Zebra Aurora Focus 10

User Guide

MN-004665-10EN Rev A

Zebra Technologies | 3 Overlook Point | Lincolnshire, IL 60069 USA
zebra.com

Copyright

2026/04/03

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries ("Zebra Technologies"). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Contents

Contents

About this Guide

This guide details using Zebra Aurora Focus to configure, deploy, and manage Fixed Industrial Scanning and Vision System jobs.

Icon Conventions

The documentation set is designed to give the reader more visual clues. The following visual indicators are used throughout the documentation set.

NOTE: The text here indicates information that is supplemental for you to know and that is not required to complete a task.

IMPORTANT: The text here indicates information that is important for you to know.

CAUTION: If the precaution is not heeded, you could receive a minor or moderate injury.

WARNING: If danger is not avoided, you CAN be seriously injured or killed.

DANGER: If danger is not avoided, you WILL be seriously injured or killed.

Notational Conventions

The following notational conventions make the content of this document easy to navigate.

- **Bold** text is used to highlight the following:
  - Dialog box, window, and screen names
  - Dropdown list and list box names
  - Checkbox and radio button names
  - Icons on a screen
  - Key names on a keypad
  - Button names on a screen
- Bullets (•) indicate:
  - Action items

  - List of alternatives
  - Lists of required steps that are not necessarily sequential
- Sequential lists (for example, those that describe step-by-step procedures) appear as numbered lists.

Service Information

If you have a problem with your equipment, contact Zebra Global Customer Support for your region. Contact information is available at: zebra.com/support.

When contacting support, please have the following information available:

- Serial number of the unit
- Model number or product name
- Software/firmware type and version number

Zebra responds to calls by email, telephone, or fax within the time limits set forth in support agreements.

If your problem cannot be solved by Zebra Customer Support, you may need to return your equipment for servicing and will be given specific directions. Zebra is not responsible for any damages incurred during shipment if the approved shipping container is not used. Shipping the units improperly can possibly void the warranty.

If you purchased your Zebra business product from a Zebra business partner, contact that business partner for support.

Getting Started

This section provides information on requirements, installation, and supported languages for the application.

Minimum PC Requirements

Review the table below to ensure that your setup provides the requirements to support the application.

Table 1 Minimum Requirements

Item	Description
CPU	Intel or AMD 64-bit processor
Random Access Memory Files	4 GB (Recommended: 8GB)
GPU	GTX 1030, Intel UHD 620 or equivalent
Graphics Memory	2 GB (Recommended: 4GB)
Disk Space	4 GB free disk space
Monitor Resolution	1280 x 800 (Recommended: 1920 x 1080)
Operating System	Windows 10 or higher

Installation

Zebra Aurora Focus is available for download on the Zebra website: Zebra Aurora Focus Software Downloads. After downloading, install the application as the administrator and enable Zebra Aurora Focus in Windows Defender.

Supported Languages

Use Aurora Focus in other languages by selecting the desired language from the menu in the upper right corner of the home screen.

Additional Help and Resources

Access additional help and resources by clicking **Help** from the top menu.

Additional resources include:

- User Guides
- How-to Videos
- Tutorials
- Support Central
- Licensing Information
- About the Application

Zebra Aurora Focus Software Overview

Zebra Aurora Focus is a unified platform with an intuitive interface for setting up, deploying, and running Fixed Industrial Scanning or Vision System jobs to control enterprise-wide manufacturing and logistics automation solutions. This tool also can scale in support of new codes and increase scanning speed with the potential to upgrade to machine vision functionality via software license upgrade.

Zebra Aurora Focus Features

Zebra Aurora Focus provides several differentiating features to rapidly process, evaluate and compare multiple images in various lighting conditions without altering any hardware configurations.

Features include:

- QuickDraw – enables the user to draw right on an image to create a tool with minimum steps.
- Object Locate and Pattern Matching – Zebra's algorithms and intuitively crafted default settings enable users to consistently create and deploy efficient tools with less trial and error involved.
- Golden Image Compare – allows users to efficiently identify and resolve issues by comparing any image to an ideal image created at setup. This tool can significantly expedite troubleshooting activities by immediately diagnosing and correcting the source of degradation.
- Image Perfect - captures up to 16 different images, each with unique settings for focus, exposure, gain, and illumination control in one capture event.
- Deep Learning Optical Character Recognition (OCR) - uses Deep Learning tools to quickly read a variety of fonts within a user-defined region of interest (ROI) without training the tool with a large dataset.

UI Overview

The **Getting Started** screen features functionality such as viewing and configuring connected or emulated devices and settings for Fixed Industrial or Machine Vision scanning jobs.

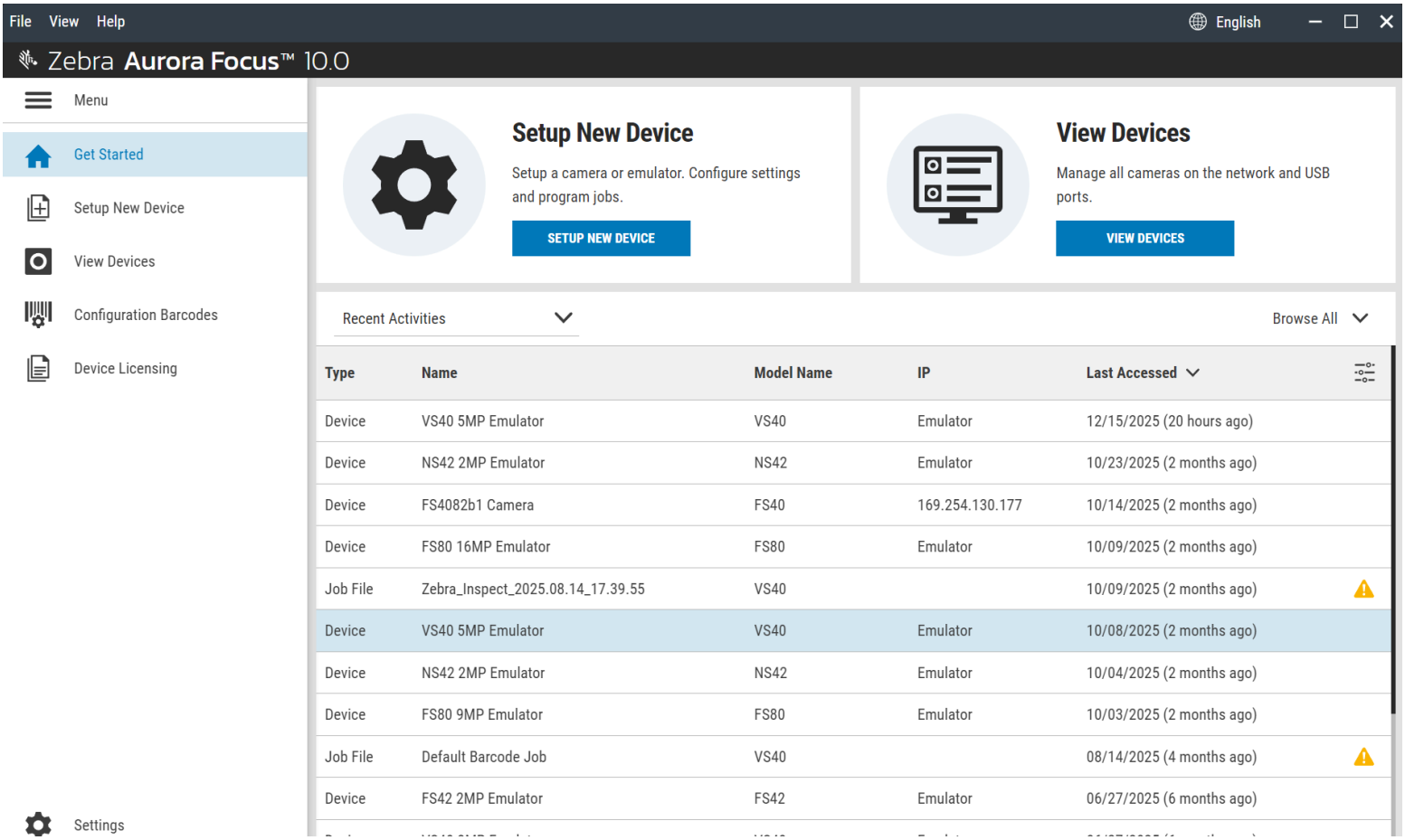


Table 2 Zebra Aurora Focus Home

Setting	Description
Setup New Device	Provides access to the configuration of existing (previously connected) and virtual (emulated) device settings and program jobs.
View Devices	View connected devices on your network.
Settings	Click **Settings** to manage localization and notification settings or clear the cache. **Enable Automatic Fixturing** options and **Image Capture Disk Management** settings on the **Job Management** tab.

