code Setup step, where you can perform the initial positioning and tuning for your application.

The top right corner has the following options:



Number	Element
1	The Trigger button manually triggers the reader.
2	The dropdown provides additional image options: - Click Upload Image to open the file browser, and select the image you want to upload for processing. - Click Train Image to open the file browser, and select the image you want to train the reader on without tuning. - Click Train Code to acquire an image and train the image without tuning.
3	The available help information, including page help and documentation: - Click Page Help to get additional information for the step that you are on. - Click Getting Started Guide to open an external website with extended help materials. - Click Launch Help Walkthrough to get an interactive overview of the given step.
4	The Settings panel. For more information, see Settings on page 81 .

Code Setup

The **Code Setup** step allows you to set up, test, and review the positioning and tuning of your reader to ensure that your reader settings are optimized for the code that you are trying to read. Follow the substeps in the menu on the left side to configure your reader.

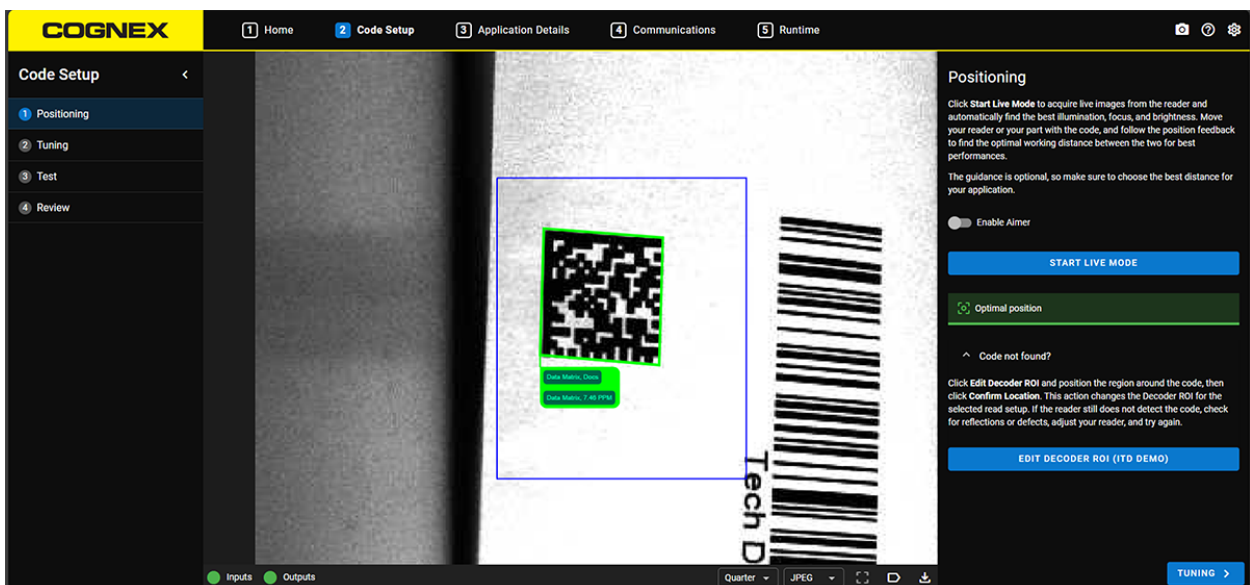
Positioning

The **Positioning** substep allows you to view live images from your reader, and read codes as you install and mount your reader.

The **Positioning** substep provides real time position guidance for the distance between the reader and the part, such as instructing you to move your reader closer to or further away from the part with the code.

Note: Position guidance is an optional guideline for finding the best possible decoding speed and consistency on challenging part markings. You can proceed with code setup regardless of your current position guidance.

You can trigger the reader manually using the **Trigger** button, or you can click **Start Live Mode** to automatically adjust the lighting, focus, and exposure for optimal results while you adjust the positioning.



Click **Start Live Mode** to acquire live images from the reader and automatically find the best illumination, focus, and brightness. Move your reader or your part with the code, and follow the position feedback to find the optimal working distance between the two for best performances.

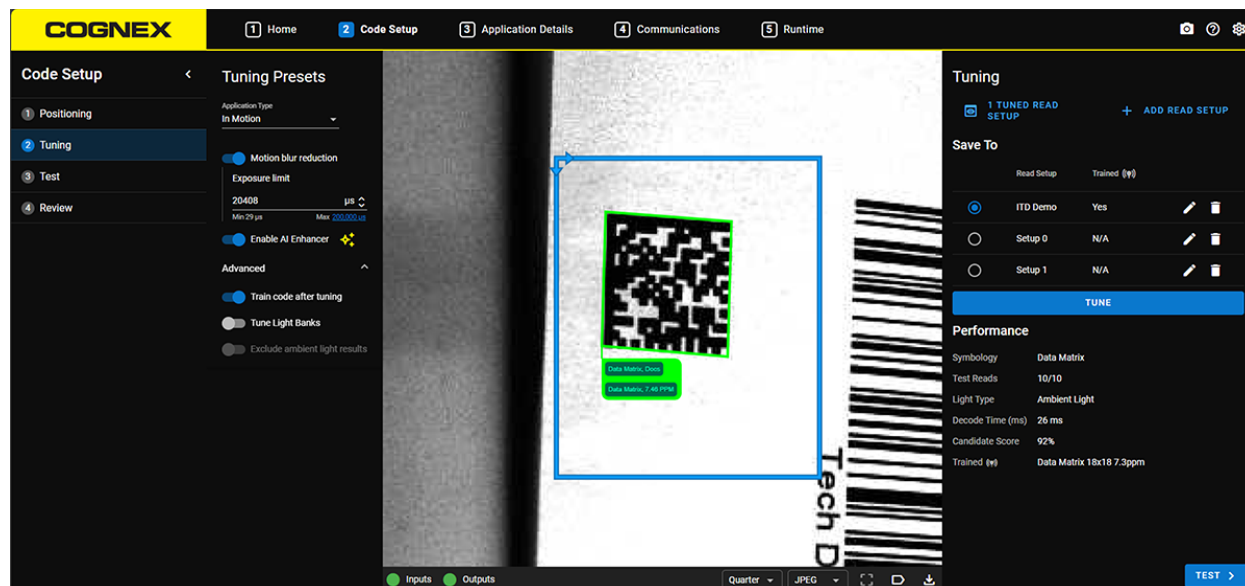
The guidance is optional, so make sure to choose the best distance for your application.

If the reader cannot find the code, open the **Code not found?** dropdown to mark the code manually.

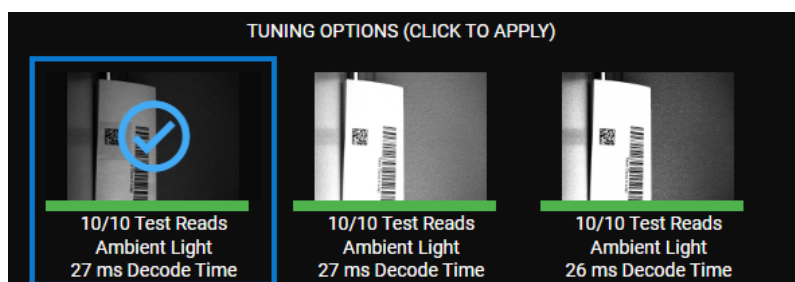
Click **Edit Decoder ROI** and position the region around the code, then click **Confirm Location**. This action changes the Decoder ROI for the selected read setup. If the reader still does not detect the code, check for reflections or defects, adjust your reader, and try again.

Tuning

The **Tuning** substep allows you to automatically find the best code reading settings for the reader.



Drag and drop the blue rectangle to define the decoder Region of Interest (ROI). The reader only decodes symbols that appear in the ROI. Click the **Tune** button to start the process of tuning the reader to the code. Once tuning is complete, WebUI displays the tuning options with the highest candidate scores, and the best candidate of the tuning options is automatically applied. You can switch to any of the other generated tuning options if you do not want to use the highest-ranked candidate.



The tuning procedure optimizes the following image acquisition parameters to obtain the best decode time and decoder performance:

- reader settings
- Light banks
- Lens Focus
- Image-filters
- Symbology attributes

You can also configure your tuning settings in the **Tuning Presets** column. Open the **Advanced** dropdown to access advanced parameters.

Select from the preset options to refine the tuning procedure:

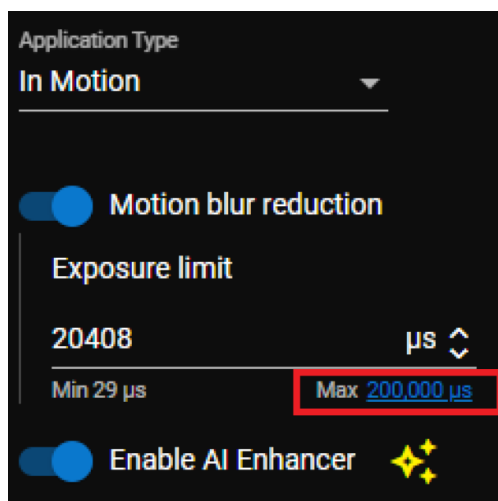
- **Application Type:** choose between **Stationary** or **In Motion**. If your application type is In Motion, you can reduce motion blur by properly adjusting the maximum exposure, either manually or using the embedded assistant.
- **Enable AI Enhancer:** use the AI Enhancer to improve decoding of damaged and degraded barcodes. The AI Enhancer uses artificial intelligence to enhance image regions which can contain a code to help improve overall decoding performance. The **Visibility** slider allows to dynamically overlay AI-adjusted regions based on the top three tuning candidates. AI Enhancer can help to increase code readability with only minimal impact to decoding speed.

Note: The DataMan 80 and DataMan 280 do not support the AI Enhancer.

- **Advanced options:**
 - **Train code after tuning:** enable this option if the codes you want to decode have a consistent symbology type and characteristics. Training the reader with the tuned symbology and its characteristics can decrease the time required to decode subsequent symbols, and help increase readability and read rate.
 - **Tune Light Banks:** enable to tune the reader illumination type.
 - **Exclude ambient light results:** the tuning procedure ignores images acquired without illumination.
 - **Field of View:** shows the current dimensions of the FoV and allows you to edit the FoV.

For **In Motion** type applications with **Motion blur reduction**, you can use the **Max Exposure Calculator** to calculate the recommended maximum exposure time:

1. Click on the **Suggested Max** number to open the **Max Exposure Calculator** dialog.



2. Enter the **Code Element Size** and **Max Line Speed** values.

The WebUI automatically calculates the recommended maximum exposure time.

3. Click **Apply** to use the calculated value as your **Exposure limit** for tuning.

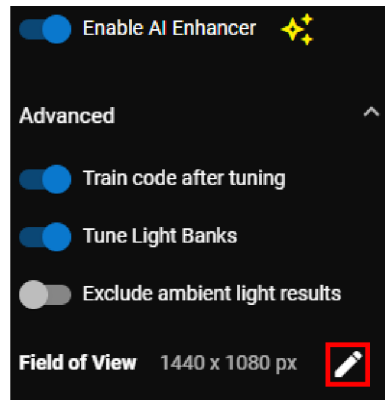
You can apply the tuning results to an existing read setup by selecting the target read setup in the **Save To** column. You can also click **Add Read Setup** to create a new read setup for the current tuning settings. You can use multiple read setups when there is wide variation between parts.

When you create a new read setup, you can copy the settings from an existing read setup, or you can perform tuning to save a new set of settings to the read setup.

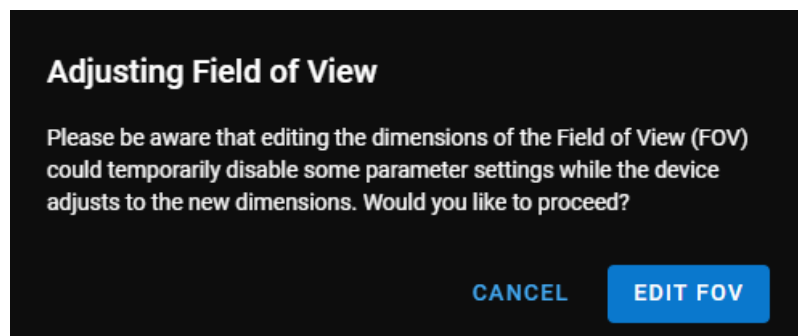
Adjusting the FoV

To resize the FoV, do the following:

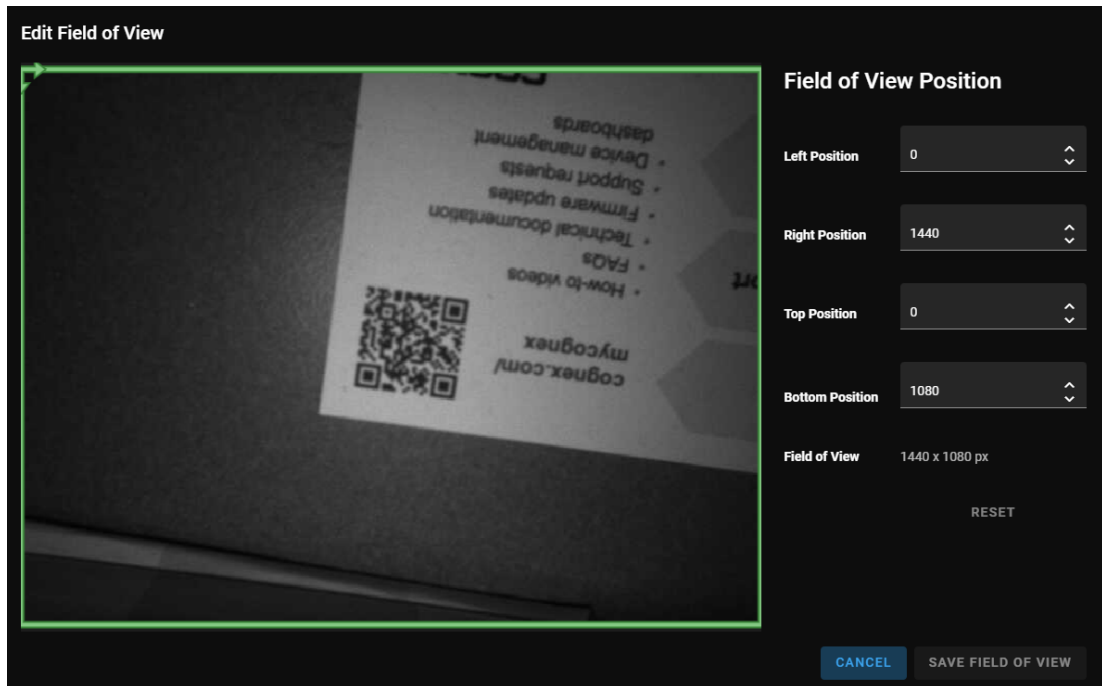
1. Click **Edit** next to the **Field of View**.



2. A dialog pops up informing you that the reader needs time to adjust to the new FoV settings, which means some of the functions are disabled during the adjustment process. Click **Edit FoV** to continue.



3. The **Edit Field of View** dialog pops up.



Edit the size of the FoV in one of the following ways:

- Drag the green rectangle to resize the FoV.
- Write values into the FoV positions.

The minimum value is 80 pixels for width and 64 pixels for height. For example, if you write 200 pixels into the **Left Position**, then you must write at least 280 pixels into the **Right Position**.

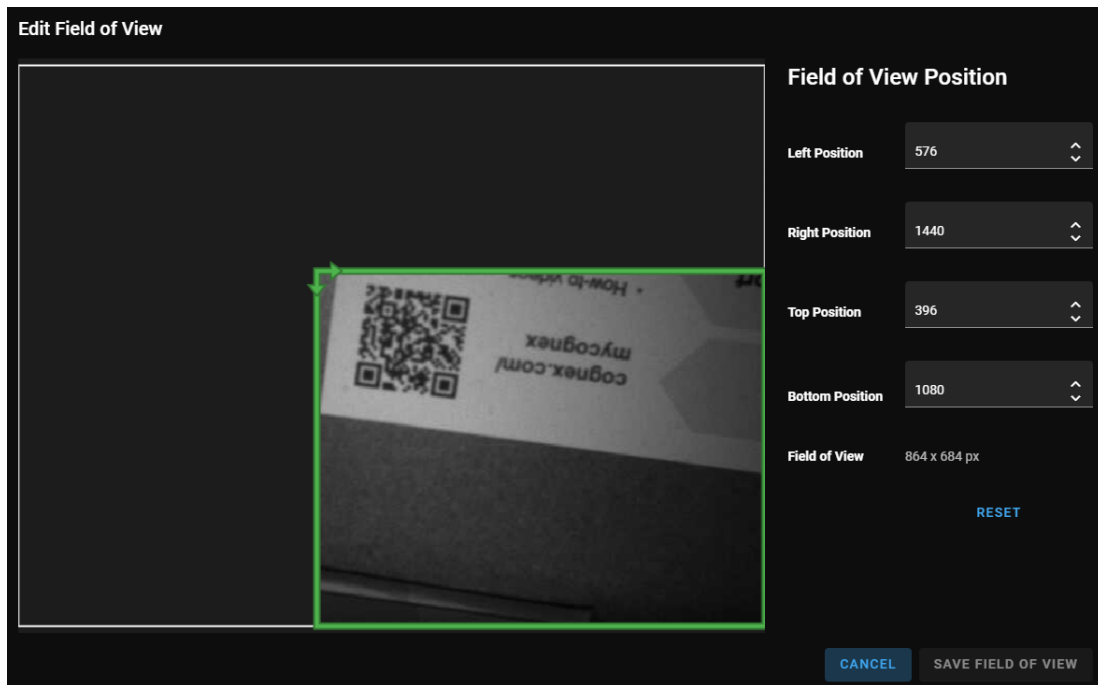
- Increase or decrease the position value by clicking on the arrows.

One click changes the value by eight pixels for width and four pixels for height. If you type in any number, the reader adjusts the value to the next lowest valid value. For example, if you type 100 pixels for the left position, the reader automatically adjusts the value to 96 pixels because 100 is not divisible by eight.

- Click **Reset** to reset the position values to the default values of the reader.

4. Click **Save Field of View** to finish. The reader acquires a new image with the new FoV.

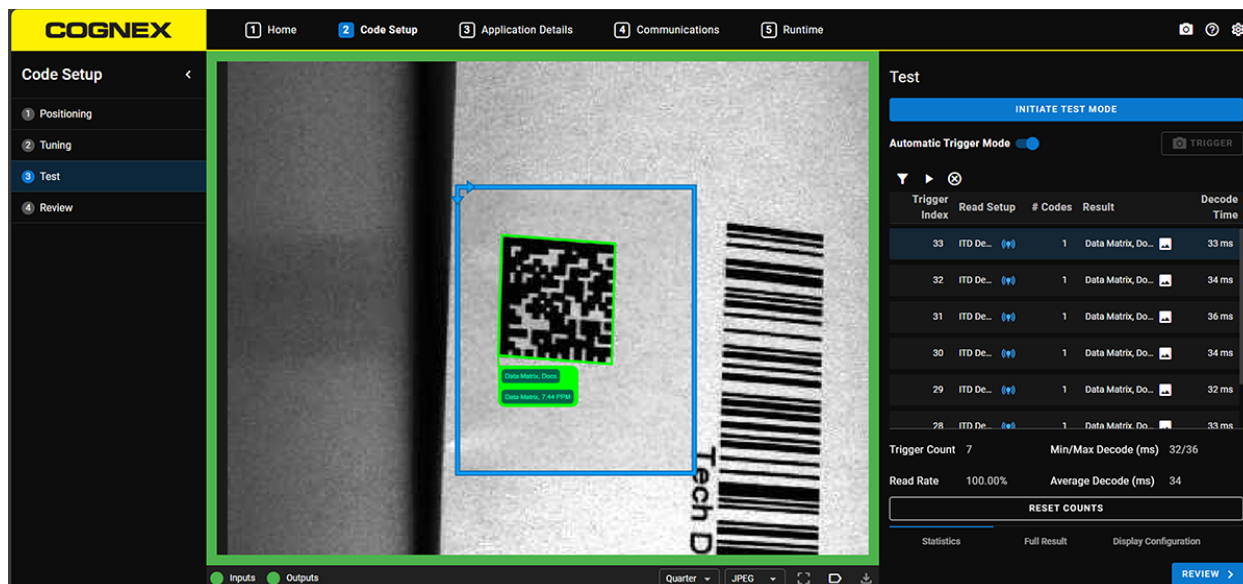
If you click **Edit Field of View** again, you can see the new FoV as the subregion of the full FoV, showing where the new FoV is compared to the full FoV.



Note: When clicking **Save Field of View**, tuning and live mode become temporarily unavailable. The decode time can change during the adjustment process if you trigger the reader. Once the reader is ready with the FoV adjustments, functions and speed resume back to standard.

Test

The **Test** substep allows you to test your tuning results without configuring a live trigger.



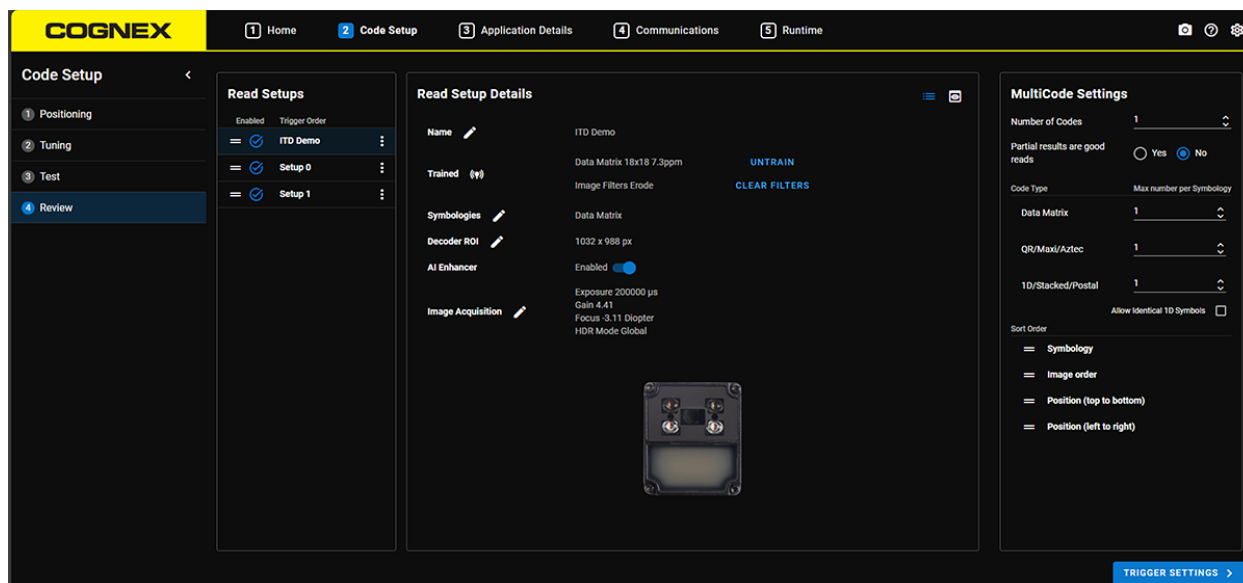
You can trigger the reader with the manual trigger button or switch to automatic mode. In **Automatic Trigger Mode**, the reader triggers as fast as possible.

Test mode automatically uses all enabled read setups. You can select the read setups you want to use for runtime in the **Review** substep.