Click the **Recent Activities** drop-down menu to sort by:

- Recent Activities
- Recent Jobs
- Recent Devices

Click the filter icon to filter by:

- Type
- Name
- Model Name
- IP
- Last Accessed

Application Settings

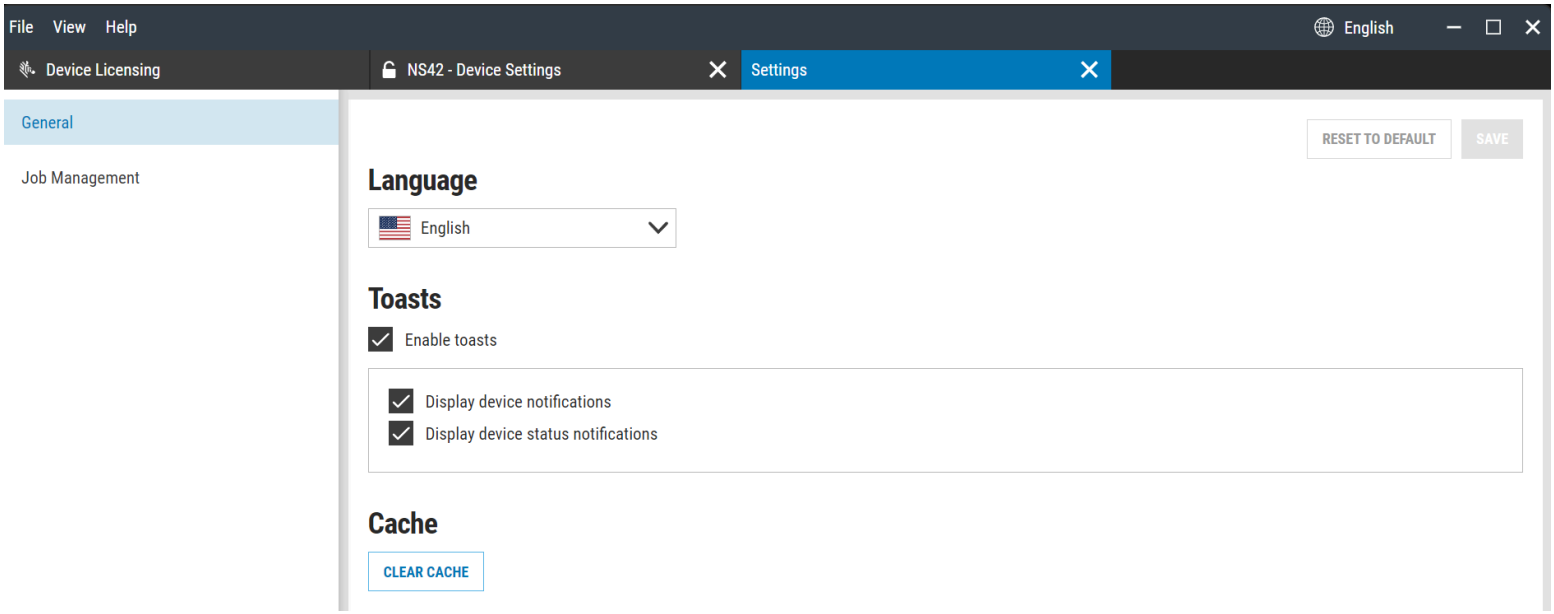
Click **File** to configure general application settings.

General Application Settings

Configure language, toasts, and cache settings.

Configure the following application settings using the General tab in Settings:

- Language - select a language for the UI text from the dropdown.
- Toasts - enable toast notifications for the device and device status.
- Cache - click **Clear Cache** to delete the cache on the device.



Job Management Settings

Configure fixturing, image capture, filmstrip management settings, and sensor cropping settings.

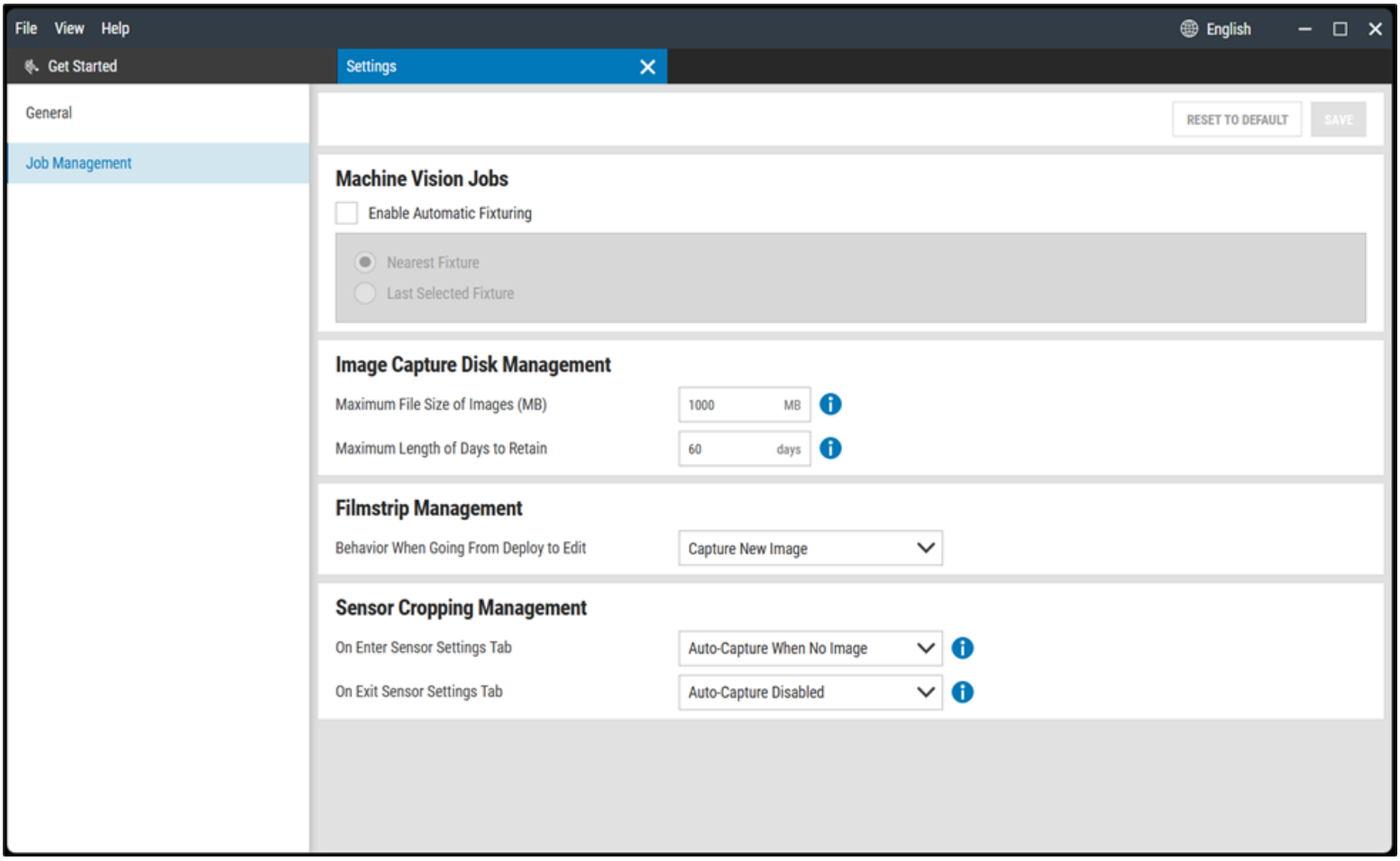
Configure the following application settings using the General tab in Settings:

- Machine Vision Jobs - enable fixturing to the nearest or last selected fixture.

- Image Capture Disk Management - determine the **Maximum Filesize of Images** (10-5000 mb, default: 1000) to save on the device and the **Maximum Length of Days to Retain** (1-1000 days, default: 60) the data on the device.
- Filmstrip Management - determine the behavior when going from Deploy to Edit mode:
  - Capture New Image
  - Use Last Filmstrip Image
  - Use Last Triggered Image

NOTE: Use Last Triggered Image attempts to download the last triggered image. If unavailable, this setting uses the last Filmstrip image as a fallback.

- Sensor Cropping Management - determine the behavior when going from Deploy to Edit mode:
  - On Enter Sensor Settings tab:
    - Auto-Capture When No Image: Auto-Capture on enter if no matching image is found.
    - Auto-Capture Enabled: Auto-Capture on every enter.
  - On Exit Sensor Settings tab:
    - Auto-Capture Disabled: No Auto-Capture
    - Auto-Capture Enabled: Auto-Capture on every enter
    - Auto-Capture on Change: Auto-Capture on exit when the Acquisition Region of Interest (AROI) has changed.



HID Keyboard Localization

Zebra Aurora Focus supports the following HID keyboard languages:

- English (North America)
- French (France) Windows
- German Windows
- Spanish (Spain) Windows
- Italian Windows
- English (UK) Linux
- German Linux
- Czech Linux
- Spanish (Mexico) Linux
- French (France) Linux
- Polish Linux
- Spanish (Spain) Linux
- Italian Linux

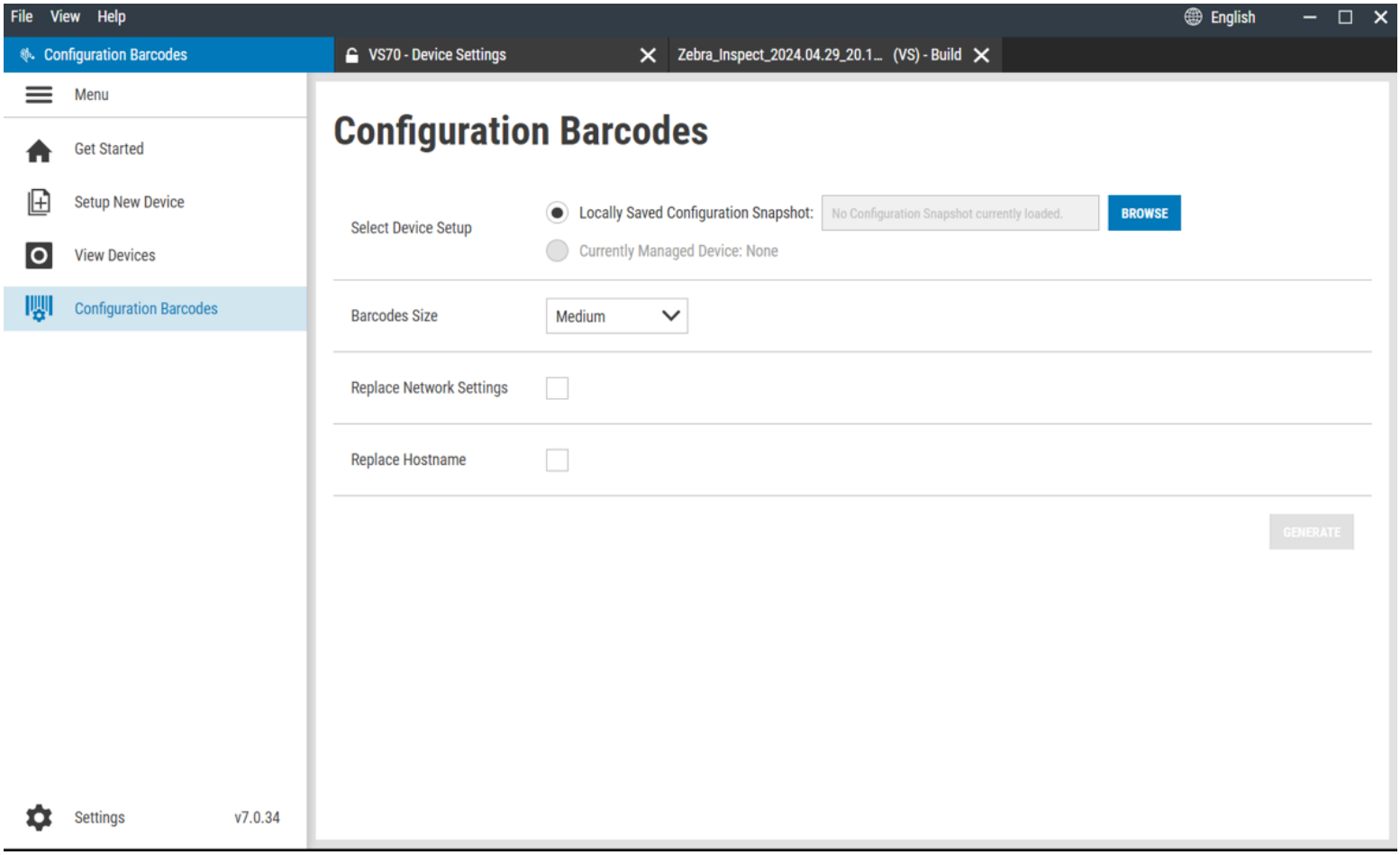
- Portuguese (Brazil) Linux
- Japanese Linux

Generating a Configuration Barcode

The **Configuration Barcodes** tab generates barcodes using a device configuration snapshot. Use an existing configuration snapshot or select the currently managed device setup to deploy to multiple cameras.

NOTE: Configuration barcodes are generated and applied to fixed scanners (FS) only. Vision scanners (VS) cannot generate or apply configuration barcodes.

1. Navigate to the **Configuration Barcodes** tab.
2. Select a **Device Setup** to create a snapshot from the current managed device or load one from the PC.



NOTE: If the device is not managed, you cannot select a setup from the device. Status is indicated next to the **Currently Managed Device** option.

NOTE: To create a saved configuration snapshot of a connected device, click **Create Configuration Snapshot** in **Devices Details.**

3. Configure the size of the barcode being exported. The options are Small, Medium, or Large. The default size is Medium.

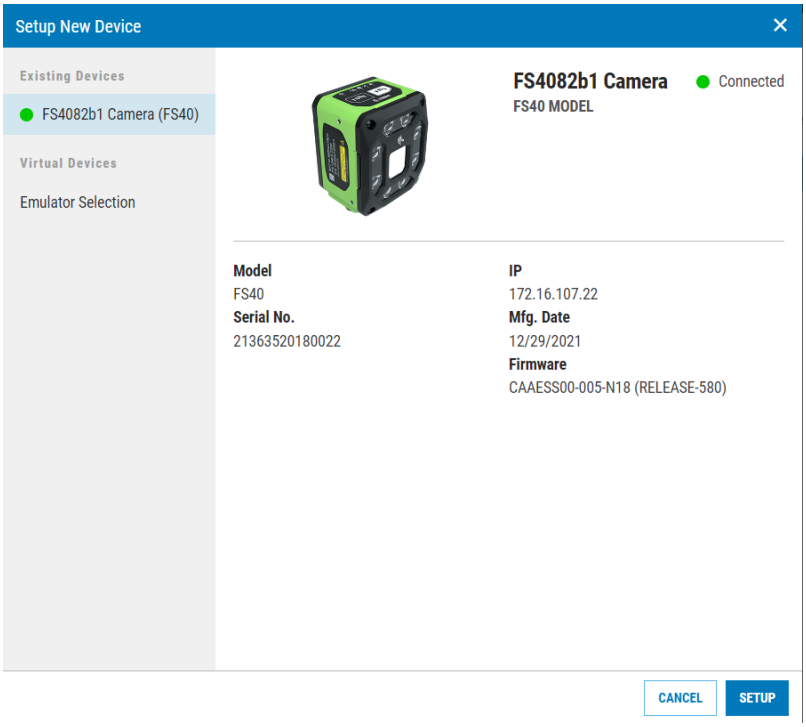
4. Enable **Replace Network Settings** to replace the network settings from the configuration file with the network settings of the new device onto which the configuration is being deployed.
5. Enable **Replace Hostname** to replace the hostname in the configuration file with the hostname of the new device that the configuration is being deployed onto.
6. Click **Generate**.

Device Discovery

Emulated devices are accessible under **Virtual Devices** on the **Setup New Device** screen. Devices that are physically connected to the system and are available to connect and set up are viewable under **Existing Devices**.

Setting Up a New Device

Click **Setup New Device** on the home screen to configure an FS or VS Emulator.



Select Emulator Selection and configure the settings for your emulator from drop-down menu.

Table 3 Setup New Device

Setting	Description
Type	Select the toolset type for the emulator.
Resolution	Select the desired resolution of the emulator.
Model	Select a device model of the emulator.
License	Select the license type for the emulator.

Adding Devices via IP Address

Connect to a camera by manually entering its IP address and clicking **Add Via IP Address** in the bottom right corner of the **View Devices** screen. Use this feature for devices with a known IP Address that are not automatically discoverable in Zebra Aurora Focus. You can also connect via IP address by entering the IP address (or hostname) into the **Add New Device via IP Address** form field and clicking **Connect**.

Updating Firmware on Multiple Devices

Select devices on the **View Devices** list to perform a firmware update.

Update devices using a file stored on an FTP/FTPS server or uploading the file to the device (default).

- Specify the host, username, password, file path, and if the server is running in FTPS mode, to use the FTP / FTPS Server option.
- Select a firmware file from the File Explorer to use the **File Based Upload** option.

Use **Dual Update** to update the primary device partition, reboot the device, update the secondary partition, and reboot the device again.

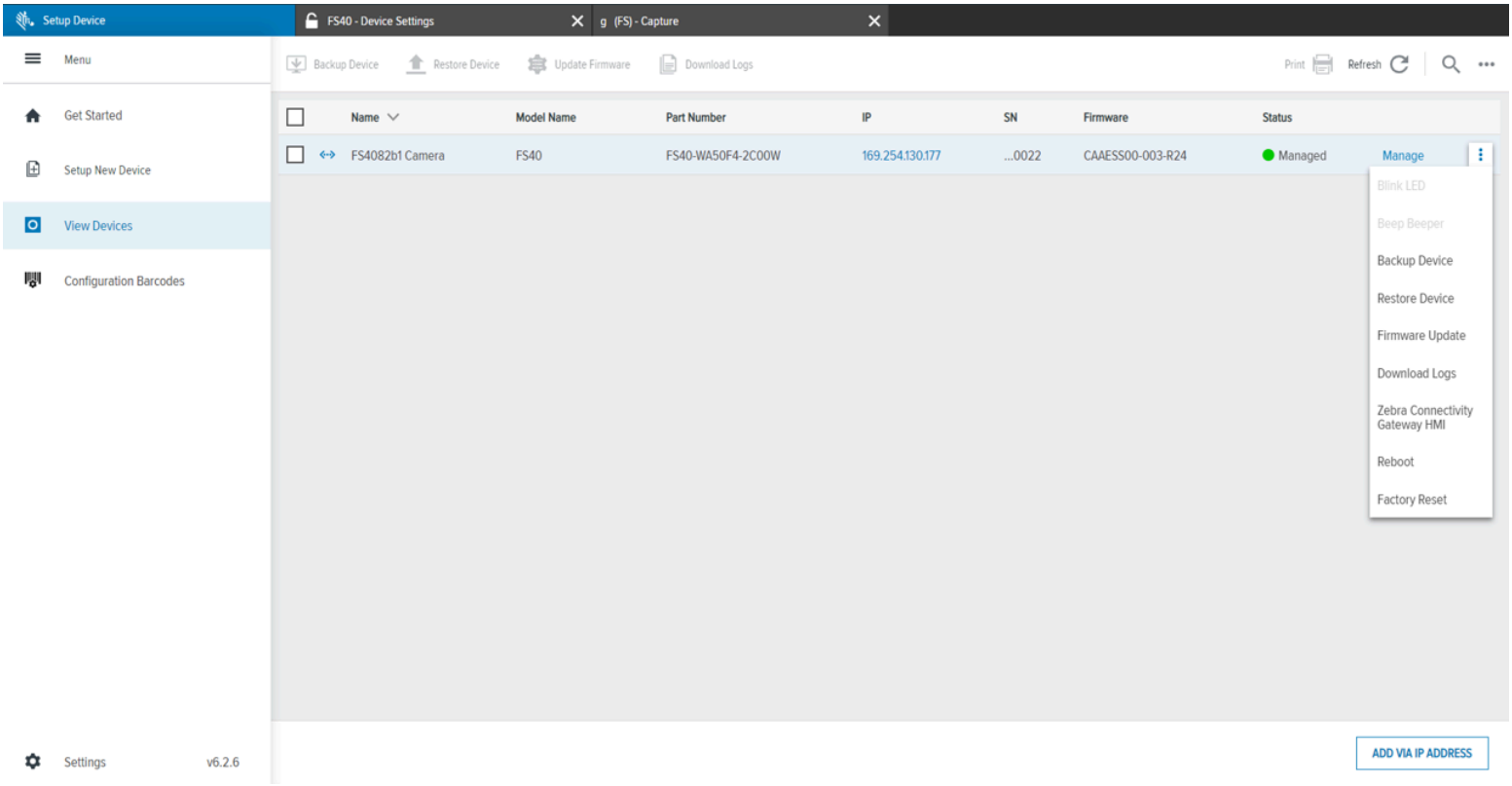
Use **Force Update** to apply the same firmware version to all devices.