Use the result history and statistics to determine if your current reader settings are suitable for your application.

Review

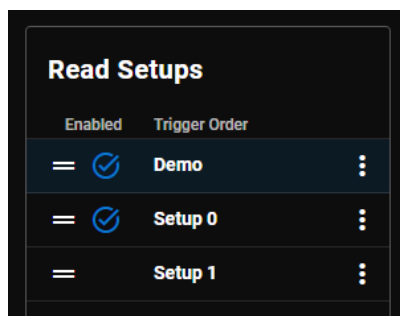
The **Review** substep allows you to view and modify your tuned settings, select which read setups to enable, delete read setups, or compare settings between them.



The **Review** substep provides a high-level overview of the tuning parameters, such as symbology, gain, and exposure for each read setup. You can also edit current read setups and configure **MultiCode Settings**.

A read setup is a set of image acquisition and decoding settings. The reader cycles through all enabled read setups until the trigger ends or the read result is complete. Use the **Read Setups** pane to enable or disable read setups, and change their **Trigger Order** by dragging the handles.

Manage all your configured read setups on the **Review** substep. The checkmark near a read setup means that this setup is currently enabled.



Click the three dots to the right of a read setup to open the read setup dropdown, where you can edit, compare, disable, or delete that setup.

You can check and configure each read setup individually. The settings shown in the **Read Setup Details** pane are based on the results of the **Tuning** substep. You can also view the image from which the tuned settings were derived, and compare the settings of two different read setups.

The following information is available for each read setup. Click on the pen icon to change a given setting:

- **Name:** The name of the currently selected read setup.
- **Trained:**
 - The type and size of the code that was trained during the Tuning step. Click **Untrain** to reset the trained code.
 - The filters that the Tuning procedure applies to the image to achieve the tuned image. Click **Clear Filters** to remove the filter.

Note: Use Setup Tool or DMCC commands to modify the training and filter configuration.

- **Symbologies:** The list of enabled symbologies that the reader recognizes.
- **Decoder ROI:** The Region of Interest (ROI) the decoder operates on.
- **AI Enhancer:** Enable or disable the use of AI enhancer.
- **Image Acquisition:**
 - **Exposure** and **Gain** affect the brightness of the image. Click **Optimize** to automatically determine good values.
 - **Focus** affects the distance where the image is the most clear. Click **Optimize** to automatically determine the best value.
 - **HDR Mode** affects whether HDR is enabled. This setting affects all read setups.
 - **High Frequency Light (HFL)** prevents the reader from rapidly flashing. In combination with the **Enabled** option, the **Always On** setting turns on the HFL at the initial trigger and maintains it after a trigger ends. You can set the **Timeout** measured in seconds after which the HFL turns off if no trigger occurs for a long period of time. This setting affects all read setups.
 - Configure which light bank of the reader is on by clicking on the reader image and selecting a light bank.

Use the **MultiCode Settings** pane to configure how to read multiple codes in one image. Determine the overall number of codes to read and the number of codes for each symbology. If you disable the **Partial results are good reads**, the reader does not return any codes, including single code images when the image contains fewer codes than specified with **Number of Codes**. If you enable **Partial results are good reads**, the reader returns a successful read result with fewer codes. The **Sort Order** list allows you to determine how the reader reports multiple results:

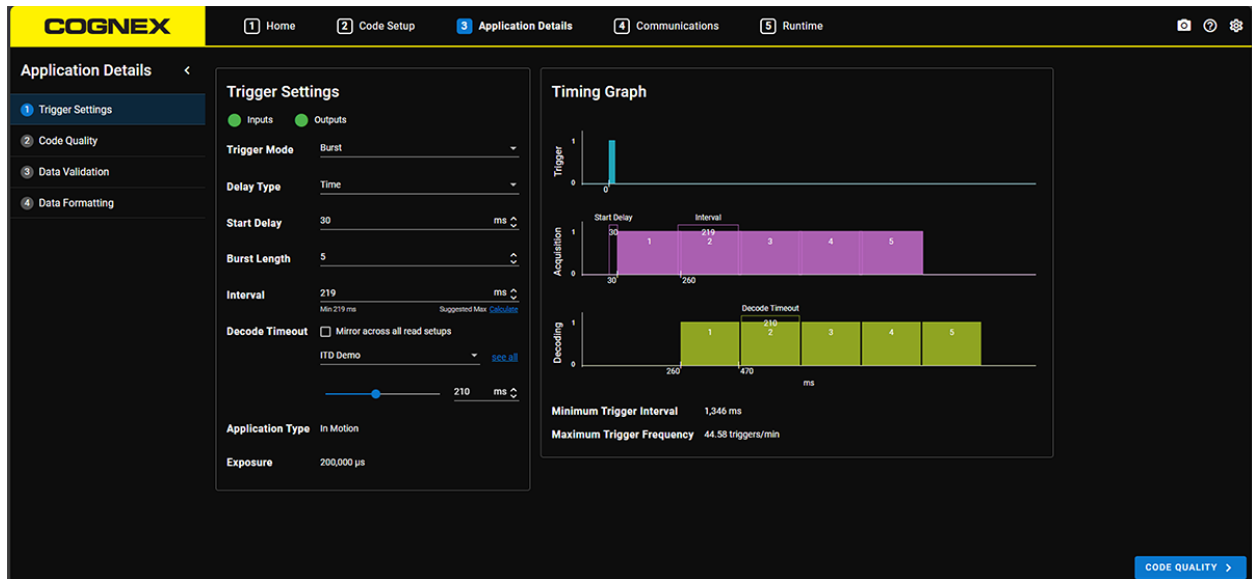
- Use the handles to change the priority order.
- The **Position (top to bottom)** and **Position (left to right)** options refer to the center of the symbols.
- Use **Image order** for Burst or Continuous trigger modes, where symbols that are decoded in earlier images are reported before symbols decoded in later images.

Application Details

The **Application Details** step allows you to configure the details of your application. Follow the substeps in the menu on the left side to configure your application.

Trigger Settings

The **Trigger Settings** substep allows you to configure your runtime processing and decoder timeout settings. This substep features a timing graph that reflects the trigger settings to give feedback on the current application setup.



Triggers that initiate a decode attempt either come from a source external to the reader or are generated by a timing mechanism internal to the reader itself.

When the reader is configured to respond to an external trigger, any of the following events signal the reader to perform a decode attempt:

- Pushing the physical button on the reader
- Sending the DataMan Control Command (DMCC) **trigger** when connected to the reader over a serial or Ethernet connection
- Receiving a digital input signal over the trigger lines of the I/O cable
- Clicking the **Manual Trigger** in WebUI

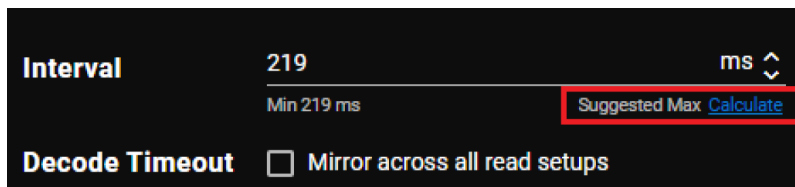
The following trigger types are available:

- **Self** trigger mode continuously acquires images at a configurable interval, and attempts to decode symbols found.
- **Single** trigger mode acquires a single image and attempts to decode any symbols found.
- **Burst** trigger mode acquires a set number of images and decodes until the expected codes are found. You can configure the number of images within each burst and the interval between each image acquisition. This trigger mode supports a read timeout. This timeout is the maximum amount of time spent on each image in the burst, and not on the burst as a whole.

- **Continuous** trigger mode acquires images as long as the trigger signal remains active, where the reader acquires images at a specific interval and attempts to scan any symbols each successive image contains. If the desired number of codes is decoded, the reader ends the trigger and sends out results. Until the desired number of codes is decoded, image acquisition continues as long as the trigger signal remains active. If the trigger signal ends and the desired number of codes is not decoded, the reader returns a NoRead result.

For **Burst** and **Continuous** trigger modes, you can use the **Max Interval Calculator** to calculate the recommended maximum interval:

1. Click **Calculate** to open the **Max Interval Calculator** dialog.



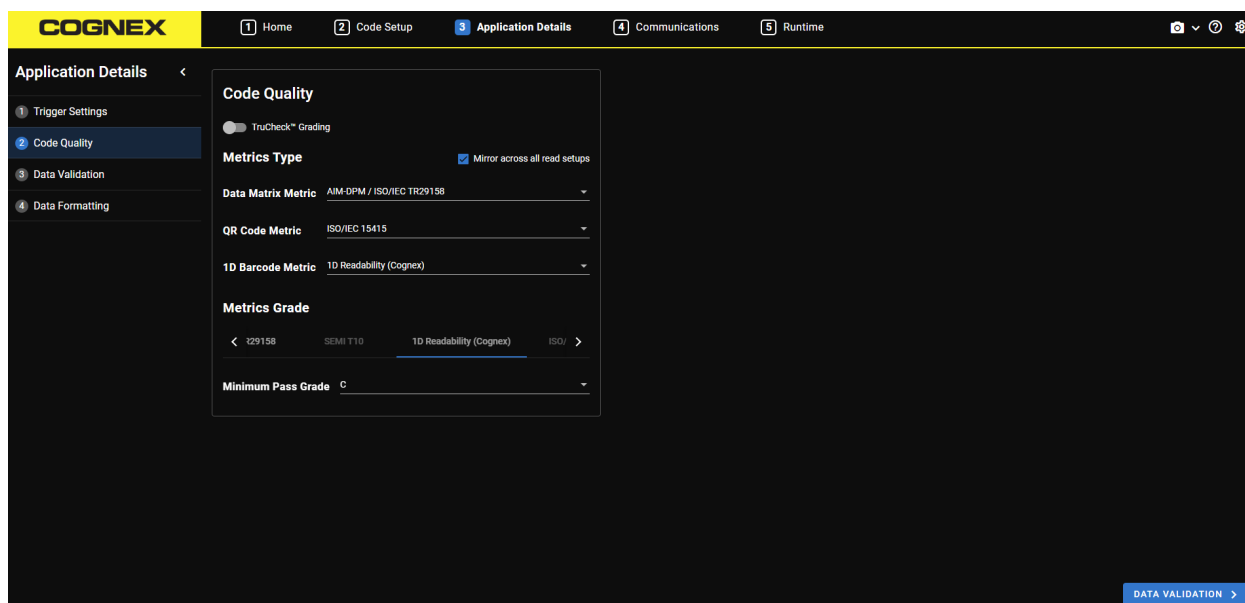
2. Choose your **Application Type** to display different options for calculating the necessary data:
 - Select **Field of View Size** to calculate the longest possible interval time by measuring the physical **Field of View Size** and giving the **Maximum Line Speed** and the size of the **Longest Code**.
 - Select **Lens Distance to Code** if you want to give the distance of the code from the lens. Give the **Lens Distance** and select the **Motion Direction** in which the code is traveling. Here you also need to give the **Max Line Speed** and the size of the **Longest Code**.
 - Select **Code Element Size** to calculate with the size of the most narrow code element. Press the **Trigger** button to read your code and obtain the **Code PPM** value. You also need to give the **Motion Direction** in which your code is traveling in, the **Max Line Speed** and the size of the **Longest Code**.

The WebUI automatically calculates the recommended maximum interval time when you fill in all the required fields.

3. Click **Confirm** to use the calculated value as your maximum **Interval** time.

Code Quality

The **Code Quality** substep allows you to validate that the symbols you use meet the quality guidelines as outlined in one of the code quality standards supported by Cognex. Codes get an overall grade after grading all or selected metrics for a given code quality standard.



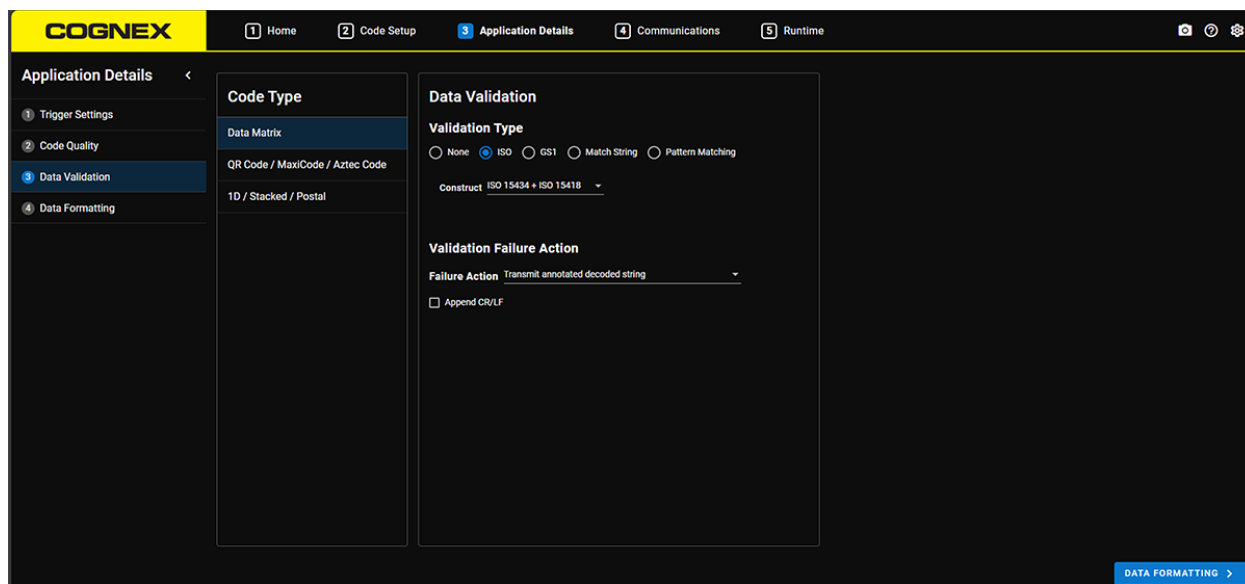
For more detailed information on specific code quality standards and their metrics, see the *DataMan Communications and Programming Guide* or the corresponding ISO standard documentation, which can be found at www.iso.org.

Configure a grade threshold for acceptable code quality with the **Minimum Pass Grade** toggle.

To enable Standards-Based Grading, enable the **TruCheck™ Grading** toggle. For more information, see [Configuring Grading on page 68](#).

Data Validation

The **Data Validation** substep allows you to confirm that the data encoded by a symbol is in the correct format for a particular company, industry, or international standard. The settings shown on the page depend on the **Validation Type** you choose.



Select the **Code Type** and assign a **Validation Type** to it.

- Use the **None** option if there is no need for any validation type.
- Use the **ISO** option to specify an ISO standard for the strings.

- Use the **GS1** to specify an application identifier for the strings.
 - In addition to the list of specific AIs that you can select, the following special tokens are available:
 - <AI(.)> to signify AI
 - <AI(*)> for an arbitrary number of AIs
 - DataMan software supports GS1 Digital Link but only for syntax checking. With GS1 Digital Link enabled, the reader ignores any specified pattern, as if the pattern was set to <AI(*)>.
- Use the **Match String** option to specify an exact string to match against the string encoded by the symbol. Only symbols containing a string that matches the specified string generate a **Pass** result.
- Use the **Pattern Matching** option to input a regular expression to confirm if the data string includes a specific string of characters.

 **Note:** The **Match String** function is case sensitive.

If a code fails validation, you can configure the reader to output a specific string in the **Outputs** substep.

Choose one the following options for the Validation Failure Action:

- **Transmit "Validation Failure":** The reader outputs the string *"Validation Failure"* after a code fails validation.
- **Transmit "Validation Failure" + decoded string:** The reader outputs the string *"Validation Failure"* followed by the complete decoded string.
- **Transmit "Validation Failure" + quoted decoded string:** The reader outputs the string *"Validation Failure"* followed by the quoted part of the decoded string.
- **Transmit annotated decoded string:** The reader outputs the decoded string along with information on the error type and error location in the string.
- **Transmit nothing:** The reader outputs no information when data validation fails.

Data Formatting

In the **Data Formatting** step, you can construct a customized output string using pre-defined building blocks. You can also test your output before proceeding to the next step.

The screenshot shows the COGNEX application interface. The top navigation bar includes 'Home', 'Code Setup', 'Application Details', 'Communications', and 'Runtime'. The 'Application Details' sidebar is active, showing 'Trigger Settings', 'Code Quality', 'Data Validation', and 'Data Formatting'. The 'Data Formatting' section is expanded, showing a 'Formatting' panel. This panel includes an 'ADD DATA RESULT' button, a table with 'Data Result' and 'Value' columns, a 'Delimiter' dropdown menu, and fields for 'Leading Characters', 'Trailing Characters', and 'No Read String'. An 'Example Output' field displays 'Data Matrix, Docs'. A 'COMMUNICATIONS' button is visible in the bottom right corner.

The output string starts with the text you enter in the **Leading Character** field. It continues with the **Data Results** you add, in the order you define. Each item is separated by the **Delimiter**. The output string ends with the text in the **Trailing Character** field.

To construct the output string:

1. Click **Add Data Result** to add a new item to the data formatting.
2. Reorder your data by dragging items to their new position.
3. Configure data formatting options that apply to the whole output string.
4. Open the **Delimiter** dropdown and select what string delimiter you want to use.
5. In the **String Options** section, define the leading characters, trailing characters, and No Read string you want to use.
6. Make sure to review your final output in the **Example Output** field before moving on to the next step.

You can configure options that impact that overall output string, such as specifying the **Leading Character** and the **Trailing Character**. The reader transmits the **No Read String** if it cannot read a given code. This can be useful for verifying that the reason you do not see any output is because the reader did not read data.

You have to configure an SNTP server to add timestamps to the output string.

Data items to construct your output string:

Category	Data	Description
----------	------	-------------

Result	Full string	All of the characters encoded in the symbol.
	Sub-string	A sub-string of the characters encoded in the symbol. You must specify a range for the sub-string when using this data. In Simple mode, you have to provide the index of starting and ending characters to construct the sub-string. In Advanced mode, you can provide a range, such as <i>1-3,5,8,10-</i> , for selecting the characters from the full string.
	Symbology	The symbology of the decoded symbol.
Formatting	Space	A space character.
	CR/LF	Carriage Return with Line Feed.
	Tab	A tab character.
Result Details	Symbology Identifier	The symbology identifier as specified by ISO/IEC 15424:2008.
	Symbol Size	The size of the symbol in pixels.
	Module Size	The width or height, in pixels, of each module in the symbol.
	Code Position	The image coordinates of each corner of the decoded symbol.
	Code Center	The image coordinates of the center of the decoded symbol.
	Code Orientation	The orientation of the decoded symbol reported in degrees. Clockwise rotations of the decoded symbol are indicated as positive changes in orientation.
	Contrast	A value, between 0 and 1, to indicate the total amount of contrast within the decoded symbol. A high value indicates the light and dark modules in the symbol have a wide range of grey values, while a low value indicates the light and dark modules consist of similar grey values.
	Unused Error Correction	The amount of available unused error correction in the symbol. Error correction is a method of reconstructing or replacing data that is lost through symbol damage. A value of <i>100</i> is ideal. A 1D symbol always returns <i>-1</i> .

TruCheck™	TruCheck™ Application Standard Name	The name of the application standard that is used for grading the code.
	TruCheck™ Application Standard Result	The result based on the application standard.
	TruCheck™ Grading Standard Name	The name of the grading standard.
	TruCheck™ Overall Grade Letter	The letter of the overall grade, for example, A.
	TruCheck™ Overall Grade Value	The value of the overall grade, for example, 3.9.
	TruCheck™ XDimension (mil)	The value of the custom X dimension in mils.
Metadata	Input string	The received serial or Ethernet input string.
	Passed Validations	The number of validation attempts the decoded symbol passed.
	Failed Validations	The number of validation attempts the decoded symbol failed.
	Result source	The name of the reader.
	Code Quality Grade	The grade that the code receives during validation.
Timing	Decode time	The amount of time, in milliseconds, it took the reader to decode the symbol. This period of time does not include image acquisition, pending image processing, communication time and additional processing such as data formatting.
	Trigger time	The amount of time, in milliseconds, that elapses between receiving the trigger and completing the decode operation.
	Filter time	The duration of filtering in milliseconds.

Timestamp	Trigger Creation Local Time	The local time when the reader created the trigger.
	Trigger Creation UTC Time	The UTC time when the reader created the trigger.
	Trigger Execution Local Time	The local time when the reader executed the trigger.
	Trigger Execution UTC Time	The UTC time when the reader executed the trigger.
	Image Acquisition Local Time	The local time when the reader acquired the image of the symbol.
	Image Acquisition UTC Time	The UTC time when the reader acquired the image of the symbol.
	Subresult Creation Local Time	The local time when the reader created the subresult.
	Subresult Creation UTC Time	The UTC time when the reader created the subresult.
	Current Local Time	The current local time.
	Current UTC Time	The current UTC time.

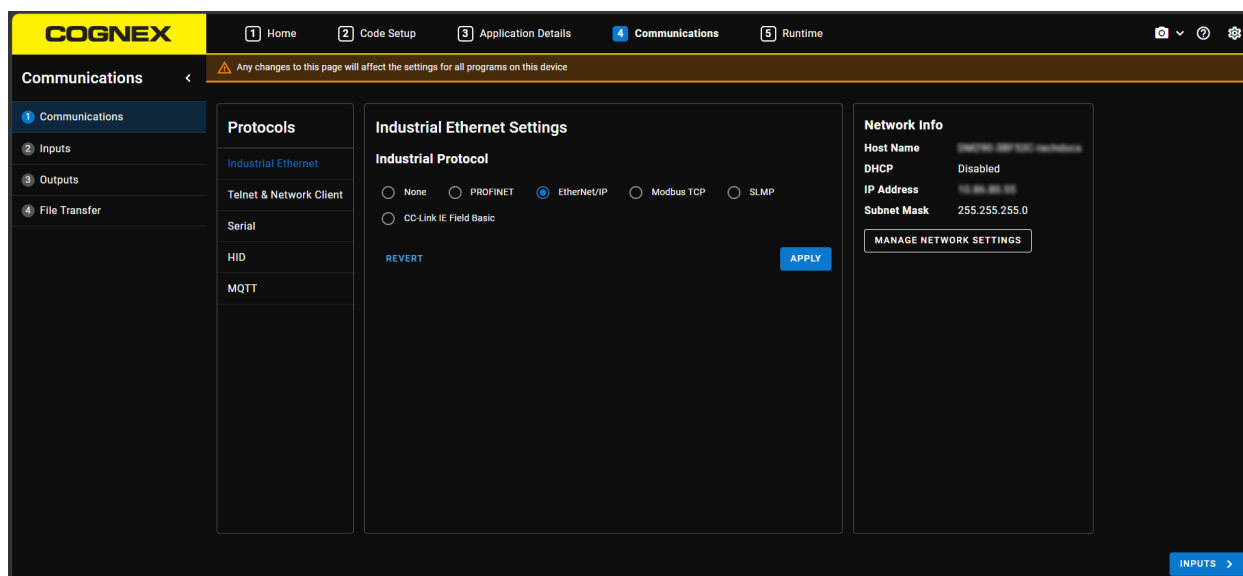
Communications

The **Communications** step allows you to configure the network, I/O, and file transfer settings for your reader. Follow the substeps in the menu on the left side to configure your communications settings.

Note: Any changes you make on the **Communications** step are applied to all programs on the reader.

Communications

The **Communications** step allows you to configure the communication protocols of your reader.



Tip: Additional communications protocols are available through the DataMan Setup Tool.

Industrial Ethernet

Industrial protocols enable the reader to exchange I/O data, alarms, and diagnostics with a PLC. Configure the industrial protocol you want to use:

Protocol	PLC
PROFINET	Siemens S7
EtherNet/IP	Rockwell ControlLogix
SLMP	Mitsubishi Q, iQ and L series

To select the protocol, click **Apply**. To disable a protocol, click **Revert**. This reverts the settings to the previous state. When changing industrial protocol settings, you have to restart the reader for the changes to take effect.

For more information, see the *Industrial Protocols Manual*.

Telnet & Network Client

Configure the **Telnet Port** to be used for Telnet communication.

Disable the **Telnet Port** to stop sending data from the reader through Telnet port. Disabling Telnet push messages still supports DMCC communication through the Telnet port.

The **Network Client** behavior is a means for the reader to actively open an Ethernet connection to another device on the network. The connection has the same basic operational behavior as a Telnet connection: the reader can transmit results to the external device and process DMCC commands received from the external device.

To use the reader as a network client, enable the **Network Client** toggle and provide the **Host Address**.

Serial Settings

Set the values to match the reader serial settings to the device you are connecting to:

- Baud Rate
- Data Bits
- Stop Bits
- Parity
- Handshake

HID Settings

Turn on the **Enable HID interface** toggle.

- **Keyboard language:** Set the language. For Alt Key Codes, you can enter special characters using the **Alt** key and the numeric keypad. The read results are then interpreted as numeric keystrokes.
- **Inter-character delay:** Set the delay time between characters when in HID keyboard mode.
- **Keyboard report interval:** Set the delay between key down and key up HID messages.

MQTT

MQTT allows you to forward read results and images from your DataMan reader.

By default, the Enterprise Connectivity service is disabled. Enable the service using the **Enable Enterprise Connectivity** toggle to display the configuration options:

- The **Connections** tab allows you to configure MQTT connections
- The **Attributes** tab allows you to specify additional device attributes to include in your MQTT messages, which you can reference in your output format or connection properties.
- The **Diagnostics** tab allows you to troubleshoot and verify MQTT connections. When the **Enable Enterprise Connectivity** toggle is on and you have at least one connection set up, the **Diagnostics** tab displays diagnostic messages in real time.

For more information on how to set up an MQTT connection, see [Configure MQTT Connection on page 30](#).

Configure MQTT Connection

This section describes the configuration procedure for MQTT connections.

Consider the following before setting up MQTT:

- On readers with multiple Ethernet ports, you can use either port for MQTT communication.
- You can use the same reader port for MQTT and industrial protocols.

Set Up MQTT Broker and Client on Your PC

Note: You can skip this section if you already have a broker that you can connect to.

Before you can set up MQTT forwarding on your reader, you need to set up an MQTT broker and client on the PC where you intend to receive MQTT messages. The following procedure provides an example setup for the broker and client applications, but any application that supports MQTT v3.1.1 or v5 is compatible.

1. Install the latest version of [mosquitto for Windows](#) to use as an MQTT broker, only checking the **Service** option when prompted during the installation.
2. Open the Windows command prompt and execute the following command to put mosquitto on your path:

```
set "PATH=%PATH%;C:\Program Files\mosquitto"
```

3. Create a new text file called *my_mosquitto.conf* in the mosquitto directory, and add the following two lines to the file:

```
allow_anonymous true
listener <port-number> <your-machines-ip-address>
```

4. Start the mosquitto application with the following command:

```
&> mosquitto -c .\my_mosquitto.conf -v
```

5. Install the latest version of [MQTTX for Windows](#) to use as an MQTT client.
6. Start the MQTTX application.
7. Connect your MQTTX application to the mosquitto broker that you set up using the network IP address and port number that you defined in the *my_mosquitto.conf* file.
8. Create a new subscription:
 - Subscribe to **#** to show all topics in the client window.
 - Subscribe to a specific device to filter out messages originating from that reader. For example, **cognex/<device host name>/#**.

Set Up Certificates

If you want to establish secure communication over MQTT, you have to install the necessary certificate on your DataMan reader before configuring your connection. To install your certificate, use the [Certificates on page 98](#) setting option.

You can install the following types of certificates:

- **TLS:** allows other devices to trust your DataMan reader:
 - WebUI (HTTPS)
 - Enterprise Connectivity (MQTT with mTLS)

- **Trusted (Certificate Authority):** The self-signed certificate of the Certificate Authority (CA). It allows your DataMan reader to trust a server configured with a certificate signed by the same Certificate Authority:
 - Enterprise Connectivity (MQTT with TLS)

 **Note:** The reader ships with the pre-installed TLS certificate file *default-tls.crt*. However, Cognex recommends installing your own certificate.

Configure MQTT Connection on the WebUI

Once you have the MQTT broker and client set up, and you have installed the necessary certificates, if any, you can configure the MQTT connection for your DataMan reader.

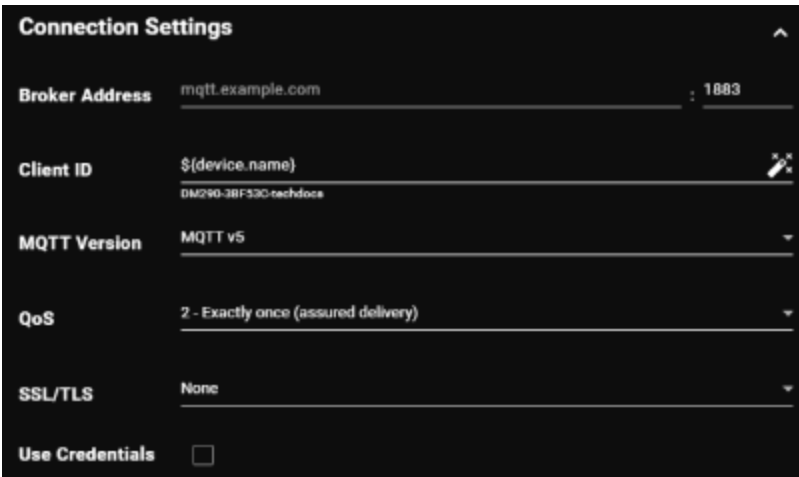
1. Launch the WebUI, navigate to the **Communications** substep, and select **MQTT**.
2. Turn on the **Enable Enterprise Connectivity Service** toggle to enable the service. Otherwise the pane does not display any other settings.

The **MQTT Connection** configuration pane appears on the **Connections** tab.

3. In the **Connection Settings** section, enter the following information:

Connection Settings

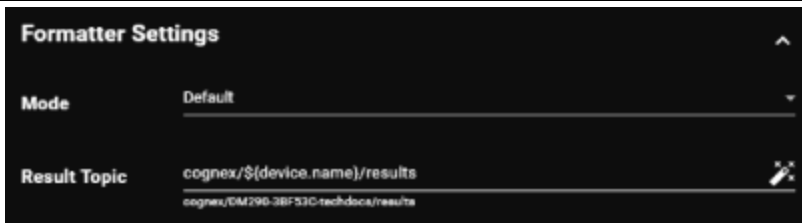
Parameter	Description
Broker Address	Enter the hostname or IP address of your MQTT broker, followed by the port number. MQTT connections use the following default ports: 1883 for MQTT 8883 for MQTTS
Client ID	Define the unique identifier for your DataMan reader on the MQTT broker. The broker uses the client ID to track client connections. A default client ID is assigned automatically based on the serial number of your reader.
MQTT Version	Select the MQTT version of your broker: - MQTT v3.1.1 - MQTT v5
QoS	The Quality of Service level for the MQTT connection to determine delivery guarantees: 0 - At most once (fire and forget) 1 - At least once (acknowledged delivery) 2 - Exactly once (assured delivery)
SSL/TLS	Select the security protocol you want to use for certification: None TLS - requires a Trusted Certificate Authority (CA) to be uploaded. mTLS - requires a certificate for the reader that is signed by a CA that the MQTT broker trusts.  Note: If you select TLS or mTLS, make sure you have a valid certificate installed using the Certificates on page 98 setting option.

	
Parameter	Description
Certificate	Select the certificate you want to use. For more information on certificate management, see Certificates on page 98 .
Use Credentials	Enable or disable the use of credentials when attempting to establish the MQTT connection. This is necessary when the MQTT broker requires authentication for communication. If you check the Use Credentials checkbox, the following additional fields appear: Username Enter a valid user name for the MQTT broker. Password Enter the password for the user name.

4. Open the **Formatter Settings** dropdown to configure MQTT formatting:

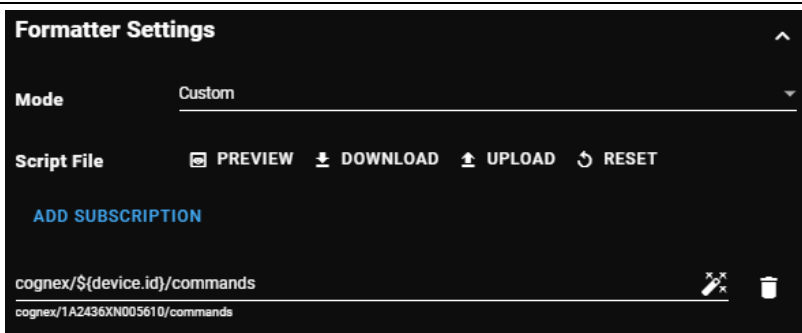
- If you select the **Default** option for the **Mode** setting, your DataMan reader uses the default script provided by Cognex. You can modify the following additional settings:

Formatter Settings - Default Mode

	
Parameter	Description
Result Topic	Defines the MQTT topic for publishing read results. A default result topic is assigned automatically based on the client ID.

- If you select the **Custom** option in the **Mode** setting, you can upload your own custom script file to define the MQTT format. You can also click the **Add Subscription** button to add custom MQTT subscriptions.

Formatter Settings - Custom Mode

	
Parameter	Description
Script File	View, upload, download, or reset your custom formatter script: Preview - open the current script in your browser. Download - save the current script. Upload - load a script from your computer. Reset - clear and use the default script.
Add Subscription	Add any number of MQTT topics to listen to for incoming messages. Each subscription allows your script to react to messages on that topic.

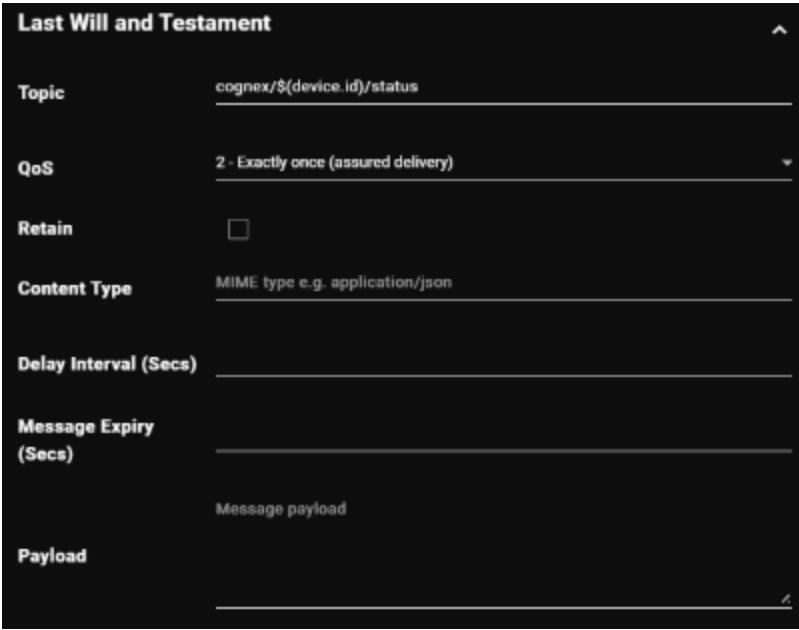
5. Open the **Image Settings** dropdown to enable and configure image transfer within MQTT messages:

Image Settings

Image Settings ^ Enable ☒ Include ☒ OK ☒ NG Graphics ☒ None	
Parameter	Description
Enable	Enables or disables MQTT image transfer. i Note: When disabled, all other settings in the Image Settings dropdown are hidden.
Include	Enables or disables the transfer of images for specific inspection results: OK Enables or disables image transfer for OK results. NG Enables or disables image transfer for NG results.
Graphics	The WebUI does not support sending SVG overlay images over MQTT.

6. Open the **Last Will and Testament (LWT)** dropdown to configure any final messages in case of an unexpected disconnection:

Last Will and Testament Settings

	
Parameter	Description
Topic	The MQTT topic where the LWT message is published if the reader disconnects unexpectedly.
QoS	The Quality of Service level for the LWT message to determine delivery guarantees: 0 - At most once (fire and forget) 1 - At least once (acknowledged delivery) 2 - Exactly once (assured delivery)
Retain	If enabled, the broker retains the LWT message on the topic so that new subscribers receive it immediately upon subscribing.
Clear On Connect	If enabled, the reader sends an empty message to the topic when it connects, clearing any retained LWT message on the topic.
Content Type	Defines the MIME type to describe the format of the message payload, such as <code>application/json</code> or <code>text/plain</code> . <i>Note:</i> This setting is only available when using MQTT version 5.0.
Delay Interval (Secs)	Defines the delay in seconds between the disconnection and the publishing of the LWT message. <i>Note:</i> This setting is only available when using MQTT version 5.0.

Last Will and Testament
⬆

Topic

cognex/\${device.id}/status

QoS

2 - Exactly once (assured delivery) ⬇

Retain

☐

Content Type

MIME type e.g. application/json

Delay Interval (Secs)

Message Expiry (Secs)

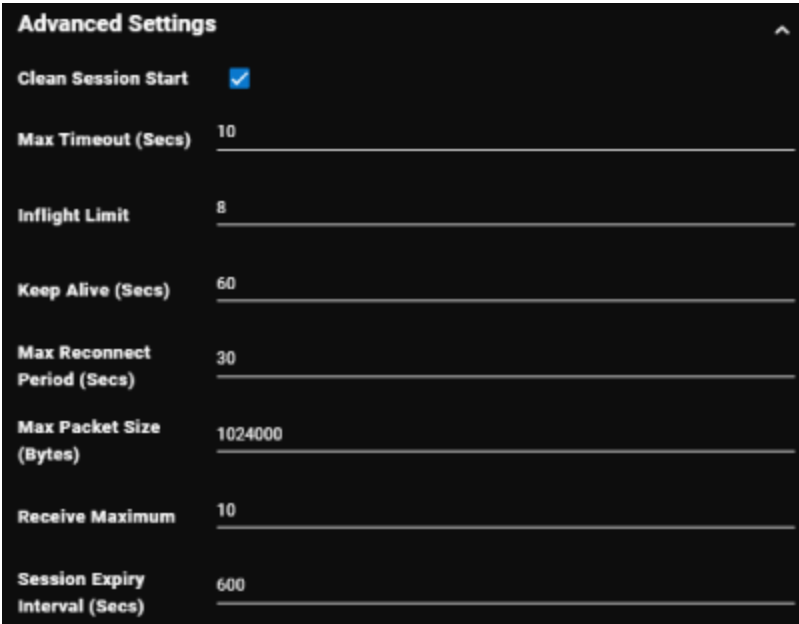
Payload

Message payload

Parameter	Description
Message Expiry (Secs)	Defines the expiration of the LWT message in seconds, after which the broker removes the message. ⓘ Note: This setting is only available when using MQTT version 5.0.
Payload	Defines the content of the LWT message. This can be a plain text string or a JSON object, depending on the application.

7. Open the **Advanced Settings** dropdown to fine-tune the MQTT connection for reliability, performance, and compatibility with your broker and network environment:

Advanced Settings

	
Parameter	Description
Clean Session Start	If enabled, the reader starts each connection with a clean session, clearing any previous subscriptions and pending messages.
Max Timeout (Secs)	Defines the maximum interval in seconds to wait before abandoning a non-responsive connection.
Inflight Limit	Defines the maximum number of unacknowledged messages that are allowed to be sent at once. This helps control message flow and reader load.
Keep Alive (Secs)	Defines the interval in seconds for sending keep-alive pings to the broker to maintain the connection.
Max Reconnect Period (Secs)	Defines the maximum waiting period in seconds between automatic reconnection attempts after a disconnect.
Max Packet Size (Bytes)	Defines the maximum allowed size in bytes for outgoing MQTT packets. This prevents sending oversized messages.
Receive Maximum	Defines the maximum number of QoS 1 and QoS 2 messages that the reader processes concurrently.
Session Expiry Interval (Secs)	Defines the time in seconds after which the MQTT session expires when the reader has disconnected. This controls how long the broker retains subscriptions and pending messages.
Accept Invalid Certificate	If enabled, this setting allows connections to brokers with invalid or expired SSL/TLS certificates.

Advanced Settings

Clean Session Start

☒

Max Timeout (Secs)

10

Inflight Limit

8

Keep Alive (Secs)

60

Max Reconnect Period (Secs)

30

Max Packet Size (Bytes)

1024000

Receive Maximum

10

Session Expiry Interval (Secs)

600

Parameter	Description
Accept Invalid Hostname	If enabled, this setting allows connections even if the hostname of the broker does not match on the certificate.
Max Heap Size (MB)	Defines the maximum heap size allocated to the MQTT script.
Max Execution Time (ms)	Defines the maximum time the MQTT script can run.
Concurrency Limit	Defines the number of allowed inflight script promises.

8. Click **Create** to apply your changes.

You now have an MQTT connection configured:

MQTT

☒ Enable Enterprise Connectivity

Connections

Attributes

Diagnostics

+ ADD CONNECTION


mqtt - testtopic1234

Last successful upload: -

Results Processed 0

Incoming Messages 0

Publish OK 0

Publish Pending 0

Publish Errors 0

Publish Overflowed 0

Results Overflowed 0

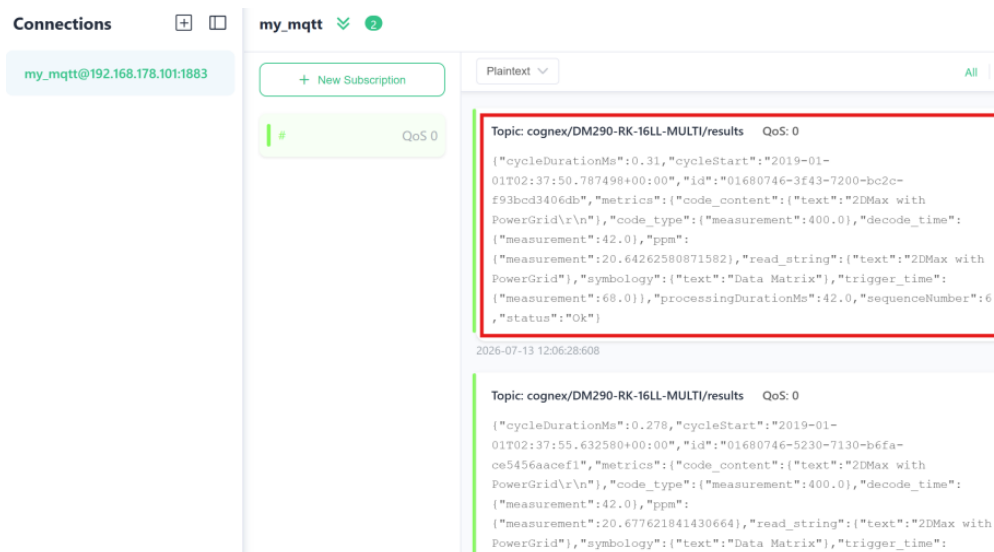
Results Skipped 0

RESET STATISTICS

Monitor the Published Data

Once you have your MQTT connection set up, you can view the read results and images the reader publishes in your MQTT client.

1. Open your MQTT client and subscribe to the **Result Topic** specified in the **Formatter Settings**.
2. Trigger the reader and verify that you receive the results.



3. If you do not receive read results, refer to the section [Troubleshoot MQTT Connection on page 42](#).

Configure Device Attributes

You can assign custom attributes to the enterprise connection of your reader in the **Attributes** tab of the **MQTT** pane, allowing you to include information such as line association or station. You can reference these attributes in your output

format or connection properties. The  icon indicates fields that accept attributes.

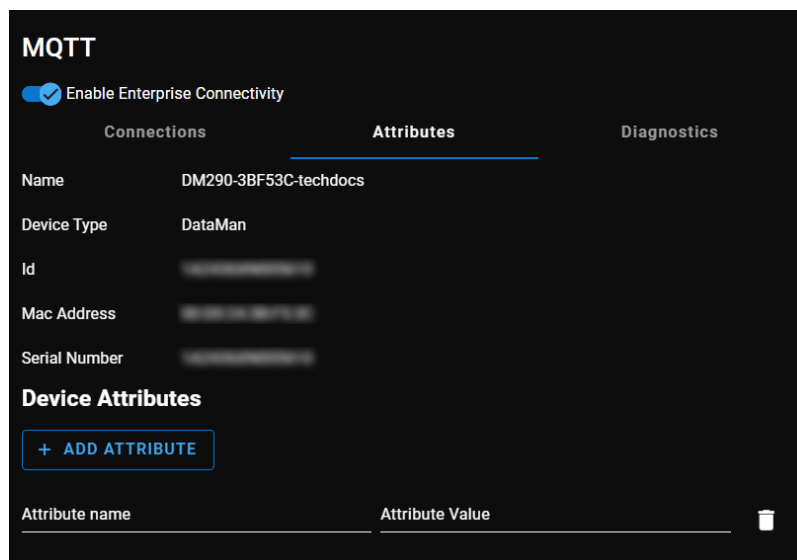
Your reader has the following attributes by default:

- Name
- Device Type
- ID
- MAC Address
- Serial Number

Note: You cannot delete the default attributes for your reader.

To add new attributes to your reader:

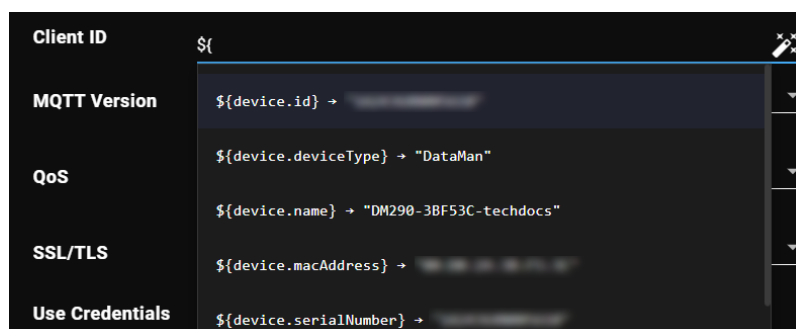
1. Open the **Attributes** tab of the **MQTT** pane.



The screenshot shows the MQTT configuration interface. At the top, there's a toggle for 'Enable Enterprise Connectivity' which is turned on. Below this are three tabs: 'Connections', 'Attributes' (which is selected), and 'Diagnostics'. Under the 'Attributes' tab, there's a form with fields for 'Name' (DM290-3BF53C-techdocs), 'Device Type' (DataMan), 'Id', 'Mac Address', and 'Serial Number'. Below these fields is a section titled 'Device Attributes' with a '+ ADD ATTRIBUTE' button. At the bottom, there's a table with two columns: 'Attribute name' and 'Attribute Value', and a trashcan icon on the right.