Firmware updates are performed on Connected devices only.

NOTE: It is not possible to open more than one **Update Firmware** window.

Viewing Devices

Click **View Devices** to access additional device information such as name, part number, IP, serial number, firmware version, and status.



NOTE: Click the device IP address to access the Zebra Web HMI.

Table 4 Device Information

UI Element	Description
Backup Device	Create a backup of the selected device. The resulting file is an encrypted archive that can be used only by the restore function.
Restore Device	Select a backup file and restore the device configuration. The process also triggers a complete reboot of the device. **WARNING:** Applying a backup from one model type to another can create some issues.
Update Firmware	Use this command to update the device firmware. This operation can also be done using the Web HMI.
Download Logs	This command refreshes the list of discoverable devices on the network.
Zebra Connectivity Gateway HMI	Access the Zebra Connectivity Gateway HMI.
Reboot	Reboot the device.
Factory Reset	Perform a factory reset on the device.

Click on the device row to view additional device information.

NOTE: Copy the device's Serial Number, Part Number, and Firmware Version before contacting technical support.

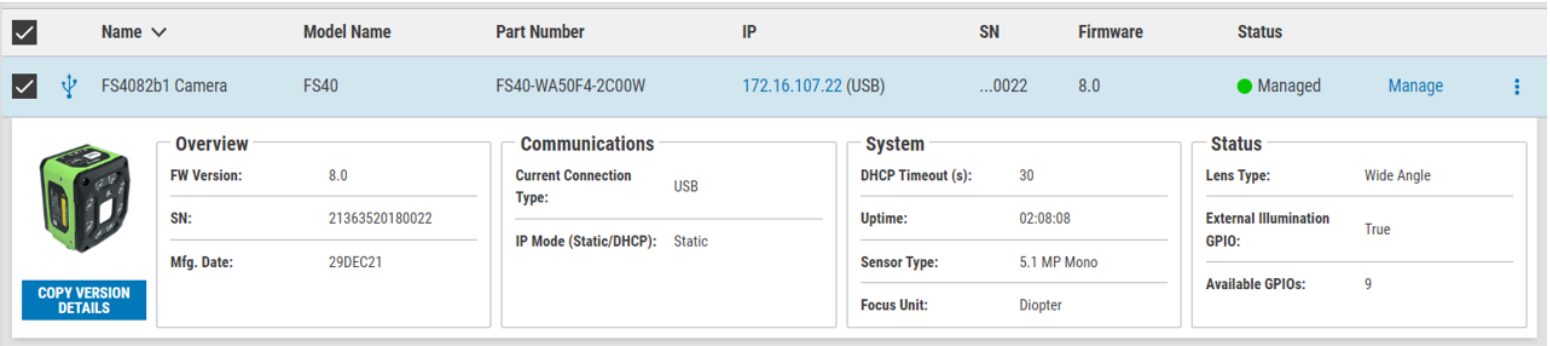


Table 5 Device Information

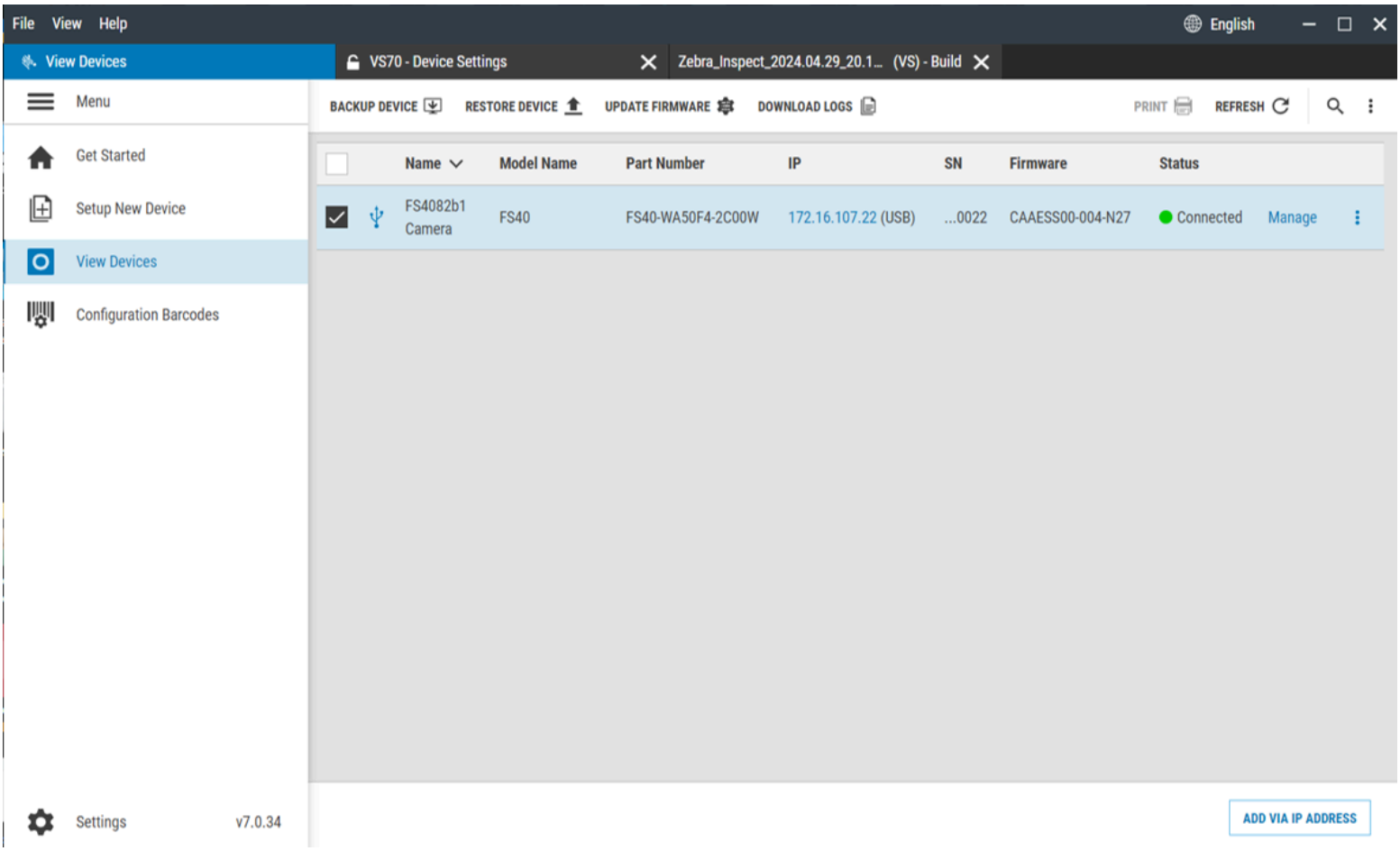
UI Element	Description
Overview	View device information, including firmware version, serial number, manufacturing date, core services version, and ChiCore library version.
Communications	View communication settings, including connection type and IP mode.

Table 5 Device Information (Continued)

UI Element	Description
System	View system settings, including DHCP timeouts, uptime, sensor type, and firmware version, and focus calibration type.
Status	View lens and illumination settings, including lens type, illumination type, external illumination, external illumination GPIO, and number of available GPIOs.

Multi-Device Management

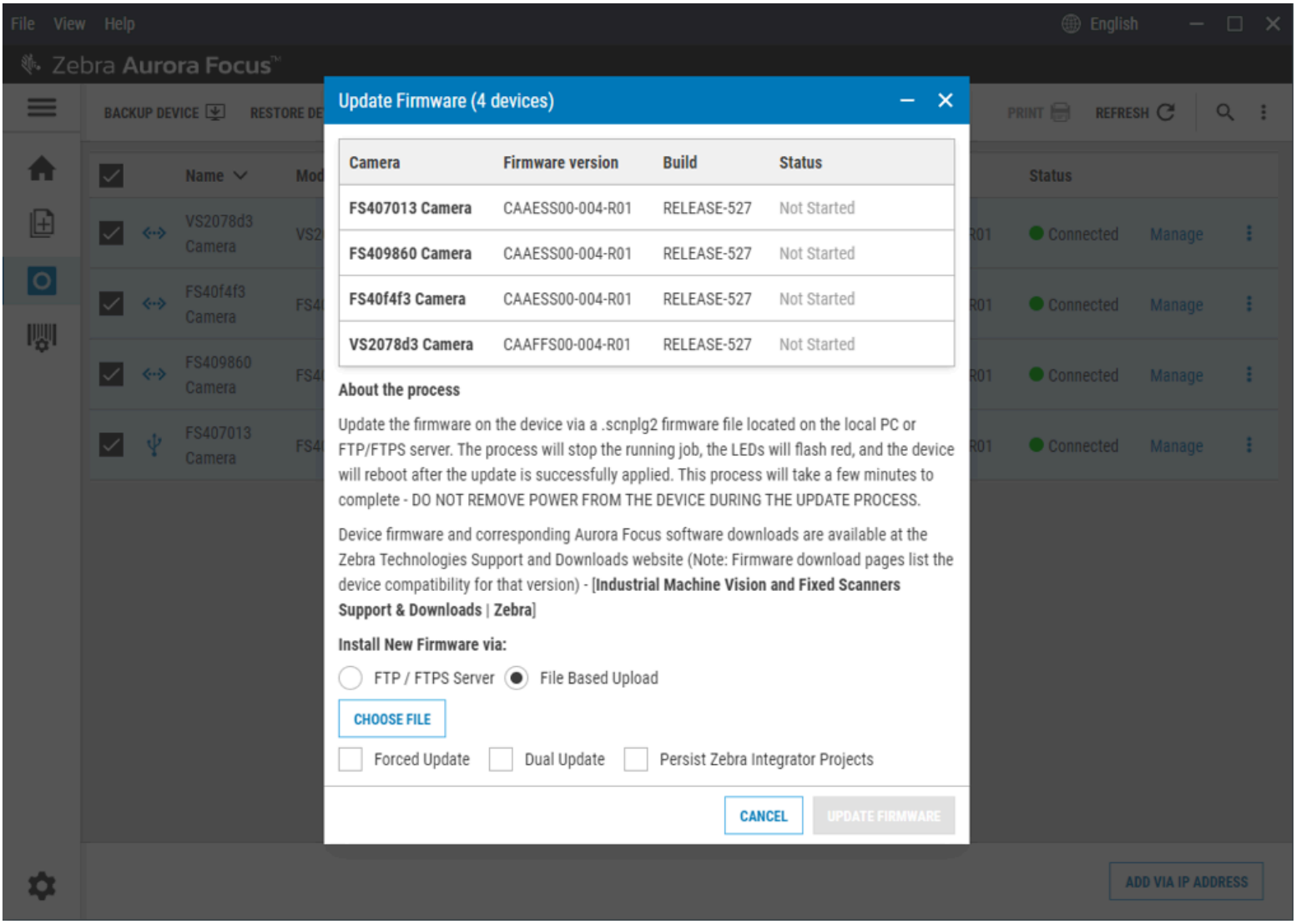
Manage multiple cameras from Aurora Focus using **View Devices** list.