2. Click **Add Attribute** to create a new entry.
3. Enter the attribute name in the **Name** field of the new entry.
4. Enter the attribute value in the **Value** field of the new entry.

You can use attributes as variables in your connection settings:



The screenshot shows the MQTT configuration interface with a dropdown menu open for the 'Client ID' field. The dropdown menu displays several options, each starting with a variable reference: '\${device.id}', '\${device.deviceType}', '\${device.name}', '\${device.macAddress}', and '\${device.serialNumber}'. The 'Client ID' field is currently set to '\${'.

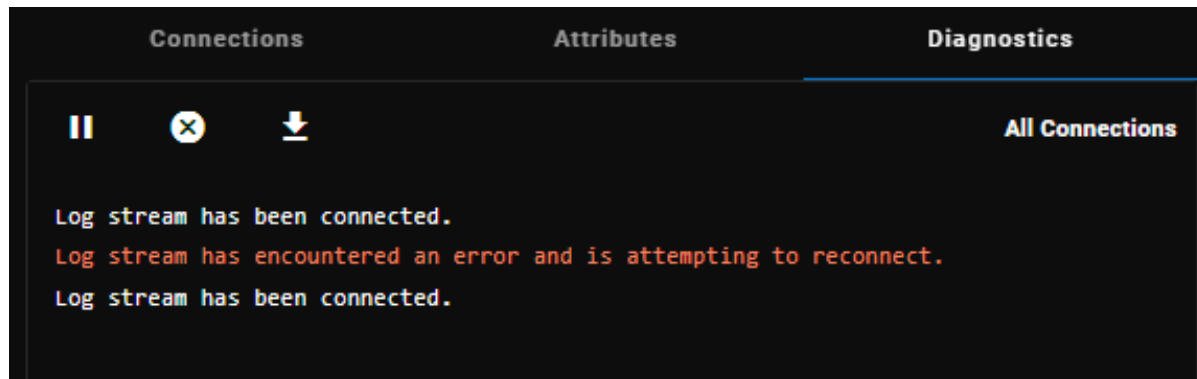
To delete a custom attribute, click on the trashcan icon next to the entry that you want to delete.

Troubleshoot MQTT Connection

This section provides troubleshooting instructions for the most common issues with establishing an MQTT connection.

Diagnostic Log

The **Diagnostics** tab allows you to troubleshoot and verify MQTT connections. When the **Enable Enterprise Connectivity** toggle is on and you have at least one connection set up, the **Diagnostics** tab displays diagnostic messages in real time.



Messages are displayed in the following colors to indicate the type of the message:

Type	Color	Description
INFO	White	Messages regarding connection settings changes and other non-critical messages.
DEBUG	Gray	Messages regarding debugging messages such as scripting-specific logic.
WARN	Orange	Messages regarding warnings such as missed image uploads and buffer overflows.
ERROR	Red	Messages regarding connection timeouts, unexpected disconnects, and other critical messages.

The buttons in the top left of the log screen allow you pause, clear, or export the messages, respectively.

No Results Published

If you are not seeing any published results from your reader, perform the following troubleshooting procedure:

1. Navigate to **Communication > Communication > MQTT**, and make sure that **Enterprise Connectivity** is enabled on your reader.
2. Trigger the reader and check the **Connection** tab of the **MQTT** pane to verify that Enterprise Connectivity is processing and publishing results.

The **Results Processed** and **Publish OK** counters increase with every successful result transfer.

3. Go to the **Diagnostics** tab of the **MQTT** pane to verify that the connection to the MQTT broker is active.

If your reader is disconnected, you can see an error message under the Disconnected state on the diagnostics page.

MQTT Broker Disconnected

If your MQTT broker is disconnected, perform the following troubleshooting procedure:

1. Set up a local MQTT broker to ensure that the reader is working correctly:
 - a. Download and install the latest version of [mosquitto for Windows](#) to a directory where you have full access.
 - b. Uncheck the service option, as you are going to run the broker from the command prompt in order to see the logs.
 - c. Allow the broker to receive connections from a different device by issuing the following command in PowerShell:


```
@("listener 1883 0.0.0.0`nallow_anonymous true") + (Get-Content "c:\mqtt-broker\mosquitto.conf") | Set-Content "c:\mqtt-broker\mosquitto.conf"
```
 - d. Run the broker by issuing the following command in PowerShell:


```
& 'C:\mqtt-broker\mosquitto.exe' -v -c .\mosquitto.conf
```
2. Use an MQTT client such as MQTTX to test that you can publish and subscribe to topics on your broker.
3. Configure the MQTT connection to the address of the PC that is connected to your reader.
4. Remove any credentials and certificates on your reader.
5. Investigate any mismatches between the disconnected MQTT broker and your local test broker.
6. If there are no mismatches, verify that the credentials and certificates on your reader are correct.

MQTT Disconnected Error Messages

The following table lists the possible Disconnected error messages and the potential causes:

Error Message	Potential Cause or Resolution
Connection refused (os error 111)	- Check that the port is correct. - Check that the firewall of the host allows external connections.
Connection refused, return code: `NotAuthorized`	Incorrect username or password.
Connection refused, return code: `BadUserNamePassword`	Incorrect username or password.
Connection closed by peer abruptly	- The broker stopped running. - The server may require TLS.
failed to lookup address information: Temporary failure in name resolution	Unable to resolve the hostname. Check that the reader can reach that target.
Mqtt serialization/deserialization error: Promised boundary crossed, contains 256 bytes	Non-TLS configuration used against an mTLS port.
Network timeout	The target machine is not responding. - Check the address again. - Check that the reader can reach that target.
TLS: I/O: invalid peer certificate: UnknownIssuer	Missing or invalid CA certificate. The certificate of the server cannot be verified.
TLS: I/O: received fatal alert: CertificateRequired	The server connection expects a client certificate but none was provided.
TLS: I/O: received fatal alert: UnknownCA	Bad client certificate. The certificate is signed by an unknown CA or there is a hostname mismatch.

Error Message	Potential Cause or Resolution
TLS: I/O: received fatal alert: BadCertificate	TLS configuration used against an mTLS port that requires client certificates.
TLS: I/O: received fatal alert: CertificateUnknown	The broker rejected the certificate. Check that the certificate is signed by a CA trusted by the broker.
TLS: Native TLS error Connection reset by peer (os error 104)	Requesting TLS, but the broker does not support it on that port.

Connection Fails with Valid Certificate and Configuration

If your connection is failing, and you have verified that your certificate is valid and your configuration is correct, perform the following troubleshooting procedure:

1. Verify that the time is correct on your reader.
2. Make sure that you are using timestamps in UTC.
3. Make sure that your reader is synchronized with a NTP server using the [Date & Time on page 85](#) settings option.

Create Custom Script for MQTT

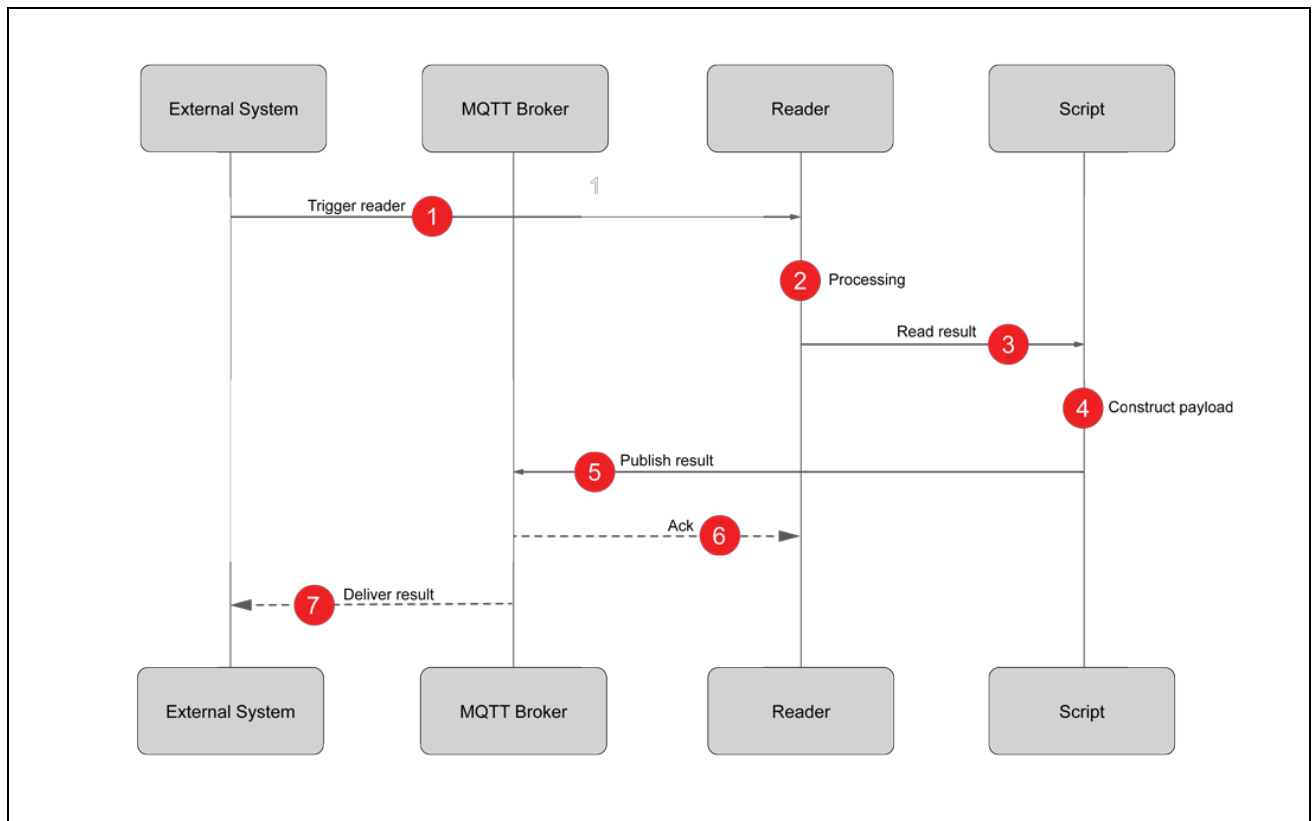
To create a custom script for MQTT communication, you can either use the default script as a template, or construct the necessary parts of the script using the provided code snippets.

When testing your custom script, use the **Diagnostics** tab of the **MQTT** pane for troubleshooting.

The following subsections of this document provide references and instructions for creating your own custom script.

MQTT Data Flow

The following image shows the data flow between external systems and your DataMan reader when communicating over an MQTT connection:



Step	Description
1	The external system triggers the reader.
2	The reader processes the acquired image.
3	The reader delivers the read result.
4	The script constructs the MQTT payload containing the read result.
5	The script publishes the read result to the result topic on the MQTT broker.
6	If your MQTT connection has QoS 1 or 2, the MQTT broker acknowledges the read result message.
7	The MQTT broker delivers the read result to the external system.

Default Script

The following code block contains the default script that your reader uses for MQTT formatting:

Note: When creating your own custom script for MQTT formatting, make sure that you include the default export class for the script engine to load. You can define any additional classes, functions, or variables.

```
import { device, connection } from "script-sdk";

// JsonFormatter provides a basic JSON protocol for
```

```

// communication between the device and the MQTT broker.
//
// It supports publishing results from the device to the connection.
export default class JsonFormatter {
  constructor() {
    // Setup topics for notifications and responses
    this.resultsTopic = `cognex/${connection.clientId}/results`;
    this.heartbeatTopic = `cognex/${connection.clientId}/heartbeat`;

    // Set up a Handler for the results from the device
    device.streams.results.setHandler(async (result) => {
      await this.onResult(result);
    });

    // Heartbeat every 60 seconds
    setInterval(async () => {
      await this.onHeartbeat();
    }, 60000);
  }

  async onResult(result) {
    // Publish the result to our results topic
    connection.publish(result, this.resultsTopic, getDefaultPublishOptions());
  }

  async onHeartbeat() {
    connection.publish({ status: "Healthy" }, this.heartbeatTopic,
    getDefaultPublishOptions());
  }
}

function getDefaultPublishOptions() {
  // Options can be set here as needed.
  // If not set, defaults will be used.
  const options = {
    // Mqtt V3 and V5
    //
    // qos: 1,
    //
    // retain: false,
    // Mqtt V5 only
    // contentType: "application/json",
    //
    // payloadFormatIndicator: 1,
    //
    // messageExpiryIntervalSecs: 60,
  }
}

```

```

        //
        // topicAlias: 23,
        //
        // correlationData: "correlation-data",
        //
        // userProperties: { "prop1": "value1" },
    };
    return options;
}

function isValidRequest(payload) {
    // Must be an object
    // Must have a "method" property of type string
    // "id" is optional, if present must be a string, number, or null
    // "params" is optional, if present must be an object or array
    if (typeof payload !== "object" || payload === null) {
        return false;
    }

    if (!("method" in payload) || typeof payload.method !== "string") {
        return false;
    }

    return true;
}

```

Script Structure

To understand how to structure your custom script, observe the structure of the default script. The default script consists of the following main parts:

- constructor
- onResult
- onHeartbeat

constructor

The `constructor` method in the script sets up the topic subscriptions for various events, such as receiving results from the reader, or initiating a heartbeat message loop.

- The following lines in the default script set up the result subscriptions:

```

device.streams.results.setHandler(async (result) => {
    await this.onResult(result);
});

```

These lines cause the `onResult` method to execute whenever the reader produces a new result.

- Start a heartbeat message loop using the `setTimeout` or `setInterval` JavaScript functions.

onResult(result)

The `onResult` method in the script constructs the result payload based on the read results from the reader.

To customize the result format, use the following `onResult` code snippet as a template, where `result` is the Cognex formatted result payload:

```

async onResult(result) {
  const metrics = result.metrics ?? {};

  // Build the new object, organized into an alternative preferred format
  let converted = {
    // Identity & state
    meta: {
      id: result.id,
      sequenceNumber: result.sequenceNumber,
      status: result.status,
      decoded: result.status === "Ok",
    },
    // Decoder related
    decoder: {
      codeContent: metrics["codeContent"]?.text || "",
      readString: metrics["readString"]?.text || "",
      symbology: metrics["symbology"]?.text || "",
      ppm: metrics["ppm"]?.measurement ?? 0,
    },
    // Time-based fields
    timing: {
      cycleStart: result.cycleStart,
      cycleDurationMs: result.cycleDurationMs ?? 0,
      processingDurationMs: result.processingDurationMs ?? 0,
      decodeTimeMs: metrics["decodeTime"]?.measurement ?? 0,
      triggerTimeMs: metrics["triggerTime"]?.measurement ?? 0,
    },
  };

  // Publish the converted result to our results topic
  connection.publish(converted, this.resultsTopic, {});
}

```

```

{
  "cycleDurationMs": 0.322,
  "cycleStart": "2026-07-07T04:46:27.534007+00:00",

```

```

    "id": "016807bb-fece-7fa3-aca1-d3c403d42a70",
    "metrics": {
      "codeContent": {
        "text": "2DMax with PowerGrid\r\n"
      },
      "decodeTime": {
        "measurement": 41.0
      },
      "ppm": {
        "measurement": 20.667573928833008
      },
      "readString": {
        "text": "2DMax with PowerGrid"
      },
      "symbology": {
        "text": "Data Matrix"
      },
      "triggerTime": {
        "measurement": 67.0
      }
    },
    "processingDurationMs": 41.0,
    "sequenceNumber": 9,
    "status": "Ok"
  }

```

onHeartbeat

The `scheduleHeartbeat` method sends periodic heartbeat messages from the reader to a designated MQTT topic, allowing external systems to monitor reader connectivity and status.

The method uses either the `setTimeout` JavaScript command to schedule the next heartbeat after sending each message, which allows for flexible or randomized intervals, or the `setInterval` command to schedule regular messages.

A heartbeat message usually includes a timestamp and message type. The script publishes the heartbeat message to a topic such as `cognex/${device.id}/heartbeat`.

To customize heartbeat messages, use the following code snippet as a template, and call the `scheduleHeartbeat` method from the constructor:

```

scheduleHeartbeat() {
  const period = 40000 + Math.random() * 10000; // 40 to 50 seconds
  setTimeout(() => {
    const heartbeatMessage = {
      Header: {
        MessageTimeStamp: new Date().toISOString(),

```

```

        MessageType: "Heartbeat",
    },
};
connection.publish(JSON.stringify(heartbeatMessage), this.heartbeatTopic, { qos: 1 });

// Once the message is sent, schedule the next heartbeat
this.scheduleHeartbeat();
}, period);
}

```

XML Processing

To enable XML processing and access XML functions, import the XML module with the following command:

```
import { XmlParser, XmlNode } from "xml";
```

The `XmlParser` constructor accepts an `options` object that allows you to customize how the script parses XML into a JavaScript object. The following options are available:

Option	Description	Example
<code>attributeNamePrefix</code>	The prefix added to attribute names in the output object. Default: <code>"@_"</code>	For example, <code><root attr="val"/></code> becomes <code>{ "root": { "@_attr": "val" } }</code> .
<code>ignoreAttributes</code>	If set to <code>true</code> , attributes are ignored and not included in the output. Default: <code>false</code>	
<code>textNodeName</code>	The property name used for text nodes in the output object. Default: <code>"#text"</code>	For example, <code><root>value</root></code> becomes <code>{ "root": { "#text": "value" } }</code> .
<code>tagValueProcessor</code>	Optional function to transform tag values. Signature: <code>(tagValue, jpath, hasAttributes) => newValue</code>	For example, you can convert all tag values to uppercase.
<code>attributeValueProcessor</code>	Optional function to process attribute values. Signature: <code>(attrName, attrValue, jpath) => newValue</code>	For example, you can add a suffix to all attribute values.

XmlParser Example

Creating XML string from JavaScript object:

```

/**
 * Recursively builds XML from a JSON object.
 * Arrays are represented as repeated 'item' tags.
 * @param {object} obj - The JSON object to convert.
 * @param {string} [nodeName='root'] - The root XML node name.

```

```

* @returns {string} XML string
*/
buildXmlFromObject(obj, nodeName = "root") {
  function buildNode(key, value) {
    if (Array.isArray(value)) {
      const arrNode = new XmlNode(key);
      value.forEach((item) => {
        arrNode.addChildNode(buildNode("item", item));
      });
      return arrNode;
    } else if (value !== null && typeof value === "object") {
      const objNode = new XmlNode(key);
      for (const [childKey, childValue] of Object.entries(value)) {
        objNode.addChildNode(buildNode(childKey, childValue));
      }
      return objNode;
    } else if (value === null) {
      return new XmlNode(key); // empty element for null
    } else {
      return XmlNode.of(key, String(value));
    }
  }

  const rootNode = buildNode(nodeName, obj);
  return rootNode.toString();
}

```

Full XML example:

```

import { device, connection } from "script-sdk";
import { XmlNode, XmlParser } from "xml";

export default class XmlFormatter {
  constructor() {
    // Setup topics for notifications and responses
    this.resultsTopic = `cognex/${connection.clientId}/results`;

    // Set up xml parser
    this.parser = new XmlParser({
      attributeNamePrefix: "",
    });

    // Set up result handling
    device.streams.results.setHandler((result) => {
      this.onResult(result);
    });
  }
}

```

```

}

async onResult(result) {
    // Determine send properties
    const payload = this.buildXmlFromObject(result, "result");

    // Publish the message
    console.log(`Publishing Result to topic ${this.resultsTopic}: ${payload}`);
    connection.publish(payload, this.resultsTopic);
}

/**
 * Recursively builds XML from a JSON object.
 * Arrays are represented as repeated 'item' tags.
 * @param {object} obj - The JSON object to convert.
 * @param {string} [nodeName='root'] - The root XML node name.
 * @returns {string} XML string
 */
buildXmlFromObject(obj, nodeName = "root") {
    function buildNode(key, value) {
        if (Array.isArray(value)) {
            const arrNode = new XmlNode(key);
            value.forEach((item) => {
                arrNode.addChildNode(buildNode("item", item));
            });
            return arrNode;
        } else if (value !== null && typeof value === "object") {
            const objNode = new XmlNode(key);
            for (const [childKey, childValue] of Object.entries(value)) {
                objNode.addChildNode(buildNode(childKey, childValue));
            }
            return objNode;
        } else if (value === null) {
            return new XmlNode(key); // empty element for null
        } else {
            return XmlNode.of(key, String(value));
        }
    }

    const rootNode = buildNode(nodeName, obj);
    return rootNode.toString();
}
}

```

Troubleshooting Your Script

Open the browser and go to the **Diagnostics** tab of the **MQTT** pane to look for errors.

Concurrency limit exceeded

Possible log error message:

- `Skipped incoming message callback due to concurrency limit`
- `Skipped device result callback due to concurrency limit`

By default, the maximum number of inflight script promises is limited. If you exceed this limit, the script can still create promises, but the following limitations apply:

- script callbacks are skipped (`onResult`)
- concurrency limit exceeded errors are logged
- the script error count statistic is incremented

If you encounter a concurrency limit exceeded error, you can adjust the concurrency limit in the **Advanced Settings** dropdown, or you can set the limit to 0 for no limit.

Note: Setting the concurrency limit to 0 can cause the reader to be unstable.

Out of Memory

Possible log error message:

- `out of memory`

By default, the maximum JavaScript heap size is limited. When the limit is exceeded, the following limitations apply:

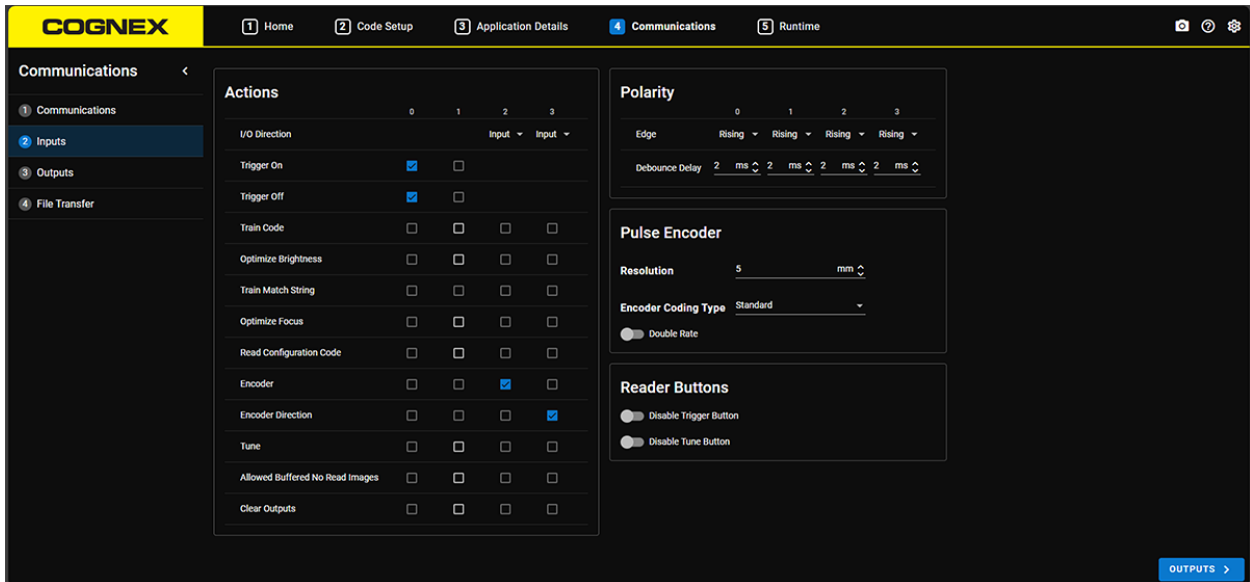
- an out of memory error is logged
- the script error count statistic is incremented

If you encounter an out of memory error, you can adjust the maximum heap size in the **Advanced Settings** dropdown, or you can set the maximum heap size to 0 for no limit, in which case the reader is only limited by available RAM.

Note: Setting the maximum heap size to 0 can cause the reader to be unstable.

Inputs

The **Inputs** substep allows you to configure what action the reader takes when it receives an input signal.



By default, the reader responds to a signal over input line 0. The reader only supports trigger signals on input lines 0 and 1.

You can configure the reader to take any of the following actions:

- **Trigger On** and **Trigger Off** are enabled by default on the same input line 0. **Trigger Off** is automatically the inverse of **Trigger On**. If you put the two signals on separate lines, you can set the polarity independently according to your needs.
- **Train Code**: the reader is trained with the expected symbology, which can decrease the time required to decode subsequent symbols.
- **Optimize Brightness**: the reader finds the best possible light and imager settings to read a code.
- **Train Match String**: the reader uses the decoded code to match and validate subsequent codes against.
- **Optimize Focus**: the reader adjusts to the best possible focus settings to read a code. This option is only available for readers with liquid lens.
- **Read Configuration Code**: the reader reads the code as a configuration command instead of a result that needs to be reported. This read does not increase trigger count and does not produce a NoRead result.
- **Encoder**: connect your encoder to this input line.
- **Encoder Direction**
- **Tune**: the reader finds the best possible settings to read a code. This is the equivalent of rerunning the Tuning substep.
- **Allow Buffered No Read Images**: when this input is active and connected to a sensor that detects a part or package, the reader buffers No Read images if it is configured for no-read image saving. This is useful, for example, when you only want to store No Read images if an actual part or package is in the field of view of the reader.
- **Clear Outputs**: all outputs are reset immediately if the input gets triggered. This option does not reset any output delay settings.

 **Note:** The input line actions Train Code, Optimize Brightness, Train Match String, Read Configuration Code, and Tune are carried out only with the next trigger.

You can also configure the following signal detection options for each input line:

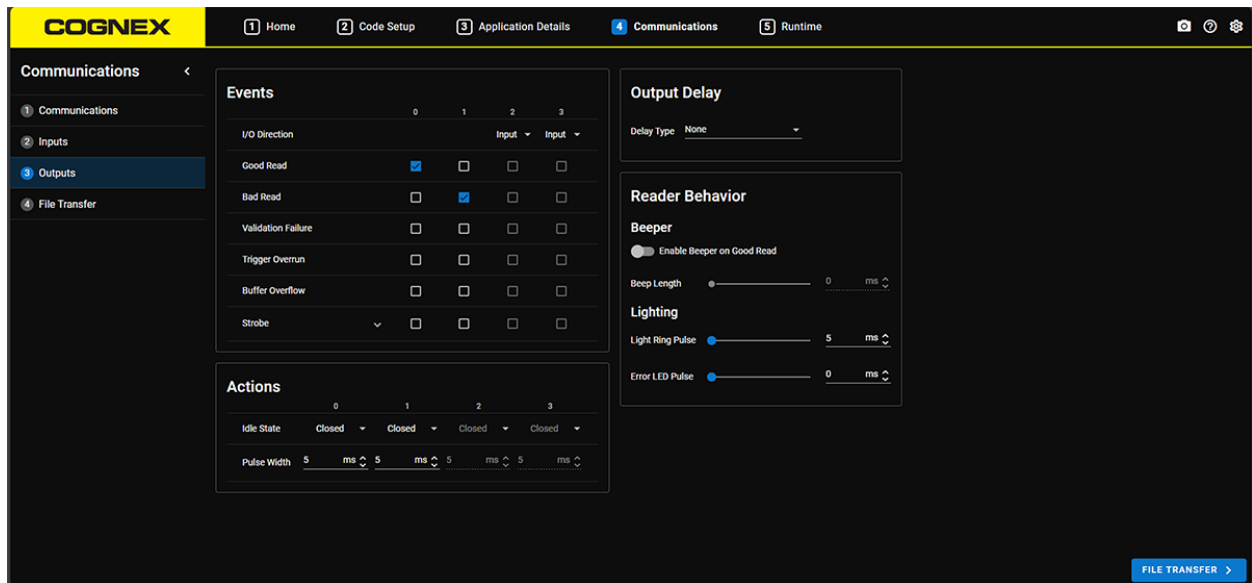
- The **Polarity** option allows you to set the change in the input signal that triggers an image acquisition.
- The **Debounce Delay** text box allows you to define how long the trigger signal must be detected in order to be recognized as valid. Use a shorter value to compensate for ESD line noise, and higher values to compensate for noise in electromechanical relays. Experiment with different **Debounce Delay** values to determine the most appropriate value for your production environment.

You can also configure whether the buttons on the reader are active. Turn on the **Disable Trigger Button** or **Disable Tune Button** toggles to prevent the reader from initiating a read, train, or tune attempt when you press the physical button on the reader.

Outputs

The **Outputs** substep allows you to configure what output signals the reader produces for different events.

You can set up an output delay of either time or distance, and update reader beeper and lighting behavior for different events such as Good Reads or Bad Reads.



I/O Direction

Configure whether I/O lines 2 and 3 behave as an Input or Output. Lines 0 and 1 are fixed output lines.

Good Read and Bad Read

If data validation is activated, the reader only signals a Read event if a code was read and passed validation. With Good Read and No Read events assigned to two outputs, the following logic applies:

- Good read, validation pass: **Good Read** output
- Good read, validation failure: **no** output, or Validation Failure if enabled
- Bad read: **Bad Read** output

Validation Failure

The reader signals the **Validation Failure** event when a code fails data validation, and outputs the error string configured in the **Data Validation** substep.

Trigger Overrun

The reader signals the **Trigger Overrun** event when an image could not be acquired because the reader was busy. Possible reasons include an ongoing burst or a long exposure for a previous trigger.

Buffer Overflow

The reader signals the **Buffer Overflow** event when the reader can acquire an image, but does not have any available internal image memory to save it. This happens when images are acquired faster than they can be processed and cleared from the buffer. A possible reason is that the decoding time is longer than the trigger interval.

Strobe

You can strobe an external illumination by checking **Strobe** on one of the output lines.

File Transfer

You can configure the reader to automatically transfer images, result data, and code quality verification reports from its memory to a destination on your network. The settings in the **File Transfer** substep allow you to configure file transfer using the following methods:

- **FTP:** The reader transfers files to an FTP server you set up on your network, for example, using FileZilla. This option uses insecure HTTP, which means that the data is not encrypted.
- **FTP Mitsubishi GOT:** The reader transfers files through the FTP Server/Client function built into Mitsubishi hardware. This option enhances FTP file transfer on Mitsubishi GOT and other FTP servers by allowing:
 - Idle timeout through periodic NOOPs
 - Optional timeout to close an idle control connection
 - Mitsubishi GOT write mode support
- **SFTP:** Same as the FTP option but with secure file encryption. Use this option when transferring files to the cloud or over untrusted networks.

Cognex recommends using SHA-256 fingerprints for SFTP communication. SHA-1 and MD5 are also available for legacy applications, but are cryptographically weak.

- **TCP:** For image transfer only. The reader transfers image data to a TCP server. The reader opens the connection and expects a listening socket on the receiving end. This is similar to FTP.
- **Local TCP Server:** For image transfer only. The reader acts as the TCP server. You configure a listening socket on the reader that waits for an incoming connection from your network.

When using the **Local TCP Server** option, keep the following in mind:

- You can have maximum one open connection.
- No buffering takes place. You cannot retrieve images captured before the connection was established. You might lose images if the connection drops and is re-established.

The screenshot displays the COGNEX software interface with the 'File Transfer' substep selected. The interface is organized into three main configuration panels:

- Image Transfer:** Configure the transfer of images to a server.
 - Storage: FTP
 - Address / Port: 21
 - File Path: (empty)
 - Username: (empty)
 - Password: (empty, with a note: Password will be sent via insecure HTTP)
 - Idle Timeout: 0 SEC
 - Images to Transfer: No images
 - File Name: image.000.bmp
 - File Type: bmp
 - Image Size: Full
 - Overlay Graphics: (toggle switch)
- Result Transfer:** Configure the transfer of results to a server.
 - Storage: FTP
 - Address / Port: 21
 - Username: (empty)
 - Password: (empty, with a note: Password will be sent via insecure HTTP)
 - Idle Timeout: 0 SEC
 - Transfer Results: (checkbox)
 - File Name: result.txt
 - Append Results: (checkbox, checked)
- Verification Report Transfer:** Configure the transfer of verification reports to a server.
 - Storage: FTP
 - Address / Port: 21
 - Username: (empty)
 - Password: (empty, with a note: Password will be sent via insecure HTTP)
 - Idle Timeout: 0 SEC
 - Transfer Reports: (checkbox)
 - File Name: pcm_report.html
 - Units: mil

The interface includes a top navigation bar with tabs for Home, Code Setup, Application Details, Communications (selected), and Runtime. A left sidebar shows the Communications section with sub-items: Communications, Inputs, Outputs, and File Transfer (selected). A 'RUNTIME' button is located at the bottom right.

Image Transfer

The **Image Transfer** option allows you to transfer image files, with optional .svg overlay graphics, to a destination on your network.

The reader cannot guarantee that all images are sent. This can happen when the reader is operating at a higher frequency and cannot schedule time for transferring images.

The following options are available:

- **Storage:** the transfer method you want to configure for image transfer, such as FTP or TCP. Select the option that best fits the requirements of your network environment.
- **Server Address:** the address and port number of the file server on your network.
 - For **FTP** connections, normally you do not need to change the default port.
 - For **Local TCP Server** connections, the address field is not available because the reader acts as the TCP server. You only need to specify the port the reader leaves open to listen for client requests.
- **SFTP Fingerprint** (only available with SFTP option): the unique hash of the public key for the SFTP server.
- **File Path:** the file path in relation to the root directory of the FTP server, for example, */failed_images/image*. The directory must be available and accessible. The reader cannot create directories on the file server.
- **Username:** the name of the user that the reader can use to access the FTP server.
- **Password:** the password for the user that the reader can use to access the FTP server.
- **Idle Timeout:** enable or disable an idle timeout, defined in seconds.
- **Images to Transfer:** select the category of images to transfer based on the inspection results.
- **File Name:** define the name to use for transferred images, appended with a three-digit index.
- **File Type:** select the image type (*.bmp*, *.jpg*, or *.png*) for the transferred images.
- **Image Size:** select the degree of downscaling, if any, for the transferred images.
- **Overlay Graphics:** enable or disable the SVG overlay of result graphics on the transferred images.

Result Transfer

The **Result Transfer** option allows you to append result data to a text file stored on an FTP server.

The following options are available:

- **Storage:** the transfer method you want to configure for result data transfer. TCP is not supported.
- **Server Address:** the address of the FTP server.
- **SFTP Fingerprint** (only available with SFTP option): the unique hash of the public key for the SFTP server.
- **Port Number:** the port number of the FTP server.
- **Username:** the name of the user that the reader can use to access the FTP server.
- **Password:** the password for the user that the reader can use to access the FTP server.
- **Idle Timeout:** enable or disable an idle timeout, defined in seconds.
- **Transfer Results:** enable or disable transferring results.
- **File Name:** allows you to specify the name of the file that collects the results. If you enable **Append Results**, make sure that the FTP user has Append permissions, which is different from Write permissions.

Report Transfer

The **Verification Report Transfer** option allows you to send a report with code quality or TruCheck™ details to an FTP server. Every trigger generates a new report file. These settings are only available if you have enabled any of the verification settings under **Code Quality** in the **Application Details** step.

The following options are available:

- **Storage:** the transfer method you want to configure for result data transfer. TCP is not supported.
- **Server Address:** the address and port number of the FTP server.
- **SFTP Fingerprint** (only available with SFTP option): the unique hash of the public key for the SFTP server.
- **Username:** the name of the user that the reader can use to access the FTP server.
- **Password:** the password for the user that the reader can use to access the FTP server.
- **Idle Timeout:** enable or disable an idle timeout, defined in seconds.
- **Transfer Reports:** enable or disable transferring reports.
- **File Name:** allows you to specify the name of the file that collects the results.
- **Units:** specifies the unit of measurement used for dimensional values in the transferred data. Select **mm** (millimeters), **mils** (thousandths of an inch), or **µm** (micrometers). Available if **TruCheck™ Grading** is enabled.

Transferring Images over TCP

TCP provides an alternative to FTP for transferring images from the reader. When you use TCP, you send image data directly to a remote destination on your network. This approach avoids saving images to a file server and then copying them to a database. Using TCP simplifies integration and reduces latency.

In contrast, FTP requires you to set up a file server, monitor the file share for new files, and copy each file to your database. This process is more complex and can slow down your workflow.

When you enable TCP image transfer, the reader sends image files over a TCP connection to a destination on your network. Your application on the remote destination receives the image data and processes each file as needed.

You can configure TCP on the reader as an alternative to FTP or SFTP. At the same time, you must implement an application on your network to receive and process images from the reader.

TCP Message Format

When you use TCP for transferring images, the reader sends each image using a simple binary message format. The TCP message consists of a header followed by the body, which contains the image data described in the header.

The header size is fixed. It contains information about the image size, image type, and the image file name. The body size is variable and depends on the image.

The following table describes the TCP message format and the size of each field:

Section	Contents	Size	
Header	Image Size	4 bytes	
	Image Type	4 bytes Valid values:	
		0	BMP
		1	PNG
		2	JPG
	9	SVG	
	Image File Name	128 bytes	
Body	Image Data	Variable	

Sample TCP Server for TCP Option

To help you implement a TCP server that receives images from the reader, the following Python sample script demonstrates how to read and process the TCP message format described above.

The script performs the following steps:

1. Accept a TCP connection from the reader.
2. Read the fields of the fixed-size header.
3. Read the image data using the image size to determine how many bytes to receive.
4. Process or store the image.

By following the structure of the TCP message format, you can adapt the sample script to the requirements of your application.

```
"""
This code is provided as an example. It is not intended for use in a production environment
without further testing and validation. The author and the organization they represent are not
responsible for any issues
that may arise from using this code.
"""

import socket
import struct

# host machine IP address and port
server_ip = "<insert host ip>"
server_port = 4444

# Create a socket object
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Bind the socket to a specific address and port
s.bind((server_ip, server_port))

# Start listening for incoming connections
s.listen(1)

print("Server is listening on port", server_port)

filesReceived = 0

try:
    # Receive data from the client
    while True:
        # Accept a connection
        print("Waiting on connection...")
        c, addr = s.accept()
        print("Got connection from", addr)

        data = ""

        while True:
```

```

# Read header
data = c.recv(8 + 128)
print("Received header length", len(data))
if not data:
    print("Header not received, aborting read")
    break

fileSize = 0
fileSizeBuffer = data[:4]
fileSize = struct.unpack("<I", fileSizeBuffer)[0]
print("FileSize", fileSize)
fileTypeBuffer = data[4:8]
fileType = struct.unpack("<I", fileTypeBuffer)[0]
print("FileType", fileType)
fileNameBuffer = data[8:]
fileName = struct.unpack("128s", fileNameBuffer)[0]
print("FileName", fileName.decode())

# Read data in 1024 packets

received = 0
while received < fileSize:
    readPacketSize = 1024
    if fileSize - received < 1024:
        readPacketSize = fileSize - received
    data = c.recv(readPacketSize)
    received += len(data)

    if not data:
        print("Got bad data packet aborting file read")
        break

# Save the file
filesReceived += 1
print("Files Received", filesReceived, "Last file size:", fileSize)
print("")

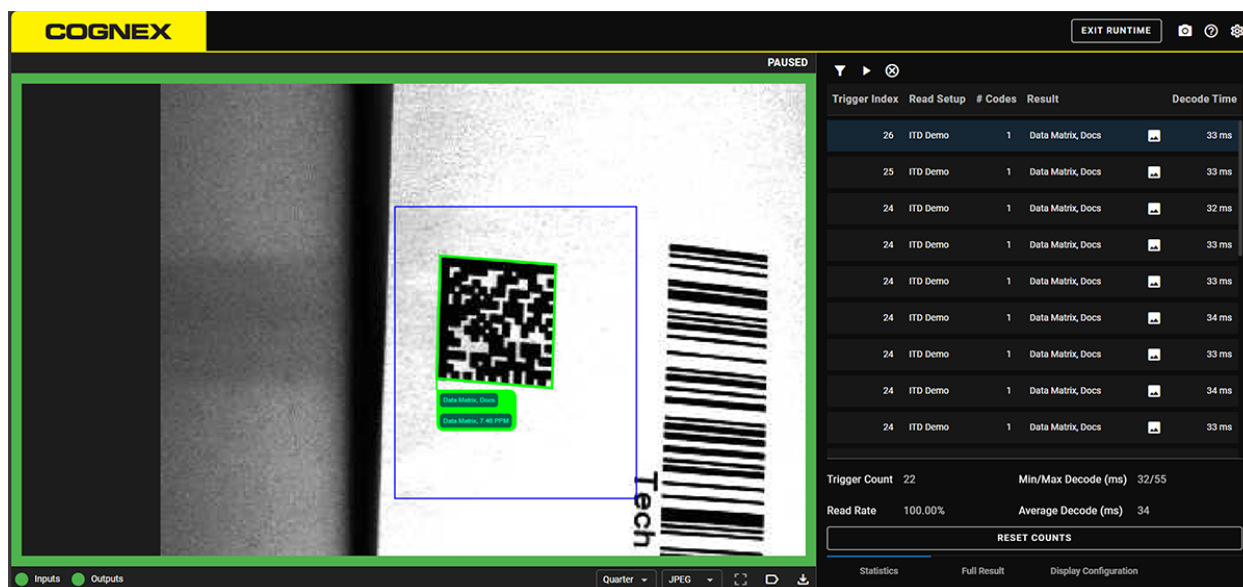
except KeyboardInterrupt:
    print("\nExiting")

```

Runtime

The **Runtime** step allows you to monitor the read results and performance of the reader in production.

The **Runtime** step provides a summary table showing trigger counts, current read rates, and decode time statistics.



The result table lists the read codes. The results table can display the **Full Result** string, provide **Statistics** about reader performance. Click **Display Configuration** to customize the data shown for individual reads and overall statistics.

You can filter, pause, or clear the incoming result data.

Click on a result with the image icon to display the image on the screen.

The screen shows the last image of the reader, or the image acquired for the selected read. You can move the image within the screen, and zoom with the mouse wheel.

Enable the **Result Status** option on the **Display Configuration** tab for more detailed status information. The **Status** column appears with the following icons to indicate different read results:

Icon	Meaning
	Good read
	Used AI enhancer (feature key required)
	Validation failure
	Code found but could not be read
	No read

Click on the **Launch Diagnostics** button to open the Diagnostics page in a separate window. The Diagnostics page is a software-based diagnostic tool that allows you to view the input and output signals your DataMan reader sends and receives. For more information, see [Diagnostics on page 64](#).

Configure the displayed image with the buttons on the bottom of the screen:

- Choose the image resolution and format.
- Fit the image to the screen.
- Turn labels on or off.
- Download the latest image in full resolution.

The bottom toolbar also indicates if **Inputs** or **Outputs** are enabled.

 **Note:** You can password protect the UI through the **Settings** menu to make only the **Home** and **Runtime** steps accessible. For more information, see [Password Protection on page 86](#).

Diagnostics

The Diagnostics page is a software-based diagnostic tool that allows you to view the input and output signals of your DataMan reader. By using Diagnostics, you can understand system loads to optimize performance, troubleshoot application issues, and identify corrective measures.

Diagnostics visualizes the high-low transitions of the I/O signals based on software events, not the actual electrical signals that the reader transmits. As a result, timestamps might differ from real electrical measurements. Diagnostics is suitable for logical sequence debugging, but not for accurate latency troubleshooting. Make sure that the browser window is always active and visible to avoid missing events.

Diagnostics records several minutes of activity for the live view. The current view of the display is available as a PNG format image. You can also download the last 65,536 recorded events in an IEEE-1364-2005 compliant VCD file for analysis in a third-party application, such as GTKWave.

Note: You cannot upload VCD files to the WebUI.

User Interface



Number	Element
1	Play/pause button
2	Display mode menu
3	Reader signals
4	Live I/O status
5	Automatic measurements
6	Manual measurement
7	Reset display
8	Download screenshot
9	Download VCD file
10	Window duration

Signals

The Diagnostics page displays input and output signals in an oscilloscope-style view. The following signals are available for inspection:

- Input 0, 1, 2, 3: Indicate activity on the given input line.

Note: Input lines do not show encoder signals. You can only view encoder activity in the live I/O status display.

- Trigger Overrun: The reader cannot acquire a new image, because it is currently executing the previous trigger sequence. The following events can cause a trigger overrun:
 - The reader receives a trigger while the current trigger is still active.
 - The reader triggers faster than it can acquire images. This can be indicated by a high Camera Busy signal.
- Trigger Event: The reader accepts and executes a trigger.
- Camera Busy: The reader is reserved for capturing images. Since trigger delay impacts the Camera Busy signal, you can use it to visualize the complete image acquisition time for each trigger.
- Acquisition: Image acquisition is in process.

Note: The Acquisition signal only shows the time when the reader scheduled the acquisition. It does not account for timing differences caused by PTP delays or other sources.

- Buffer Overflow: The image buffer memory of the reader is full, and it cannot store incoming images.
- Process Result: The reader is compiling a decode result with data formatting. If you use script-based data formatting, this signal shows the execution time of the `onResult()` function.
- Output 0, 1, 2, 3: Indicate activity on the given output line. By default, line 0 signals a Good-read and line 1 is a No-read.

Note: On a reader with bidirectional lines, all the input and output signals are present, but directionality is indicated by graying out the given line. For example, if you set Input 2 to Output, the Input 2 line is gray.

Display Modes

For each signal, the following display modes are available:

- **AUTO**: A free-running mode that continuously displays the signals. Diagnostics runs in **AUTO** mode by default.
- **HOLD**: One-shot mode that pauses the live view when the given signal reaches the selected state. When the display pauses, the signal is highlighted in orange.
- **SYNC**: A free running mode that centers the view to the selected state of the given signal. You can only configure one line at a time to be in **SYNC** mode. When synced events occur, the signal is highlighted in blue.

To configure display modes:

1. Click the chevron button on the signal you want to monitor.
2. Select a rising or falling edge to enter **HOLD** mode.
3. Click the circular arrow button to switch to **SYNC** mode.