Backing Up Multiple Devices

1. Select a device from the **View Devices** list to back up a device configuration, jobs, or both (default).
2. Click **Create Backup** to select a path to save the backup to.

   The backup is performed on **Connected** devices. The status of the backup is shown in the **Status** column.



NOTE: Performing a backup interrupts the current job execution.

Restoring Multiple Devices

To perform a restore on connected devices:

1. Select a device on the **View Devices** list to restore a backup.
2. Select the backup file.
3. Click **Restore Device** to initiate a device reboot.

Network Setup

Aurora Focus can connect the device via Ethernet or USB-C-to-A (or C; host-side) cable to the Host PC. When connected, navigate to the **View Devices** screen to view all connected devices and their properties, such as **Name, Model Name, Part Number, IP Address, Serial Number, Firmware Version** and **Status**.

There are three options to connect a device to be discoverable in Zebra Aurora Focus:

1. Connect the device directly to a PC using a USB cable. Using this method does not have access to an internet source by default. In this case, bridge your internet connection using an adaptor.
2. Use an M12-ETH cable to connect the device (M12) directly to the PC's Ethernet port. This option does not provide access to the outside internet by default. In this case, bridge your internet connection using an adaptor. Go to Bridging an Internet Connection for additional information.
3. Use an M12-ETH cable to connect the device (M12) to a switch or router (ETH) on the same subnet as the PC. In this setup, the device typically receives an IP address from the router provided by an internet Service Provider and can reach the outside internet to reach a license server. This is the recommended setup when performing license upgrades and typically does not require bridging a connection.

If access is restricted due to a firewall:

- Contact the IT department to allow network traffic.
- Create a proxy server to allow traffic to the license server.
- Move the PC to a home network and attempt to connect again with a less restrictive firewall configuration.

Configuring Device Settings

Configurable device settings include details on the device, general beeper, power and LED settings, communication settings, and GPIO mapping.

General Settings

Configure the beeper, power, and LED settings using the **General Settings** tab. Beeper settings can be set to turn on or off, and the beep's volume, tone, and duration can be configured to fit the needs of a specific use case.

NOTE: If you are configuring the xS42 with the Zebra Integrated Multifunction Light (ZIML), ensure you emit light from only one source at a time. For example, if you use the aimer on one device, disable it on the other. Beepers, LEDs, illumination sources, and aimers are mutually exclusive and must be configured individually on each device.

NOTE: Hidden Strobe is not available when using the ZIML with the 2MP version of the xS42.

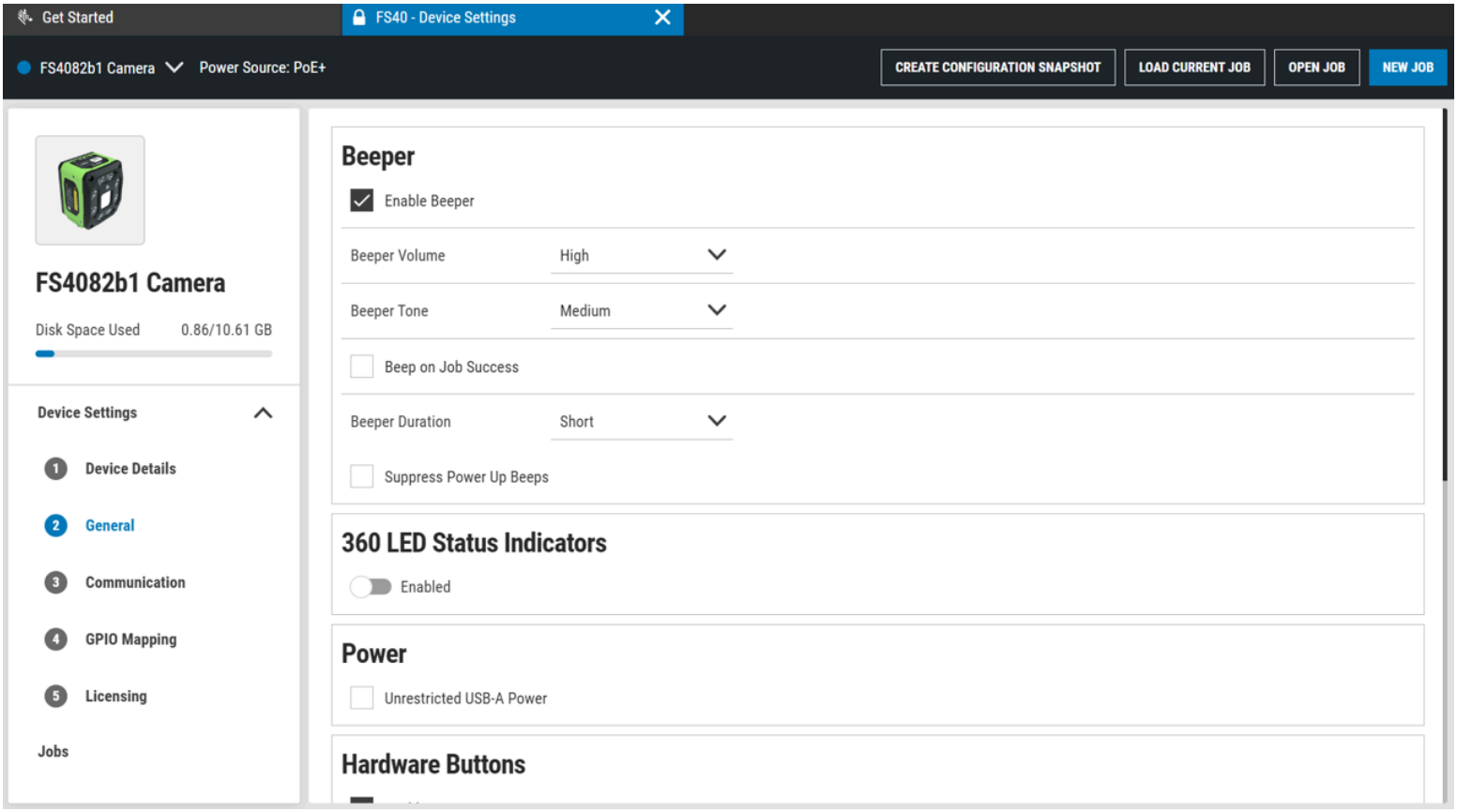


Table 6 General Settings

Setting	Description
Beeper	
Beeper Enable	Enable the device's beeper.
Beeper Volume	Configure the volume of the beeper. • High • Medium • Low • Off
Beeper Tone	Configure the tone of the beeper. • High • Medium • Low • Off
Beep on Job Success	When enabled, the device beeps upon successful job completion.

Table 6 General Settings (Continued)

Setting	Description
Beeper Duration	Configure the tone of the beeper. • Short • Medium • Long
Suppress Power-Up Beeps	When enabled, the warble-beep sequence that plays during device start-up plays at a reduced volume.
Power	
Unrestricted USB-A Power	When enabled, the device receives unrestricted power from a USB-A power source. This is required to enable full-power internal lighting.
Hardware Buttons	
Tune Button Enable	Enable or disable the Tune button on the device.
Trigger Button Enable	Enable or disable the Trigger button on the device.
Configuration Barcodes	
Enable Configuration Barcodes	When enabled, the device configuration can be changed using a barcode.
Timeout	Configure the maximum time (ms) to attempt to generate the barcode before a timeout occurs.
Zebra Integrated Multifunction Light (ZIML) Settings	
Enable ZIML	Slide the toggle to enable the light and select the light model (Full-Featured or Standard Model) from drop-down. **NOTE:** Configurable settings depend on the model.
Enable Laser Aimer	Click the checkbox to enable the laser aimer.
Beeper	Click the checkbox to enable the 360 LED status indicators. • Beeper Volume (Default: High) • Beeper Tone (Default: Medium) • Beeper Duration (Default: Long)
ZIML 360 LED Status Indicators	Click the checkbox to enable the 360 LED status indicators. Click the checkbox to hold the indicators until the next trigger. Configure the number of flashes (Default: 1). Configure the amount of time (ms) the indicator illuminates per flash (Default: 50 ms).
ZIML Power Mode	Select Low Power or High Power.
360° LED	
Number of Flashes	Use the slider to configure the number of flashes the LED illuminates upon decode.