You can pause event playback at any time with the **Pause** button or by interacting with the display. When the display is on pause, hover over a signal to display the following information:

- Time between the closest rising edges
- Time between the closest falling edges
- Time between the closest rising and falling edge

User Actions

The following actions are available to control the display:

- Play/Pause: click the play button or press the **E** key to pause or resume event playback.
- Measure: left-click and drag to display a time measurement between the selected points on the top of the screen.
- Zoom: magnify an area of the display:
 - Right-click and drag to zoom in on the selected area.
 - Hold down **CTRL** and scroll with the mouse wheel.
- Pan: use the mouse wheel to pan around when the view is paused.
- Window duration: select a time scale preset to provide more context for your signals or focus in on fast events.
- Reset display: click the button or the middle mouse button to reset the display to the state before it was paused. The display also resets to the configured window duration when you resume the live view.

Examples

The following table lists possible uses of the Diagnostics page.

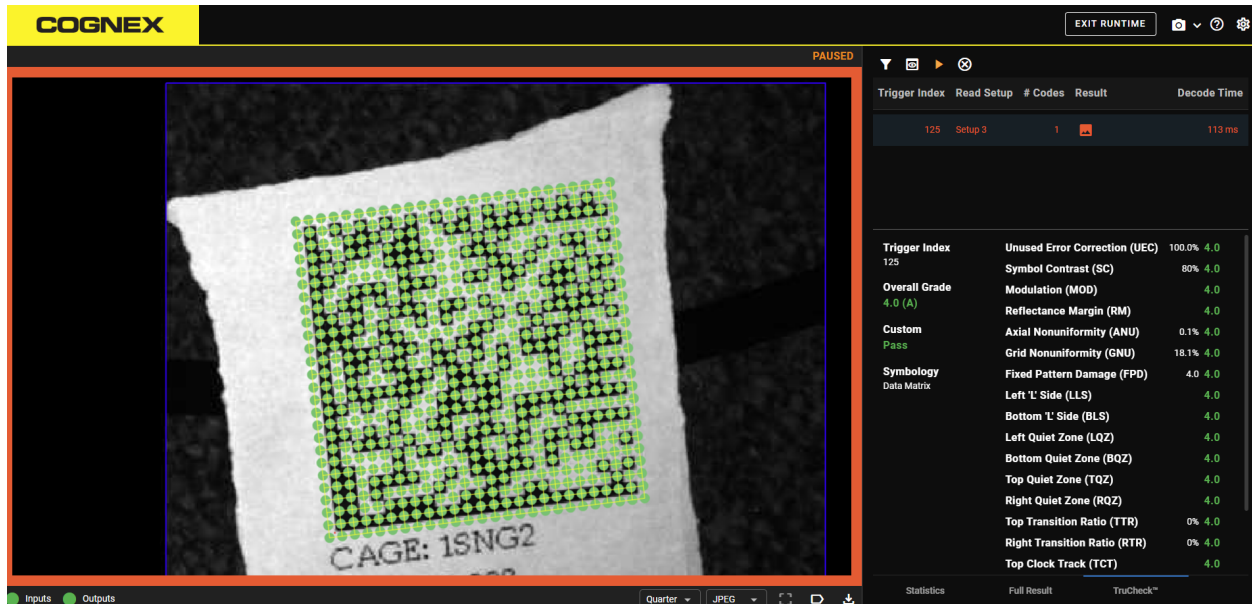
Use Case	Description
Initial application setup	To validate that the reader functions according to the intended application, verify input and output signaling, overall trigger duration, and gaps between triggers.
Process time variation	To monitor decoding jitter and outliers, use SYNC mode on the rising edge of the trigger line, and observe relative output signaling.
Programmatic output signals	To audit script-controlled signaling, create a script that ties a user event to one of the output lines, and set HOLD mode on the given output signal. For example, output an user event to hold the given line when one of the following occurs: • 2 out of the last 5 decodes are No-reads. • 3 out of the last 10 decodes had a Decode Time larger than 50 ms. • 3 out of the latest 5 decodes had a quality grade lower than B. For more details on scripting, refer to the <i>DataMan Communications and Programming Guide</i>. Note that scripting is not available on the WebUI.

Failure event analysis	To watch for errors, set HOLD mode on the Trigger Overrun or Buffer Overflow signals.
	A typical source of trigger overruns are false triggers. False triggering can be caused by, for example: • Noisy or poor quality discrete signals • Incorrect photo eye sensor setup at a conveyor belt • Reflections Try fixing the signal quality by reducing grounding issues, the photo eye signal, or the application. If not possible, use a debounce delay.
	Typical sources of Buffer Overflow: • A combination of fast trigger times and long output delay • High-volume image transfer Try configuring a larger image buffer or reduce the number of images to be transferred.

TruCheck™ Grading

TruCheck™ is a feature that implements ISO/IEC 15415, ISO/IEC 15416, and ISO/IEC 29158 barcode quality grading. TruCheck™ is available on verifiers and select readers. When you use TruCheck™ on a reader that does not use strictly compliant lighting, the results are called Standards-Based Grading (SBG). SBG provides grades based on an image captured by a reader that might not be ISO-compliant, but calculates the grades based on ISO standards. Therefore, the results of SBG can correlate to ISO-compliant verification results depending on the reader setup.

SBG is useful in applications where ISO compliance is not possible, for example, when there is not enough space to mount a reader and lighting at required distances and angles. SBG setups might also provide a Field of View or resolution that is different from that of verifiers.

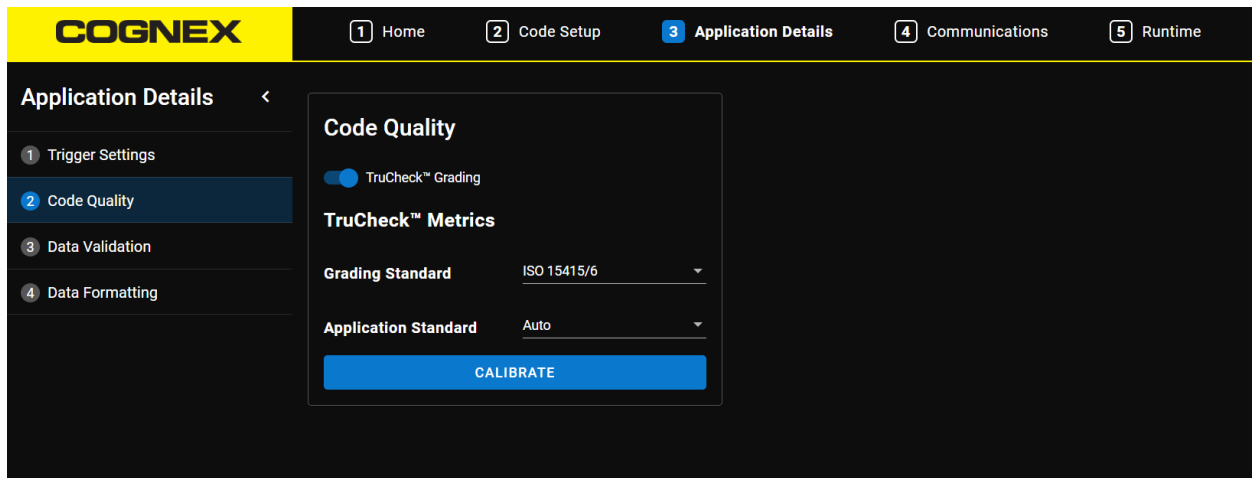


To use TruCheck™ Grading, set up your reader using the regular application steps and perform the TruCheck™-specific actions in the following substeps:

- Code Quality: Enable and configure TruCheck™ Grading. For more information, see [Configuring Grading on page 68](#).
- Data Formatting: Add data results from the TruCheck™ category to the string. For more information, see [Data Formatting on page 24](#).
- File Transfer: Set up the **Verification Report Transfer** to send SBG results to an FTP server. For more information, see [Transferring Verification Reports on page 71](#).
- Runtime: Check the SBG quality parameters under the TruCheck™ tab. For more information, see [Examining TruCheck™ Results on page 72](#).

Configuring Grading

The **Code Quality** substep includes the settings for enabling and configuring TruCheck™.



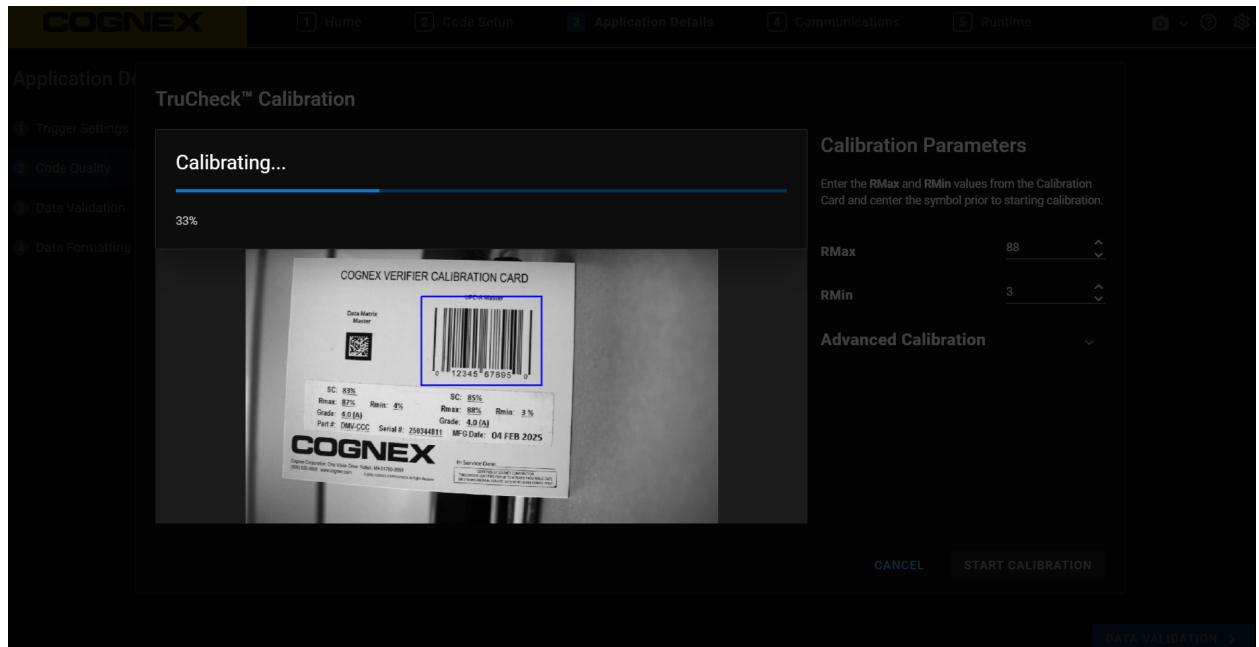
Setting up TruCheck™

To set up TruCheck™, do the following:

1. Enable **TruCheck™ Grading**.
As a result, the TruCheck™-specific options appear and the process control metrics are disabled.
2. Select the **Grading Standard** and **Application Standard**.
The **Dot Peen** option is available if you select **ISO 29158 (AIM-DPM)** for the **Grading Standard**.
3. Click **Calibrate**.
4. Set the **RMax** and **RMin** values to the values that your calibration card displays.
5. Place the code in the center of the Field of View and edit the ROI.



Tip: If you have the DMV-CCC Cognex Calibration Card, you can use either the UPC or the Data Matrix code for calibration, regardless of your application. If the Field of View is not large enough for the UPC code, you can use the Data Matrix code.

6. Click **Start Calibration**.7. When the calibration is complete, click **Save Calibration**.

Application Standards

The following table details the application standards you can select when setting up TruCheck™ Grading:

Application Standard	Description
GS1	The GS1 application standard follows GS1 General Specification guidelines in code verification.
HIBCC	The HIBCC application standard follows the HIBCC standard in code verification.
UDI (GS1 or HIBCC)	The UDI (GS1 or HIBCC) application standard checks symbols that meet UDI requirements for medical devices using GS1 or HIBCC guidelines.
UID (MIL-STD-130)	The UID (MIL-STD-130) application standard reports the quality standard according to the MIL-STD-130, specifying UID marks Construct 1 and Construct 2 that use data structure and code grade for verification. The MIL-STD-130 defines acceptable grades and requirements for data format.
Custom	The Custom application standard option allows you to customize the settings for grading. This option is useful when the code grading is not expected to adhere to any predefined industry conformance standards. The option has the following settings: • Minimum and Maximum X-dimension • Overall Pass Grade • Data Format Check • Aperture Setting Note: The Aperture Setting option is automatically set to Auto Aperture when the Grading Standard is ISO 29158 (AIM-DPM).

Application Standard	Description
Auto	The Auto application standard option allows the reader to select the correct application standard based on the format of the data encoded in the symbology. If the reader detects GS1, HIBCC, or MIL-STD-130 standards within a symbology during verification, the reader uses the correct application standard.
Cryptocode	The Cryptocode application standard checks symbols that meet Russian Cryptocode standard or other Cryptocode applications.

Calibration Process

Calibration sets reflectance and spatial references by checking grey scale levels to measure contrast and pixel size to measure code dimensions.

Note: You must use a code for calibration that has known RMin and RMax values, for example, a Cognex Calibration Card, such as the DMV-CCC or DMV-DMCC calibration cards.

Transferring Verification Reports

The **File Transfer** substep includes an option to send verification reports with a selected protocol with TruCheck™ Grading. The settings to configure sending the TruCheck™ results appear here if you enable **TruCheck™ Grading** in the **Code Quality** substep.

The screenshot displays the COGNEX software interface with the 'Communications' tab selected. The interface is divided into three main sections for configuring data transfer:

- Image Transfer:** Configure the transfer of images to a server. Fields include Storage (SFTP), Server Address, File Path, Username, Password, Idle Timeout, Images to Transfer, File Name, File Type, Image Size, and an Overlay Graphics toggle.
- Result Transfer:** Configure the transfer of results to a server. Fields include Storage (SFTP), Server Address, SFTP Fingerprint, Username, Password, Idle Timeout, Transfer Results, File Name, and an Append Results checkbox.
- Verification Report Transfer:** Configure the transfer of verification reports to a server. Fields include Storage (FTP), Server Address, Username, Password, Idle Timeout, Transfer Reports, File Name, and Units.

A 'RUNTIME' button is visible at the bottom right of the interface.

The **Verification Report Transfer** option allows you to send a report with code quality or TruCheck™ details to an FTP server. Every trigger generates a new report file. These settings are only available if you have enabled any of the verification settings under **Code Quality** in the **Application Details** step.

The following options are available:

- **Storage:** the transfer method you want to configure for result data transfer. TCP is not supported.
- **Server Address:** the address and port number of the FTP server.
- **SFTP Fingerprint** (only available with SFTP option): the unique hash of the public key for the SFTP server.
- **Username:** the name of the user that the reader can use to access the FTP server.

- **Password:** the password for the user that the reader can use to access the FTP server.
- **Idle Timeout:** enable or disable an idle timeout, defined in seconds.
- **Transfer Reports:** enable or disable transferring reports.
- **File Name:** allows you to specify the name of the file that collects the results.
- **Units:** specifies the unit of measurement used for dimensional values in the transferred data. Select **mm** (millimeters), **mils** (thousandths of an inch), or **µm** (micrometers). Available if **TruCheck™ Grading** is enabled.

Examining TruCheck™ Results

The **Runtime** step displays TruCheck™ results for the selected trigger. This includes, for example, **Trigger Index**, **Symbology**, and SBG quality parameters. The grading results appear below the result table, the **TruCheck™** tab is available when you enable **TruCheck™ Grading** in the **Code Quality** substep.

The screenshot shows the COGNEX TruCheck™ Runtime interface. The main window displays a paused image of a barcode with a green grid overlay. Below the image, the text "CAGE: 1SNG2" is visible. The right sidebar shows the results table with the following data:

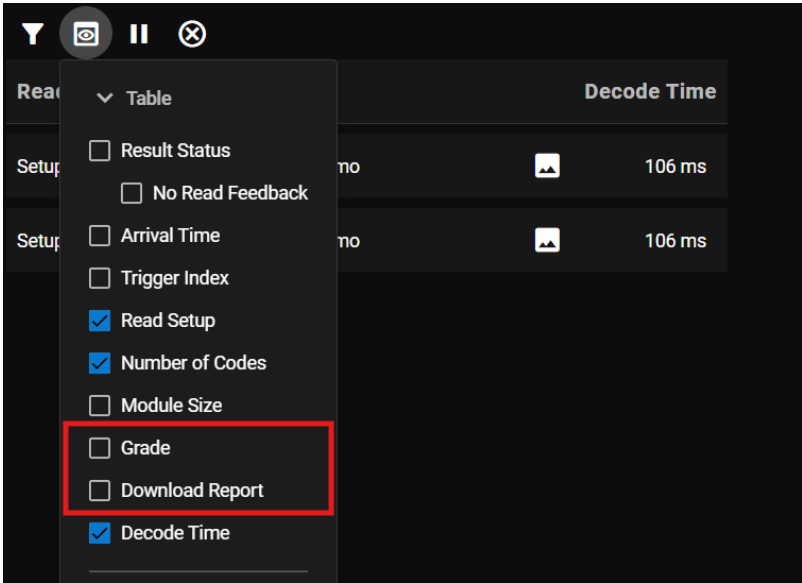
Trigger Index	Read Setup	# Codes	Result	Decode Time
125	Setup 3	1		113 ms

Trigger Index	Unused Error Correction (UEC)	100.0%	4.0
125	Symbol Contrast (SC)	80%	4.0
Overall Grade	Modulation (MOD)		4.0
4.0 (A)	Reflectance Margin (RM)		4.0
Custom	Axial Nonuniformity (ANU)	0.1%	4.0
Pass	Grid Nonuniformity (GNU)	18.1%	4.0
Symbology	Fixed Pattern Damage (FPD)	4.0	4.0
Data Matrix	Left 'L' Side (LLS)		4.0
	Bottom 'L' Side (BLS)		4.0
	Left Quiet Zone (LQZ)		4.0
	Bottom Quiet Zone (BQZ)		4.0
	Top Quiet Zone (TQZ)		4.0
	Right Quiet Zone (RQZ)		4.0
	Top Transition Ratio (TTR)	0%	4.0
	Right Transition Ratio (RTR)	0%	4.0
	Top Clock Track (TCT)		4.0

The tab shows the quality parameter grades in letter or number form. The format depends on the version of the standard. The following revisions replaced the letters, ranging from F to A, with numbers, ranging from 0.0 to 4.0 in increments of 0.1:

- ISO/IEC 15416:2016
- ISO/IEC 15415:2024
- ISO/IEC TR 29158:2020

With TruCheck™ enabled, the result table can display the grade of the trigger, and the verification report for the latest trigger can be downloaded. To see these columns in the result table, enable them in the **Display Configuration**:



Note:
When you download the verification report, consider the following:

- The download button only appears for the latest report and if the column is enabled.
- Depending on your browser settings, the browser might warn you that the download is not secure. This is because the reader uses HTTP, not HTTPS.

ISO/IEC 15416 Quality Parameters

The ISO/IEC 15416 standard defines the grading standards for 1D barcodes.

Trigger Index	Edge	0 F
154	Reflectance Light / Reflectance Dark (RI/Rd)	0/0 F
Overall Grade	Symbol Contrast (SC)	-
0.0 (F)	Minimum Edge Contrast (MinEC)	-
GS1	Modulation (MOD)	74% A
Fail (Quality)	Defect (Def)	80% F
Symbology	Decode (DCD)	0/10 F
Code 128	Decodability (DEC)	0% F
	Minimum Quiet Zone (MinQZ)	1 F
Statistics Full Result TruCheck™		

The following table lists and details the quality parameters for ISO/IEC 15416 that the **TruCheck™** tab shows in the **Runtime** step.

Quality Parameter	Parameter Description	Value Description	Grading
Edge	Edge determination. The parameter checks that the reader detects each edge between bars and spaces. The edges can blur due to splitting of bars or ink splattered into spaces.	The Edge value is the number of bars and spaces without quiet zones.	Pass/Fail grade (A or F , respectively). Pass grade if the number of bars and spaces is correct.
Reflectance Light / Reflectance Dark (RI/Rd)	The ratio of minimum reflectance. The parameter checks whether the darkness of the bars is sufficient. The light reflected by the bars must be less than half of the light reflected by the spaces.	The first value represents the lightest value in spaces. The second value represents the darkest value in bars.	Pass/Fail grade (A or F , respectively). Pass grade if the reflectance is sufficient.
Symbol Contrast (SC)	The parameter measures the contrast between the brightest space and the darkest bar.	The value is a percentage of contrast. It ranges from 100, which is the highest contrast, to 0, which is the lowest contrast.	A > 70 B 55 – 69 C 40 – 54 D 20 – 39 F < 20
Minimum Edge Contrast (MinEC)	The parameter checks the contrast between bars and spaces. As opposed to symbol contrast, minimum edge contrast checks for the worst contrast difference between each bar and space.	The value is a percentage of edge contrast.	Pass/Fail grade (A or F , respectively). Pass grade if the value exceeds 15%.
Modulation (MOD)	The parameter checks the edge contrast as the ratio of the minimum edge contrast and symbol contrast. It represents the consistency of the darkness of the bars and the lightness of the spaces.	The value is a percentage of modulation.	A >= 70 B 60 – 69 C 50 – 59 D 40 – 49 F < 30
Defect (Def)	The parameter checks for defects in bars and spaces. Out of all the changes in reflectance in bars and spaces, Defect takes the highest value, which is the worst one and could influence the readability of the code the most.	The value is a percentage of the worst change in reflectance of a single bar or space.	A <= 15 B 16 – 20 C 21 – 25 D 26 – 30 F > 30
Decode (DCD)	The parameter checks whether the code can be decoded based on a specific formula.	The value is a score out of ten.	Pass/Fail grade (A or F , respectively).

Quality Parameter	Parameter Description	Value Description	Grading
Decodability (DEC)	The parameter determines how accurate the bar and space widths are and how easy it is to determine the widths. The lower the score, the more inconsistent and distorted the bar and space widths are.	The value is a percentage of the overall tolerance range for a bar or space range that is accurate.	A >= 62 B 50 – 61 C 37 – 49 D 25 – 36 F < 25
Minimum Quiet Zone (MinQZ)	The quiet zone represents the empty spaces to the left and right of the code. Different symbologies have different requirements for the minimum quiet zone.	The value is the width of the blank spaces on either side of the code, measured in modules.	Pass/Fail grade (A or F, respectively). Pass grade depends on the symbology.

ISO/IEC 15415 Quality Parameters

The ISO/IEC 15415 standard defines the grading standards for 2D codes.

Trigger Index 125	Unused Error Correction (UEC)	100.0%	4.0
	Symbol Contrast (SC)	80%	4.0
Overall Grade 4.0 (A)	Modulation (MOD)		4.0
	Reflectance Margin (RM)		4.0
Custom Pass	Axial Nonuniformity (ANU)	0.1%	4.0
	Grid Nonuniformity (GNU)	18.1%	4.0
Symbology Data Matrix	Fixed Pattern Damage (FPD)	4.0	4.0
	Left 'L' Side (LLS)		4.0
	Bottom 'L' Side (BLS)		4.0
	Left Quiet Zone (LQZ)		4.0
	Bottom Quiet Zone (BQZ)		4.0
	Top Quiet Zone (TQZ)		4.0
	Right Quiet Zone (RQZ)		4.0
	Top Transition Ratio (TTR)	0%	4.0
	Right Transition Ratio (RTR)	0%	4.0
	Top Clock Track (TCT)		4.0

The following table lists and details the quality parameters for ISO/IEC 15415 that the TruCheck™ tab shows in the Runtime step.

Quality Parameter	Parameter Description	Value Description	Grading
Unused Error Correction (UEC)	Measures the remaining capability for error correction that is available for further incorrect modules.	The value is the percentage of the remaining error correction capability.	A > 62 B 50 – 61 C 37 – 49 D 25 – 36 F < 25
Symbol Contrast	The parameter measures the contrast between the brightest space and the darkest bar.	The value is a percentage of contrast. It ranges from 100, which is the highest contrast, to 0, which is the lowest contrast.	A > 70 B 55 – 70 C 40 – 55 D 20 – 40 F < 20
Print Growth (PG)	Measures the deviation from the original, intended sizes of the modules.  Note: The Print Growth (PG) quality parameter is only present from ISO/IEC 15415:2024.	A percentage of deviation from the original size of modules.	4.0 <= 10 3.9 – 10.1 – 3.0 28 2.9 – 28.1 – 2.0 34 1.9 – 34.1 – 1.0 40 0.9 – 40.1 – 0.1 60 0.0 > 60
Modulation (MOD) and Reflectance Margin (RM)	Measures the amount of reflectivity of the modules. The grade is calculated using a formula that compares the reflectivity of each module to the global threshold and the overall symbol contrast.  Note: The Reflectance Margin (RM) quality parameter is not displayed in ISO/IEC 15415:2024.	A percentage of modulation.	A > 50 B 40 – 49 C 30 – 39 D 20 – 21 F < 20
Axial Nonuniformity (ANU)	Measures the overall aspect ratio of the symbol and checks for uneven scaling of the modules.	A percentage of deviation from a square shape of the modules.	A <= 6 B 6 – 7 C 8 – 9 D 10 – 11 F >= 12

Quality Parameter	Parameter Description	Value Description	Grading
Grid Nonuniformity (GNU)	Measures the biggest deviation from the calculated center of the module and the ideal location for the center of the module.	Deviation from the center in percentage of module size.	A <= 38 B 39 – 50 C 51 – 63 D 64 – 74 F >= 75
Fixed Pattern Damage (FPD)	The overall grade of the fixed pattern components.	N/A	It is equal to the lowest grade among LLS, BLS, LQZ, BQZ, TQZ, and RQZ.
Left 'L' Side (LLS)	Measures imperfections in the 'L' finder pattern. The parameter checks the pattern for the following: - The gaps must be three modules or less. - The gaps must be separated by at least four correct modules.	The percentage of incorrect modules.	A 0 B 1 – 9 C 10 – 13 D 14 – 17 F < 17
Bottom 'L' Side (BLS)	Measures imperfections in the 'L' finder pattern. The parameter checks the pattern for the following: - The gaps must be three modules or less. - The gaps must be separated by at least four correct modules.	The percentage of incorrect modules.	A 0 B 1 – 9 C 10 – 13 D 14 – 17 F < 17
Left Quiet Zone (LQZ)	Measure imperfections in the quiet zone, which is a one-module area on either side of the code.	The percentage of incorrect modules.	A 0 B 1 – 9 C 10 – 13 D 14 – 17 F < 17
Bottom Quiet Zone (BQZ)	Measure imperfections in the quiet zone, which is a one-module area on either side of the code.	The percentage of incorrect modules.	A 0 B 1 – 9 C 10 – 13 D 14 – 17 F < 17
Top Quiet Zone (BQZ)	Measure imperfections in the quiet zone, which is a one-module area on either side of the code.	The percentage of incorrect modules.	A 0 B 1 – 9 C 10 – 13 D 14 – 17 F < 17

Quality Parameter	Parameter Description	Value Description	Grading
Right Quiet Zone (RQZ)	Measure imperfections in the quiet zone, which is a one-module area on either side of the code.	The percentage of incorrect modules.	A 0 B 1 – 9 C 10 – 13 D 14 – 17 F < 17
Top Transition Ratio (TTR)	Measures the imperfections in the transition from the clock track to the adjoining quiet zone.	The ratio of the number of transitions from light to dark or dark to light in the quiet zone divided by the number of transitions in the clock track.	A < 6 B 7 – 8 C 9 – 10 D 11 – 12 F > 12
Right Transition Ratio (RTR)	Measures the imperfections in the transition from the clock track to the adjoining quiet zone.	The ratio of the number of transitions from light to dark or dark to light in the quiet zone divided by the number of transitions in the clock track.	A < 6 B 7 – 8 C 9 – 10 D 11 – 12 F > 12
Top Clock Track (TCT)	Measures the imperfections in the clock track. Three out of five modules in every five modules in the clock track must be correct for the passing grade.	N/A	A grade of the highest modulation level for which the test passes.
Right Clock Track (RCT)	Measures the imperfections in the clock track. Three out of five modules in every five modules in the clock track must be correct for the passing grade.	N/A	A grade of the highest modulation level for which the test passes.
Average Grade (AG)	Average grade of damage across the finder pattern.	The average value of the following: - Lowest of values associated with the clock track segment: TCT, TTR, TQZ, TCT, RTR, and RQZ. - LLS - BLS - LQZ - BQZ	A 4.0 B 3.5 – 3.9 C 3.0 – 3.4 D 2.5 – 2.9 F ≤ 2.5
Decode	The parameter checks whether the code can be decoded based on a specific formula.	The value is a score out of ten.	Pass/Fail grade (A or F, respectively).

ISO/IEC TR 29158 AIM-DPM-1-2006 Quality Parameters

The ISO/IEC TR 29158 standard is an extension of ISO/IEC 15415 for Direct Part Marking use case.

The following table lists and details the quality parameters for ISO/IEC TR 29158 that the **TruCheck™** tab shows in the **Runtime** step.

Quality Parameter	Parameter Description	Value Description	Grading
Cell Contrast (CC)	The relative contrast value between bars and spaces. It is the mean value of the light and dark elements.	A percentage of contrast value.	A ≥ 30 B 25 – 29 C 20 – 24 D 15 – 19 F < 15
Cell Modulation (CMOD)	Measures the amount of reflectivity of the modules. The grade is calculated using a formula that compares the reflectivity of each module to the global threshold and the overall symbol contrast. Each module is graded based on the reflectivity of the modules, then error correction capability is calculated to negate one or few elements with low values.	A percentage of modulation.	
Distributed Damage Grade (DDG)	Average grade of the fixed pattern elements.	N/A	The result of fixed pattern damage patterns.
Minimum Reflectance (MR)	The minimum reflectance between light and dark elements.	A percentage of the lowest value of reflectance between dark and light elements.	For ISO/IEC TR 29158:2011: Pass/Fail grade (A or F, respectively). The value must be at least 5% for a pass grade. For ISO/IEC TR 29158:2020 and later: 4.0 ≥ 20 3.5 – 15 – 4.0 19.9 2.5 – 10 – 3.4 14.9 1.5 – 5 – 2.4 9.9 1.4 – < 5 0.0

Quality Parameter	Parameter Description	Value Description	Grading
Decode	The parameter checks whether the code can be decoded based on a specific formula.	The value is a score out of ten.	Pass/Fail grade (A or F , respectively).

Settings

Overview

In the **Overview** tab of **Settings**, you can set up the language and theme of the WebUI. You can also customize the behavior of the in-application help.

Configure the language and appearance of the WebUI

☐  Deutsch
 ☒  English
 ☐  Español

☒  Language
 ☐  Français
 ☐  Italiano
 ☐  日本語

☐  한국어
 ☐  Polski
 ☐  Português do Brasil

☐  中文

☒  Theme
 ☒ COGNEX DARK 
☐ COGNEX LIGHT 

☒ <> Codepage Symbology Default 
☐ Ignore ECI Indicators

☒ ? Help
 ☐ Show walkthrough content on first visit

RESET SEEN WALKTHROUGHS

Configure Codepage

You can change the codepage to be able to decode non-ASCII characters.

Click **Symbology Default** to open the drop-down of available codepages and select one of the options. Read at least one non-ASCII code and use the **Decoded Result** column to help determine which codepage is the right one for your reading. You can also use the **Filter** field to narrow down the list of available options or disable the **Show Invalid Decodings** checkbox to only show valid codepages.

COGNEX

EXIT RUNTIME





Trigger Index	Read Setup	# Codes	Result	Decode Time
474	Setup 4	1	0'0%0°Ñ00%0μ0½Ñ00°Ñ... 	41 ms
473	Setup 4	1	0'0%0°Ñ00%0μ0½Ñ00°Ñ... 	40 ms
472	Setup 4	1	0'0%0°Ñ00%0μ0½Ñ00°Ñ... 	40 ms
471	Setup 4	1	0'0%0°Ñ00%0μ0½Ñ00°Ñ... 	40 ms
470	Setup 4	1	0'0%0°Ñ00%0μ0½Ñ00°Ñ... 	41 ms
469	Setup 4	1	0'0%0°Ñ00%0μ0½Ñ00°Ñ00,... 	40 ms

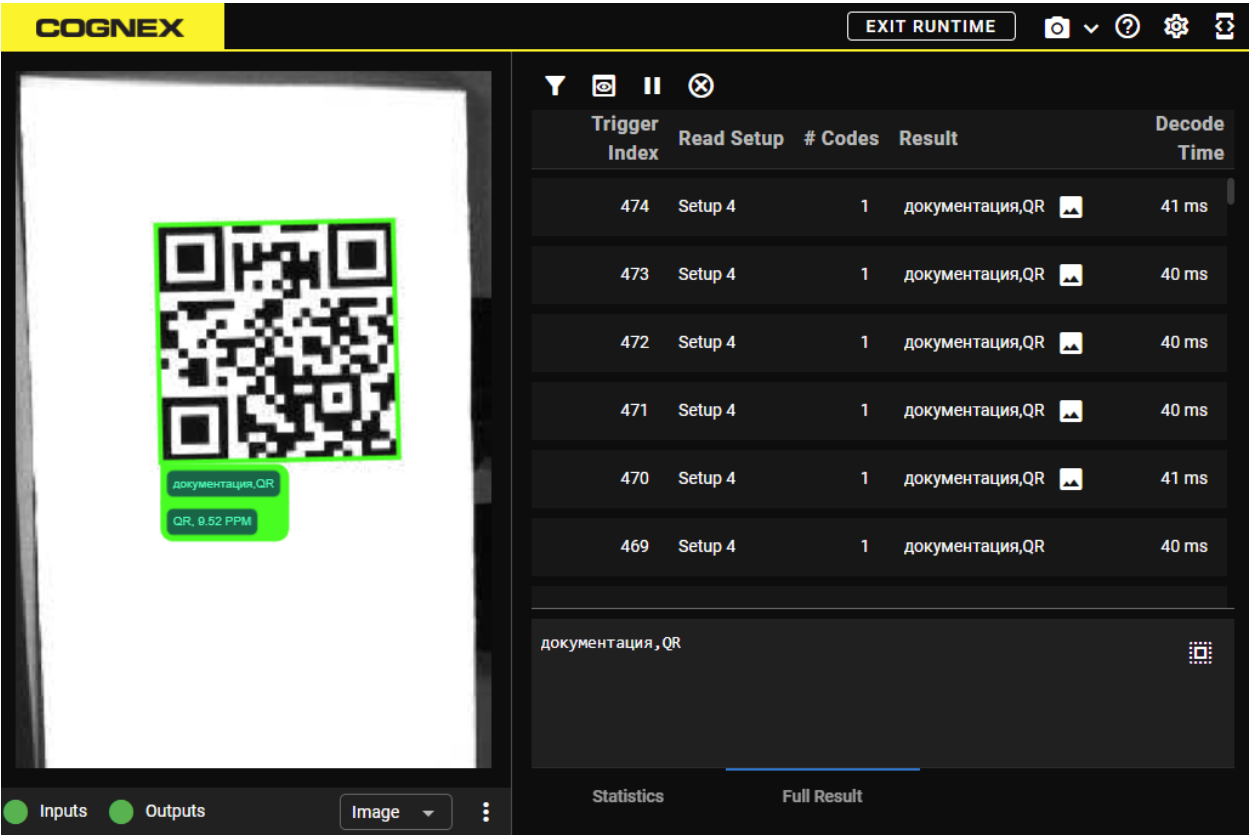
0'0%0°Ñ00%0μ0½Ñ00°Ñ00,QR

StatisticsFull Result

Inputs

Outputs

Image



By default, the WebUI recognizes Extended Channel Interpretation (ECI) indicators compliant with AIM ECI standards. When reading compliant codes, this allows the WebUI to automatically identify and use the correct codepage. Check the **Ignore ECI Indicators** box to disable support for ECI indicators, so the WebUI uses the configured codepage instead.

Note: The WebUI has limited support for non-codepage ECI indicators. The WebUI still attempts to display codes with these indicators, but correct interpretations are only possible in the expected downstream program.

Licenses

In the **Licenses** tab of **Settings**, you can view and manage the Feature Keys of your reader.

To get access to a feature, click **Create Feature License Request** and submit the file to Cognex. Click **Import New Feature License** to upload the returned key.

Current Licenses IDQuick, IDMax, BarCode, PostalCode, 2DCode, Validation, ImageDownload, Ethernet, FFP, Scripting, PowerGrid, ImageFiltering, Omnidirectional, LadderAndPicket, DirectLight

CREATE FEATURE LICENSE REQUEST

Import New Feature License 

Ready

SUBMIT FILE

Date & Time

In the **Date & Time** tab of **Settings**, you can configure date and time settings for your reader by clicking on **Enable Date & Time Settings**.

Configure the date and time settings of the device

Enable Date & Time Settings ☒

Current Time 2/17/2019, 6:16:51 AM

Timezone UTC 

SNTP Servers IP Address or Host Name

=

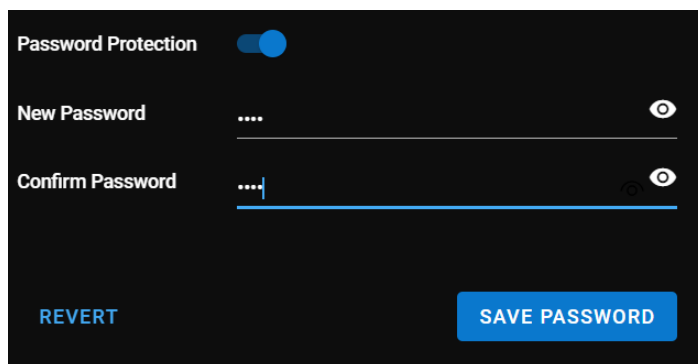
=

If you choose **SNTP Service**, you can configure clock time sources between your servers. Choose **Timezone** and enter the IP Addresses or Host Names of the servers.

 **Note:** You have to configure an SNTP server to add timestamps to the output string. For more information on the output string, see [Data Formatting on page 24](#).

Password Protection

In the **Password Protection** tab of **Settings**, you can set up a password for the reader. When you enable password protection and lock the WebUI, users cannot modify the reader configuration without a password. Use this feature to protect your settings from unwanted changes.

A screenshot of the 'Password Protection' settings interface. At the top, there is a toggle switch labeled 'Password Protection' which is currently turned on (blue). Below this, there are two input fields: 'New Password' and 'Confirm Password'. Both fields contain four dots, indicating masked text. To the right of each field is an eye icon; the 'New Password' icon is closed, and the 'Confirm Password' icon is open. At the bottom left is a 'REVERT' button, and at the bottom right is a blue 'SAVE PASSWORD' button.

Turning on Password Protection

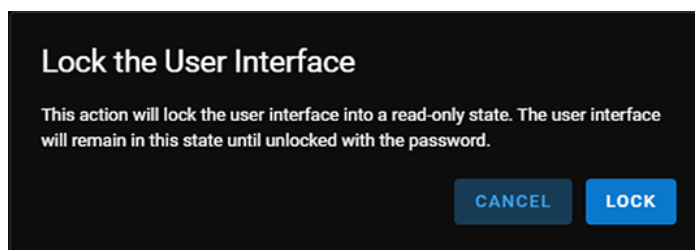
To turn on password protection:

1. Enable the **Password Protection** toggle.
2. Enter the new password in the **New Password** and **Confirm Password** fields.
3. Click **Save Password**.

Locking the User Interface

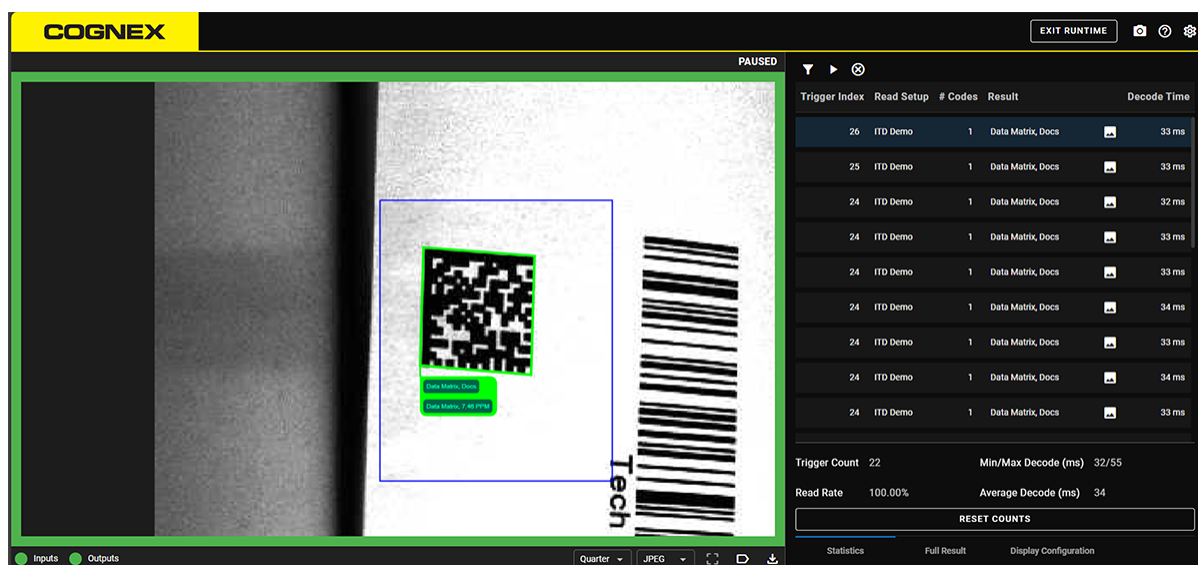
To lock the user interface:

1. Turn on password protection.
2. Click on the lock icon at the top right of the screen. The **Lock the User Interface** dialog appears.

A screenshot of the 'Lock the User Interface' dialog box. The title 'Lock the User Interface' is at the top. Below it, a message states: 'This action will lock the user interface into a read-only state. The user interface will remain in this state until unlocked with the password.' At the bottom right, there are two buttons: a grey 'CANCEL' button and a blue 'LOCK' button.

- Click **Lock**.

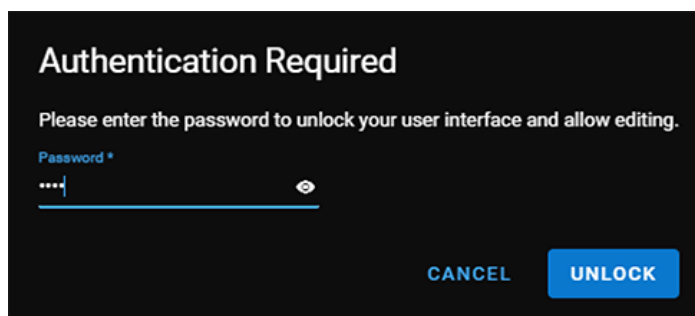
The user interface is locked, and you can only view the **Runtime** step.



To unlock the user interface:

- Exit the **Runtime** step.

The **Authentication Required** dialog appears:



- Fill in the **Password** field and click **Unlock**.

Note: The WebUI locks you out after a few minutes of inactivity.

Forgotten Password

Reset your DataMan reader to factory settings.

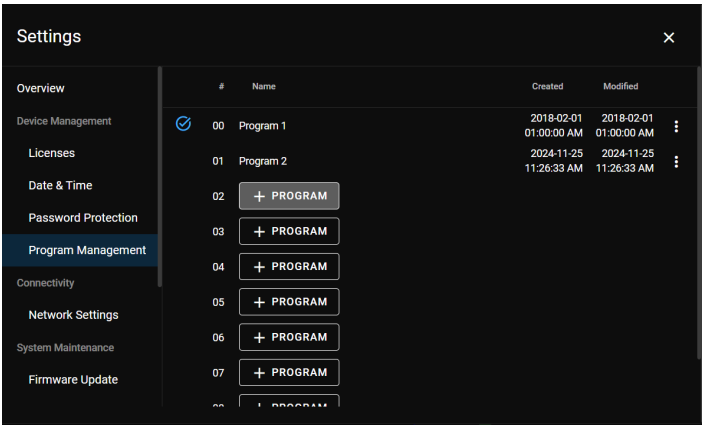
Removing the Password

Disable the **Password Protection** toggle, then click **Turn off** on the pop-up.

Program Management

The **Program Management** tab of **Settings** allows you to see information about and perform actions relating to your programs. A blue checkmark appears next to the current active program.

A program is a collection of device-level configuration settings, including one or more read setups, application details, and communication setup.



The following actions are available by clicking on the **three dots** next to each program:

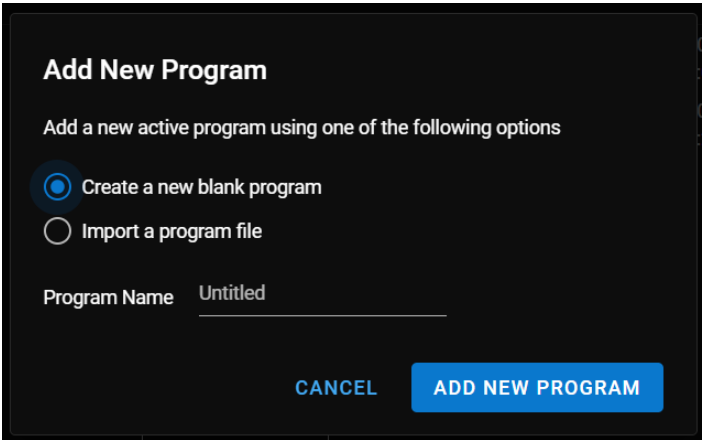
- Set as active
- Rename
- Clone
- Swap slots
- Export
- Delete

When cloning a program, the program name is kept by default, and you can change the name in the cloning window. Multiple programs can have the same name.

You can see the slot number, example images, and creation and modification dates of your previously created programs. The programs are sorted by slot number in ascending order by default, but you can click on any column to sort the programs by that value. The arrow next to **#** (slot number), **Name**, **Created**, or **Modified** signifies what metric the programs are sorted by and whether it is in ascending or descending order.

Add New Program

You can create a new program by clicking **+ Program** and choosing an **Add New Program** option:



The following options are available when adding a new program:

- **Create a new blank program:** the new program has all the default settings and no program name. You can change the default name by typing in the **Program Name** field.
- **Import a program file:** choose and import a previously exported program file. The new program copies all settings, including program name and the creation date of the exported program.

Click **Add New Program** to create the program.

Program Management with DMCC

You can also set the active program with the PROGRAM.ENABLE command, using the following syntax:

Name	Type	Range	Meaning
slot	int	[0-19]	program slot

For example, use the following command to update the current program to program slot 1:

```
SET PROGRAM.ENABLE 1
```

Network Settings

Note:

This setting option is only available on the following reader models:



- DataMan 80
- DataMan 280
- DataMan 290

In the **Network Settings** tab of **Settings**, you can configure network settings for your reader. To save network settings, click on **Apply**.