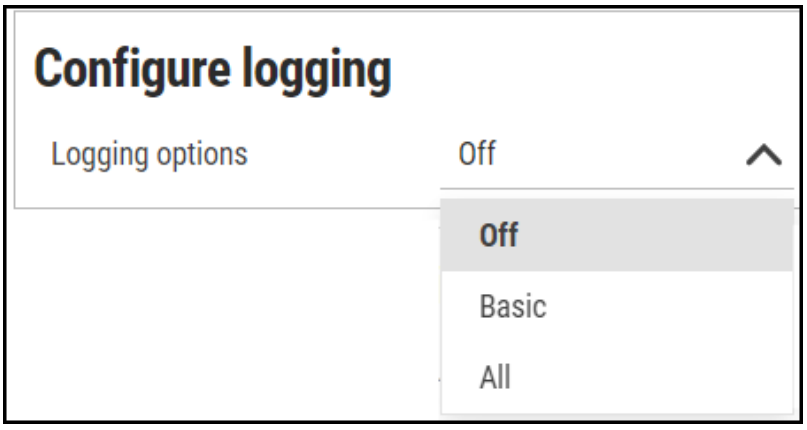
Table 6 General Settings (Continued)

Setting	Description
Time per Flash	Use the slider to configure the amount of time (ms) the LED illuminates per flash (Default: 50 ms) per decode.
Configure Logging	
Logging Options	Select **Basic** or **All** to enable logging and generate files that provide additional information on job runs. Use the Web HMI to download the log files from the device.
Continuous Acquisition Image Saving	
Continuous Acquisition Image Saving Debug Mode	When enabled, debug mode for Continuous Level and Presentation Trigger modes saves all images acquired during Continuous Trigger mode up to the maximum image saved threshold (the default is 100). The saved images provide insight into the frames acquired during a trigger session that did not result in a Pass. This mode differs from the standard No Read Image save functionality, which does not save No Read images during continuous image acquisition.
Maximum Images Saved	Define the maximum number of images saved during a continuous acquisition session. New sessions are started when switching jobs; jobs start after a device reboot and changes from Edit mode to Deploy mode.
New Sessions Clear Old Debug Images	Delete all previously saved images when a new session starts. This option is helpful to ensure all debug images are examined in the same session.

Viewing and Downloading Logs

Enable logs that can be viewed using Perfetto Trace Viewer ui.perfetto.dev/. Logs are saved every two minutes on the device. Up to 10 log files are stored on the device (older logs are replaced by the newest).

To enable Perfetto Logs using Aurora Focus, set the **Logging Options** in the **Configure Logging** section of **General Settings** to **Basic** or **All.**

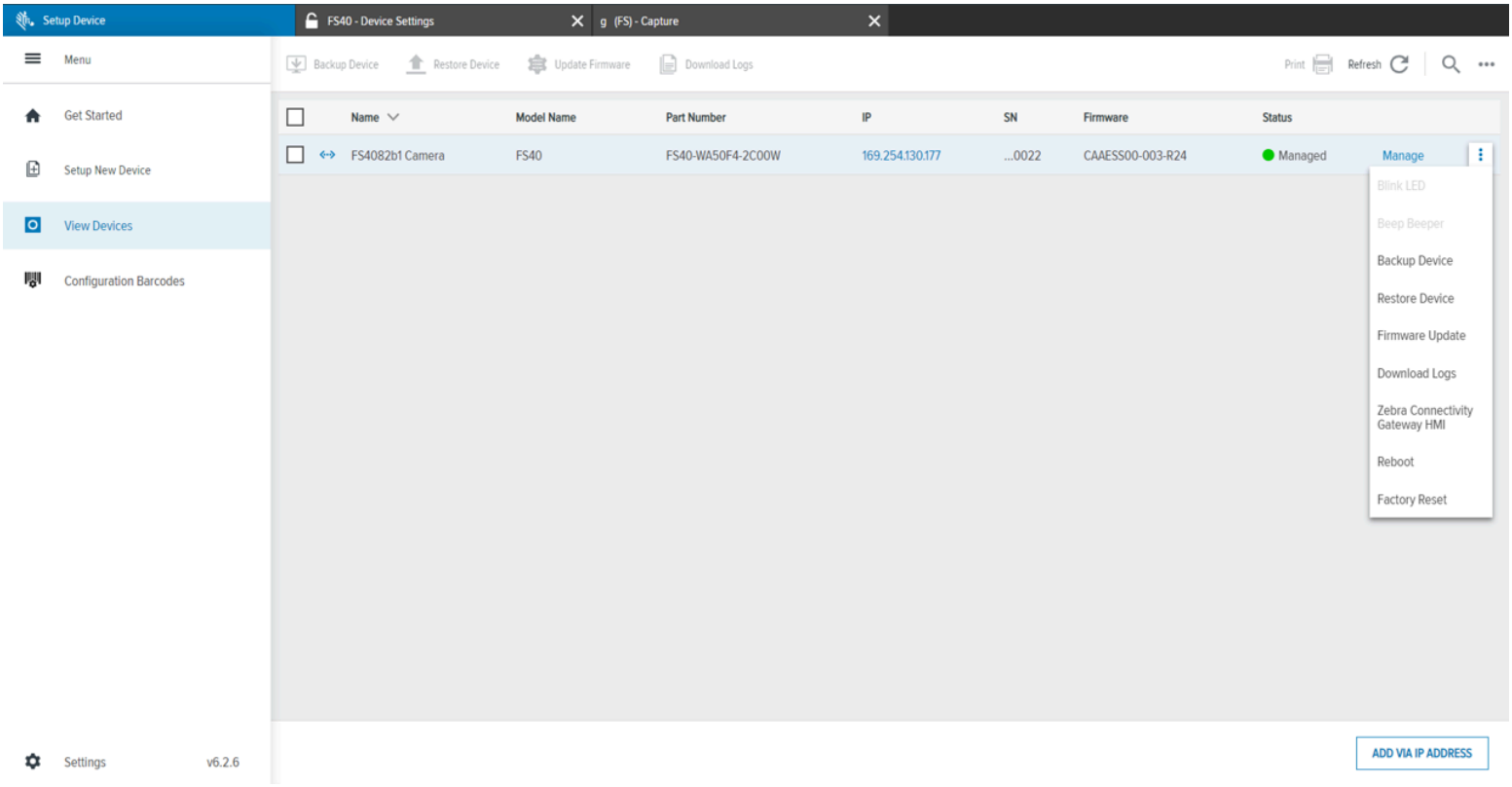


- Basic - provides standard device logging.
- All - provides logging for Zebra Support logs.

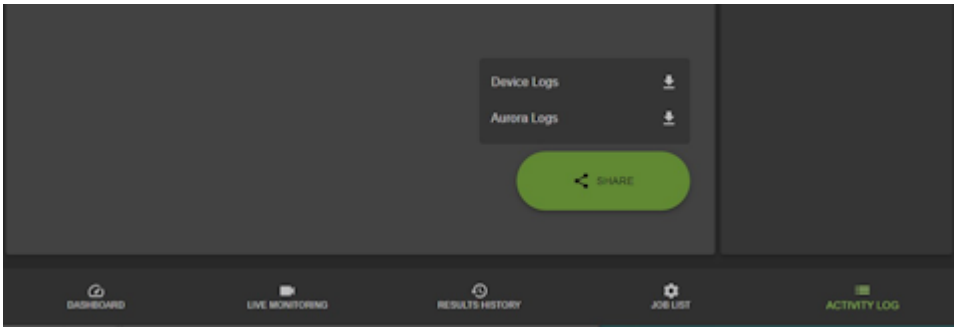
Downloading Basic Logs

Download basic log information using Aurora Focus or the Zebra Web HMI when basic logs are enabled.

Using Aurora Focus, navigate to **View Devices** select a device and click **Download Logs** or click the ellipses to download logs from a subset of specific devices.



Using the Zebra Web HMI, navigate to the **Activity Logs** tab and click Share to download Device or Aurora Logs.



NOTE: Downloaded Log files must be unzipped.

Downloading All Logs

Download logs using the Zebra Web HMI when All logs are enabled.

Download the logs in the **Activity Logs** section using the **Device Logs** option.

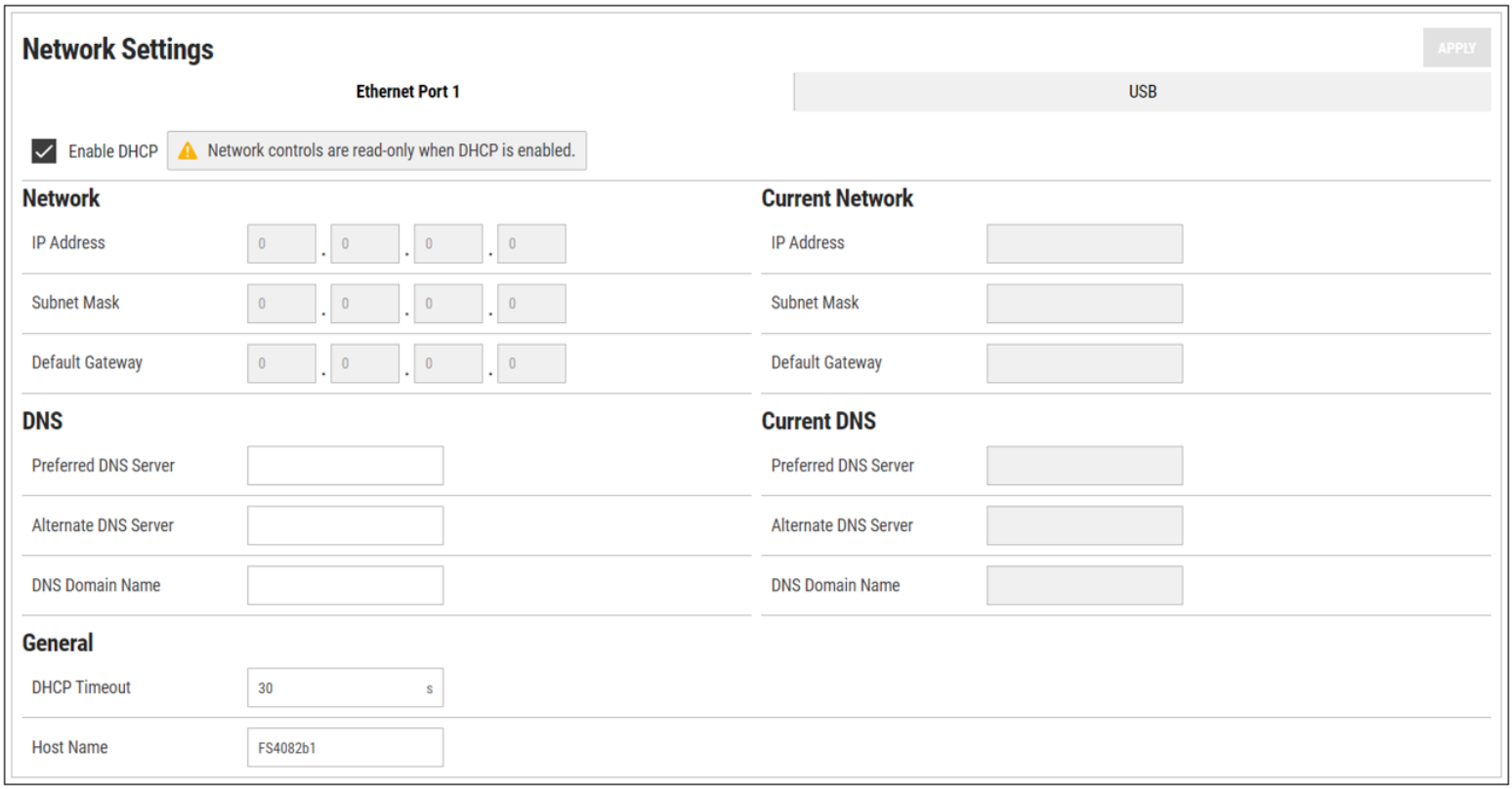
Viewing Perfetto Logs

Drag and drop the log file onto the Perfetto UI or use the **Open trace file** option from the left menu to launch File Explorer and locate the log file to import.

Communication Settings

Configurable communication settings include network, DNS, date/time, PLC protocol, and USB settings. Refer to the FS/VS Industrial Ethernet User Guide for additional information on PLC protocol and Industrial Ethernet.

Click **Enable Edit Mode** to edit the configuration on the device.



Network Settings

Configurable Network Settings include Network and DNS and General settings such as DHCP timeout and Host Name. Network settings vary for each Network interface: Ethernet Port 1, Ethernet Port 2, and USB.

Table 7 Network Settings

Setting	Description
Enable DHCP	Enable DHCP to allow self-configuration of the device in an industrial network.
IP Address	Define the network IP address.

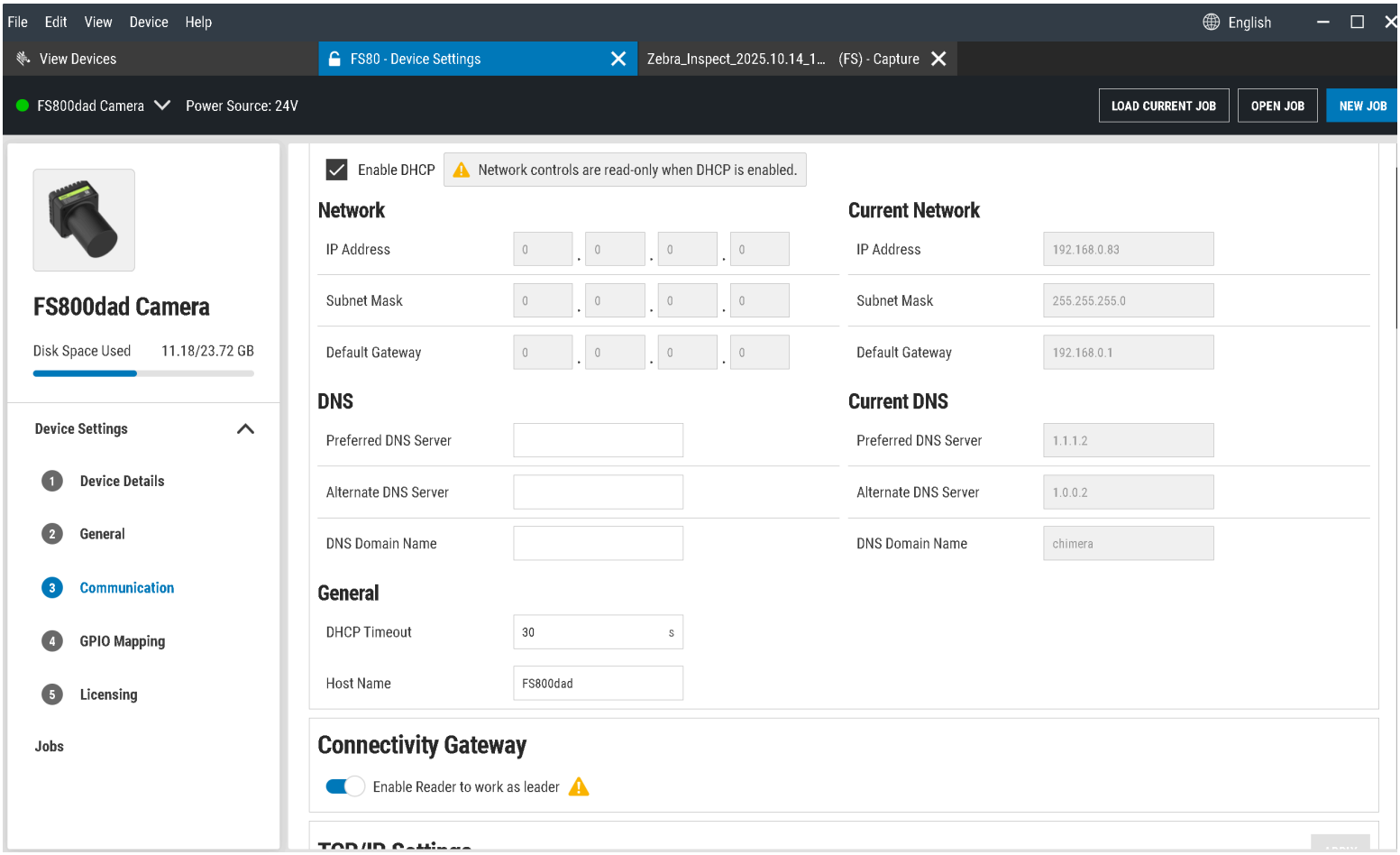
Table 7 Network Settings (Continued)

Setting	Description
Subnet Mask	Define the network subnet mask.
Default Gateway	Set the default gateway to pass information to the device.
Preferred DNS Server	Set the preferred DNS server to connect the device to.
Alternate DNS Server	Set an alternate DNS server in case the preferred server is not available.
DNS Domain Name	Define a DNS domain name.
DHCP Timeout	Set a timeout for the time the device can use the DHCP server.
Host Name	Define the hostname.

Enabling a Device as the Leader Module

Configure a device as the leader module in a connectivity gateway.

1. Go to **Device Details**
2. Select **Communication** from the menu.



3. Click **Enable Edit Mode**
4. Slide the toggle to enable the reader to work as a leader.

NOTE: This change requires a reboot.

TCP/IP Settings

Configure TCP/IP settings, such as enabling TCP.IP Control, selecting a port number, selecting a terminator, or setting a trigger string.

Table 8 TCP/IP Settings

Setting	Description
TCP/IP Settings	
Enable TCP/IP Control	Enabling the TCP/IP control interface allows the device to receive triggers over a TCP/IP connection. **NOTE:** Disabling this interface when it is not in use is recommended.

Table 8 TCP/IP Settings (Continued)

Setting	Description
Same Port for Control and Result	TCP/IP input and output communications are on the same port number when enabled.
Connection Type	Server (Default): Host systems attach the device to the Control Port Number, and the data is sent out with the Control Terminator. Client - The device connects to a host system at the specified IP address, Port Number and use Control Terminator to send the data packet.
IP Address	Enter the host IP address when the Connection Type is set to Client.
Control Port Number	Select a port number that accepts ASCII command strings for the device.
Control Terminator	Select a terminator that follows the incoming ASCII command string.
Trigger String Type	Choose between a single trigger string to toggle the deployed job, or seperate strings to assert or deassert the deployed job. • Toggle • Explicit
Trigger String	Set a customizable trigger command. Sending this string to the Control Port with the Control Terminator triggers the deployed job on the camera when the TCP/IP trigger is selected in the Capture tab.
Results	
Enable TCP/IP Results	Enabling the TCP/IP Results interface allows the device to output result data over a TCP/IP connection. **NOTE:** Disabling this interface when it is not in use is recommended.
Connection Type	The Server Host system attaches to the device using the Result Port Number. The data is sent out with the Results Terminator. The device connects to a host system when the Client is enabled at the specified IP address. Port Number uses Result Terminator to send the data packet.
IP Address	If the Connection Type is set to Client, enter the IP address.
Results Port Number	When in Server mode, the port that the device sends the data out on. When in Client mode, the port on the device's host system connects to.
Results Terminator	For both Client and Server modes, the Results Terminator is the termination text of the string data package. The default setting is CR+LF.
Timeout	When in Client mode, the time the device attempts to connect to the host system.
Deploy Mode Heartbeat	

Table 8 TCP/IP Settings (Continued)

Setting	Description
Enable TCP/IP Heartbeat	When enabled, this setting prompts the device to send out a heartbeat message at the defined interval to the connected TCP result host. The host uses this heartbeat message to verify that the connection with the device is active. **NOTE:** This setting only sends out a heartbeat message while a job is deployed. The heartbeat stops when the device is in Edit mode and resets after each TCP/IP output.
Heartbeat Interval	Set the heartbeat time interval.
Heartbeat Sequence	Create an arbitrary sequence of text and special characters for the heartbeat message.

Click **Test Connection** to check port availability before attempting to connect.

ZETI Settings

Enable ZETI to enable remote control and configuration of the device using ZETI TCP commands.

NOTE: The communication port for ZETI settings is 23.

RS-232 Settings

Configure various RS-232 settings such as a terminator, trigger string, baud rate, or parity.

NOTE: Enabling RS-232 control enables the device to receive control messages from RS-232. For additional information, go to the Connectivity Guidelines section.

Table 9 RS-232 Settings

Setting	Description
Control	Configure the device control settings over RS232.
Enable RS-232 Control	Enable or disable control messages issued to the device through the RS-232 serial port.
Control Terminator	Identify a terminator.
Trigger String Type	Choose between a single trigger string to toggle the deployed job, or seperate strings to assert or deassert the deployed job. • Toggle • Explicit
Trigger String	Identify a customizable trigger command. Sending this string to the Control Port with the Control Terminator triggers the active job on the device, with TCP/IP Trigger selected in the Capture tab.
Results	Enable or disable the transmission of the result message to the serial port.
Speed (Baud Rate)	Configure the speed at which information is transferred to the device.

Table 9 RS-232 Settings (Continued)

Setting	Description
Data Bits	Determine the number of data bits per block of data transmitted.
Parity	Select an error-checking bit that returns a logic bit depending on the number of data bits expected.
Stop Bits	Determine the number of bit periods to wait before sending the next start bit.

Date/Time Settings

Configure date/time settings by providing an NTP server.

NOTE: There is no internal battery in the device, therefore, date and time settings are not preserved.

Table 10 Date/Time Settings

Setting	Description
NTP Server 1	The IP address of the primary Network Time Protocol server to synchronize to. Default: 0.pool.ntp.org
NTP Server 2	The IP address of the second Network Time Protocol server to synchronize to. Default: 1.pool.ntp.org

PLC Protocol

Determine the appropriate PLC protocol based on your use case.

For additional information, refer to the FS/VS Smart Camera Series Industrial Ethernet User Guide.

NOTE: The FS80 supports EtherNet/IP and Modbus TCP protocols.

Table 11 PLC Protocol

Settings	Description
Industrial Ethernet/PLC Options	Select the desired industrial protocol for device communication. • None • EtherNet/IP • Profinet • Modbus TCP

NOTE: Use the **Connect** tab to configure **Industrial Ethernet** settings.

USB Settings

Enable specific USB settings such as a HID keyboard, keystroke delay, or terminator.

Table 12 USB Settings

Settings	Description
Enable HID Keyboard	Enabling the HID keyboard interface allows the device to output result data in a HID keyboard mode. This restarts USB communication as a Human Interface Device class keyboard to emulate keystrokes. **NOTE:** It is recommended to disable this interface when not in use. The device can be set to either HID keyboard or USB CDC-Serial mode. Both modes cannot be enabled simultaneously.
Keyboard Country Type	Select your country from the list of available Keyboard Country Types. • English (North America) • French (France) Windows • German Windows • Spanish (Spain) Windows • Italian Windows • English (UK) Linux • German Linux • Czech Linux • Spanish (Mexico) Linux • French (France) Linux • Polish Linux • Spanish (Spain) Linux • Italian Linux • Portuguese (Brazil) Linux • Japanese Linux
Keystroke Delay	Set the delay (in ms) between emulated keystrokes when HID is enabled.
Special Key Keystroke Delay	Set the delay (in ms) between emulated special keystrokes when HID is enabled.
Control	
Enable USB CDC-Serial Control	Activate the management of special commands received by the CDC-Serial port.
Control Terminator	Set the message terminator that follows the incoming ASCII command string.
Trigger String	Command string used to trigger new image acquisition.
Results	

Table 12 USB Settings (Continued)

Settings	Description
Enable USB CDC-Serial Results	Enable the transmission of the results over the CDC-Serial port.

GPIO Mapping

Configure the GPIO on the 12-pin and 5-pin connectors by selecting the GPIO and configuring its direction and signal type. Enable External Illumination mode for the 5-pin connector by specifying its mode.

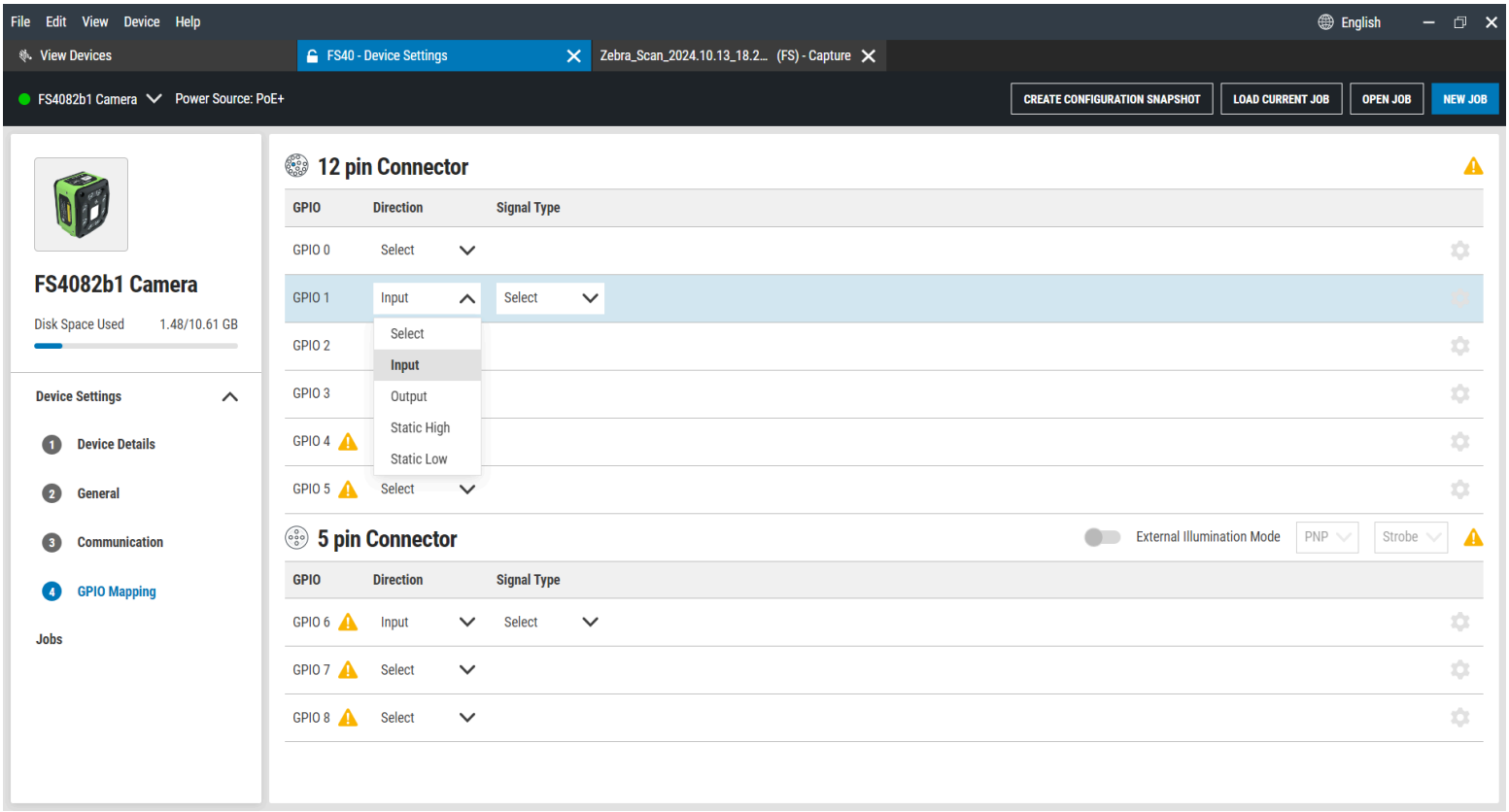


Table 13 GPIO Mapping

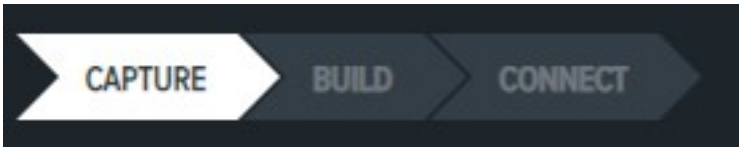
Setting	Description
Direction	Configure the input signals coming into the device, or the output signals sent out. Select the GPIO port configuration: Input, Output, Static High, and Static Low. Static High sources the current for an external load. Static Low sinks the current for an external load.

Table 13 GPIO Mapping (Continued)

Setting	Description
Signal Type	Configure the Signal Type: Normal or Hardware Trigger. Normal Trigger - signal is routed through the device software. Normal Trigger is useful for Continuous, Level, periodic, Presentation, and Burst modes. Hardware Trigger - signal is routed through the device hardware directly to the sensor, bypassing the processing of the trigger. Hardware Trigger is more responsive and useful for Single Trigger use cases.
Advanced Settings	Use Input to configure Input Debounce and Input Delay. Use Output to configure Pulse Width and Output Delay.
External Illumination Mode	