Configure network settings for the device

Host Name

DHCP ☒

IP Address

Subnet Mask

Default Gateway

DNS Server

Domain Name

Ready

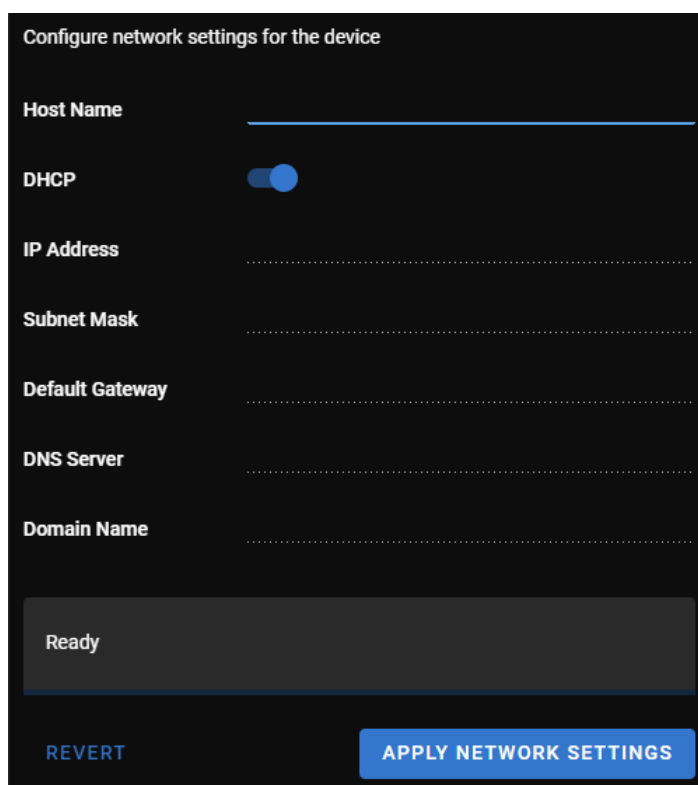
REVERT **APPLY NETWORK SETTINGS**

Ethernet 0 Settings



Note: This setting option is only available on DataMan 390 readers.

In the **Ethernet 0 Settings** tab of **Settings**, you can configure network settings for the ENET 0 port of your reader.



Configure network settings for the device

Host Name

DHCP ☒

IP Address

Subnet Mask

Default Gateway

DNS Server

Domain Name

Ready

[REVERT](#) [APPLY NETWORK SETTINGS](#)

To save network settings, click on **Apply**.

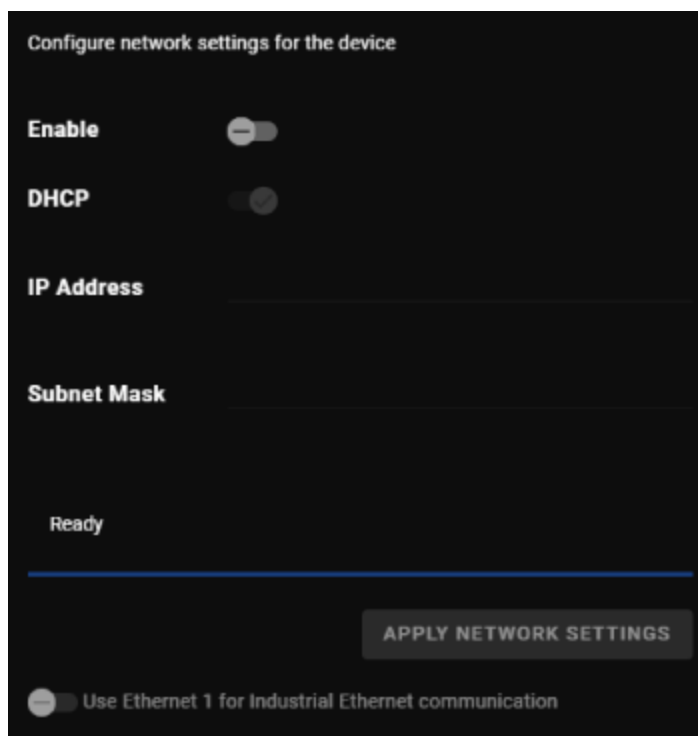
Note: When using two Ethernet ports, you cannot assign both to a DHCP server. If you try to set both ports to DHCP, the reader automatically disables the ENET1 port.

By default, the reader uses the ENET 0 port for Industrial Ethernet, and data communication and configuration. If you enable industrial protocols on the [Ethernet 1 Settings on page 91](#) tab, the reader automatically disables industrial protocol communications on the ENET 0 port.

Ethernet 1 Settings

Note: This setting option is only available on DataMan 390 readers.

In the **Ethernet 1 Settings** tab of **Settings**, you can configure network settings for the ENET 1 port of your reader.



The screenshot shows a dark-themed configuration window titled "Configure network settings for the device". It contains several settings: an "Enable" toggle switch (currently off), a "DHCP" toggle switch (currently on), and input fields for "IP Address" and "Subnet Mask". Below these is a "Ready" status indicator. At the bottom right is a button labeled "APPLY NETWORK SETTINGS". At the bottom left is another toggle switch labeled "Use Ethernet 1 for Industrial Ethernet communication", which is currently off.

Click the **Enable** toggle to enable network communication on the ENET 1 port. To save network settings, click on **Apply**.

Note: When using two Ethernet ports, you cannot assign both to a DHCP server. If you try to set both ports to DHCP, the reader automatically disables the ENET1 port.

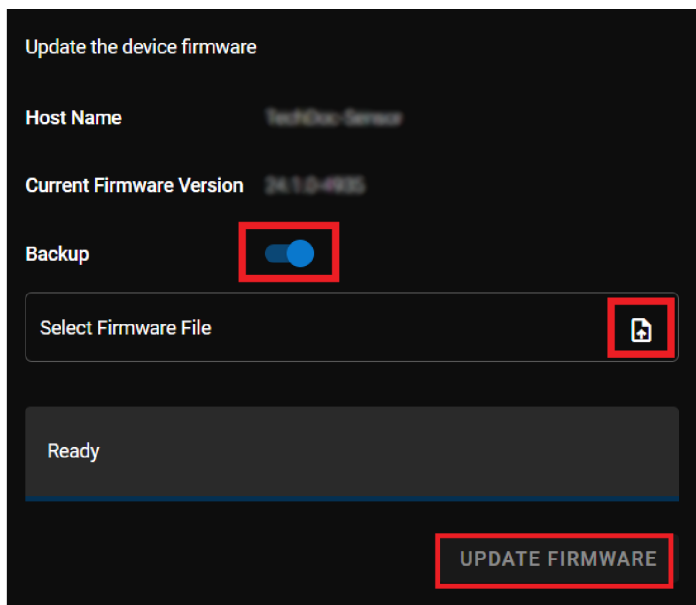
You can configure global Ethernet settings, including host name, default gateway, DNS server, and domain name on the [Ethernet 0 Settings on page 90](#) tab.

By default, the reader uses the ENET 0 port for Industrial Ethernet, and data communication and configuration. Click the **Use Ethernet 1 for Industrial Ethernet communication** toggle to enable industrial protocols on the ENET 1 port. This setting automatically disables industrial protocol communications on the ENET 0 port.

Firmware Update

In the **Firmware Update** tab of **Settings**, you can update the firmware of your reader by uploading the firmware file from your PC and then clicking on **Update Firmware**.

Turn on the **Backup** toggle to save a backup file before the update.



Services & Ports

In the **Services & Ports** tab of **Settings**, you can enable or disable major APIs and interfaces of the reader. Disabling unused interfaces does not affect normal reader operation. The list of configurable interfaces depends on your reader model and configuration.

Note: Disabling most services might leave physical access as the only method to access the reader. Follow the factory reset procedure to access a hardened device with a lost password.

Enable or disable device functions

Service Name ↑	Enabled	Configurable Ports
DMCC	☒	
Enterprise Connectivity	☒	
IP Address Recovery	☐	
RP Codes	☒	
Scripting	☐	
Serial	☒	
Serial Over USB	☒	
SetupTool	☒	

Ready

REVERT
APPLY SETTINGS

You can enable or disable the following functions:

Name	Action	Available in Emergency Boot Mode	Notes	Additional Documentation
DMCC	Enables DataMan Control Commands (DMCCs) for configuring and controlling your reader from a COM port.	No	DMCCs have non-compliant authorization control.	<i>Command Reference</i>
Enterprise Connectivity	Enables communication using the MQTT protocol.	No	N/A	N/A
IP Address Recovery	Enables you to change the IP address of the reader outside of the WebUI. For example, from the Repair and Support page of DataMan Setup Tool, or through Station Manager.	No	Uses port 1069.	<i>Troubleshooting a Network Connection</i> section of the <i>DataMan Communications and Programming Guide</i>
RP Codes	Enables the use of Reader Programming Codes, which are special Data Matrix codes to configure your reader without the DataMan Setup Tool or the WebUI.	Yes	N/A	<i>DataMan Configuration Codes</i>

Name	Action	Available in Emergency Boot Mode	Notes	Additional Documentation
Scripting	Enables the execution of custom scripts uploaded to the reader.	Yes	Scripting is not available on the WebUI.	<i>Scripting</i> section of the <i>DataMan Communications and Programming Guide</i>
Serial	Enables serial communication.	Yes	Requires access to the physical lines on the M12 connector.	N/A
Serial Over USB	Enables serial communication through USB with a virtual COM port.	Yes	Requires a USB cable connected to the reader.	<i>Emulating Serial Functionality</i> in the <i>Reference Manual</i> of your reader
SetupTool	Enables communication with the reader through the DataMan Setup Tool.	No	DataMan Setup Tool has non-compliant authorization control.	<i>Setup Tool Reference Manual</i>
HTTP	Enables access to the WebUI over an unsecured HTTP connection.	No	N/A	N/A
HTTPS	Enables access to the WebUI over an encrypted HTTPS connection.	No	N/A	N/A

Backup Device

In the **Backup** tab of **Settings**, you can save the reader configuration to a backup file by clicking on **Save Backup**. The backup file includes the following settings:

- All programs
- Global communication settings
- **Certificates** settings

Save the device configuration to a backup file

Host Name

Firmware Version

Ready

SAVE BACKUP

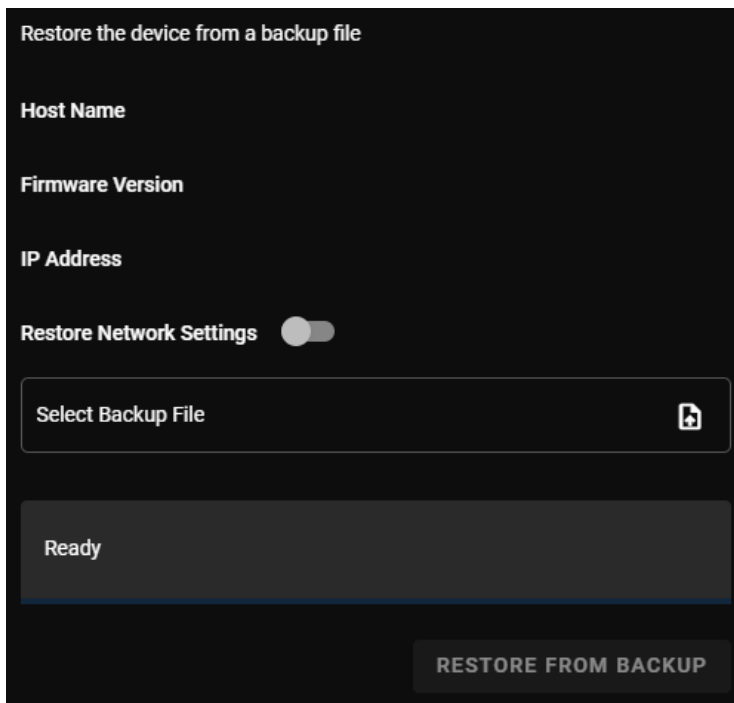
Restore Device

In the **Restore Device** tab of **Settings**, you can restore the reader from a backup file by selecting your backup file and clicking on **Restore**.

Enable the **Restore Network Settings** toggle if you want to override the current network settings of the reader with the network settings of the backup file:

- Global communication settings
- [Certificates on page 98](#) settings, including TLS certificates

 **Note:** Trusted certificates are always restored, even if you disable the **Restore Network Settings** toggle.



The screenshot shows the 'Restore the device from a backup file' interface. It includes fields for 'Host Name', 'Firmware Version', and 'IP Address'. Below these is a 'Restore Network Settings' toggle switch, which is currently turned off. A 'Select Backup File' button with a file icon is present. At the bottom, there is a 'Ready' status bar and a 'RESTORE FROM BACKUP' button.

 **Tip:** If restoring your reader changes the IP address of the reader, your connection to the reader is lost when the restore process finishes. You can find the new IP address of the reader using device discovery in Station Manager.

Factory Reset

In the **Factory Reset** tab of **Settings**, you can reset the reader to factory settings by clicking **Factory Reset**.

Reset the device to factory defaults

Host Name

Firmware Version

Ready

FACTORY RESET

Note: If resetting your reader changes the IP address of the reader, your connection to the reader is lost when the reset process finishes. You can find the new IP address of the reader using device discovery in Station Manager.

Device Log

In the **Device Log** tab of **Settings**, you can access the device log of your reader.

View logged device events and history




CLEAR LOG
Filter

Level	Created ↓	Message
INFO	2025-06-18 13:23:17.590	Device log downloaded.
INFO	2025-06-18 13:23:17.574	Device log cleared.

The reader generates logged events with timestamps. The WebUI displays these events in the device log.

The reader fetches the device log automatically when you open the **Device Log** tab. Click **Refresh** to retrieve any updates manually, or the **Play** button to start receiving updates automatically.

The device log contains the following message types:

- **Info:** general information about the reader.
- **Warning:** a non-critical issue.
- **Error:** a problem that needs to be resolved.

Note: Error events trigger the error LED on the reader. Configure the pulse duration of the error LED on the **Outputs** substep.

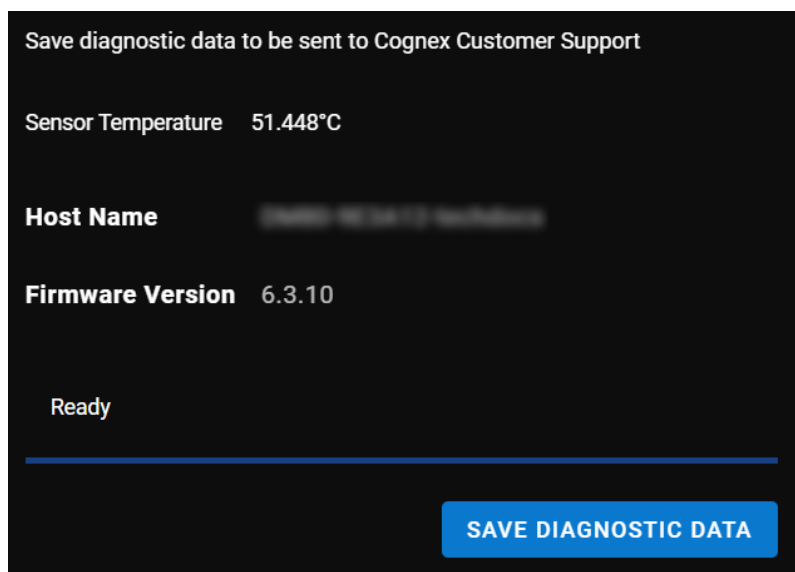
The following actions are available in the **Device Log** tab:

- **Download:** download the log in .csv format.
- **Refresh:** get the latest log entries.

- Clear: remove all entries from the log.
- Filter: input any string to filter the displayed log entries.
- Sort: click on a column header to sort the log entries in descending or ascending order.

Field Diagnostics

In the **Field Diagnostics** tab of **Settings**, you can save diagnostics data by clicking on **Save Diagnostic Data**. If necessary, you can send this data to Cognex Customer Support. The tab also displays the temperature of the image sensor on supported systems.



Save diagnostic data to be sent to Cognex Customer Support

Sensor Temperature 51.448°C

Host Name [blurred]

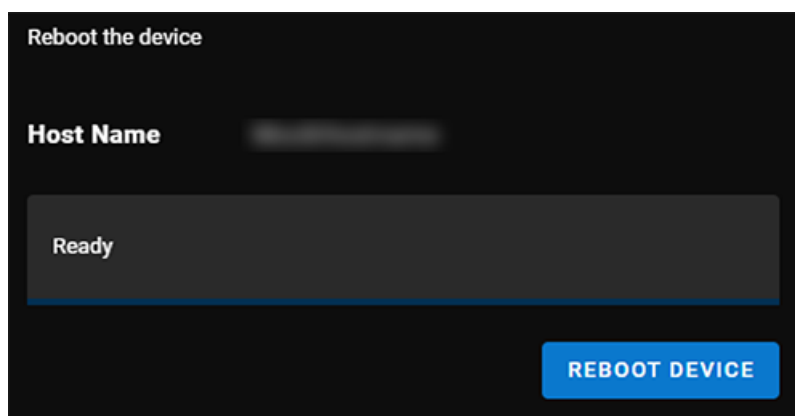
Firmware Version 6.3.10

Ready

SAVE DIAGNOSTIC DATA

Reboot

In the **Reboot** tab of **Settings**, you can reboot the device.



Reboot the device

Host Name [blurred]

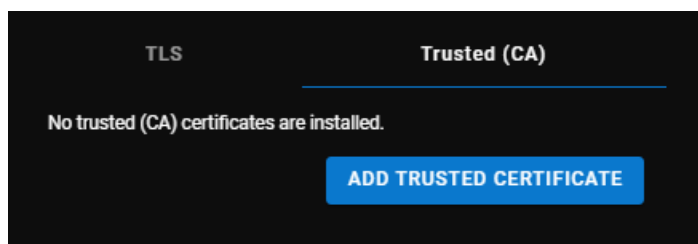
Ready

REBOOT DEVICE

After clicking **Reboot Device**, the screen confirms that the reader initiated the reboot. Rebooting takes a few seconds to finish.

Certificates

In the **Certificates** tab of **Settings**, you can install certificates on your reader for secure communication.



You can install the following types of certificates:

- **TLS:** allows other devices to trust your DataMan reader:
 - WebUI (HTTPS)
 - Enterprise Connectivity (MQTT with mTLS)
- **Trusted (Certificate Authority):** The self-signed certificate of the Certificate Authority (CA). It allows your DataMan reader to trust a server configured with a certificate signed by the same Certificate Authority:
 - Enterprise Connectivity (MQTT with TLS)

Note: The reader ships with the pre-installed TLS certificate file *default-tls.crt*. However, Cognex recommends installing your own certificate.

Prerequisites

Before installing any certificates, make sure that your reader and environment fulfill the following requirements:

- An NTP server is configured. Invalid time can cause issues when using or validating certificates.
- When using MQTT with TLS, you must have a local Certificate Authority set up for handling signing requests.

Install a TLS Certificate

You can install a TLS certificate in the following ways:

- The **New Signing Request** option allows you to create a private encryption key on your reader, and generate an accompanying signing request. Once that signing request is signed by a Certificate Authority to produce the certificate file, you can install the signed certificate file. This is the more secure option.
- The **Add Key Pair** option allows you to install an externally generated private key and signed certificate in one step. This option is less secure, but can be more convenient in certain environments.

New Signing Request

To install a TLS certificate with your own private encryption key:

1. Go to the **TLS** tab of the **Certificates** menu and click **New Signing Request** to create a new TLS certificate signing request.

The pop-up dialog automatically populates the required fields.

2. Optionally, enter the following additional signing request details:
 - Country Code
 - State
 - Locality
 - Company
 - Organizational Unit
3. Click **Generate Signing Request** to create the signing request.

A new certificate appears on the TLS tab of the dialog with **Pending Signing Request** status, and a `.csr` file is saved to your `Downloads` directory.
4. Generate a signed key for the signing request with your local Certificate Authority, and save the `.crt` file to an accessible location.
5. Click the menu icon for the certificate on the **TLS** tab, and select **Complete Signing Request**.
6. Click the **Select Certificate File** field to open the file browser, and select the `.crt` file with the signed key.

Add Key Pair

To install an externally generated private encryption key and signed certificate:

1. Go to the **TLS** tab of the **Certificates** menu and click **Add Key Pair**.
2. In the pop-up dialog that opens, click the **Select Certificate File** field to open the file browser, and select the `.der` file for the certificate you want to use.
3. Click the **Select Private Key File** field to open the file browser, and select the `.der` file you want to use as the private key.
4. In the **Password** field, enter the password for the private key file.
5. Click **Add Certificate + Key** to submit your certificate.

Assign TLS Certificate to HTTPS Connection

To assign a signed certificate to enable secure communication:

1. Click the menu icon for the certificate on the **TLS** tab and click **Assign Protocols**.
2. Enable the **HTTPS** toggle
3. Click **Assigned Protocols**.

 **Note:** You can only use one certificate for HTTPS connection at a given time.

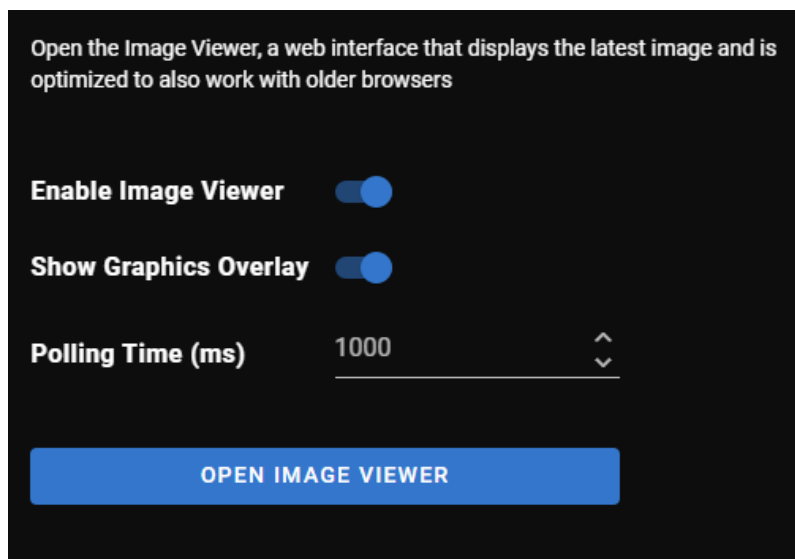
Install a Trusted Certificate

To add a trusted certificate for your Certificate Authority:

1. Go to the **Trusted (CA)** tab of the **Certificates** menu.
2. Click **Add Trusted Certificate**.
3. In the pop-up that opens, click the **Select Certificate File** field to open the file browser, and select the `.crt` file for the certificate you want to add.

Image Viewer

In the **Image Viewer** tab of **Settings**, you can configure and open the Image Viewer, which is accessible outside of WebUI. The Image Viewer is a HMI that displays the latest image captured by the reader with optional overlay graphics. The Image Viewer is optimized to also work with older browsers.



The **Polling Time (ms)** defines how often the web interface displays a new image. The minimum value is 500 ms. Modern browsers ignore the polling time and display the latest trigger.

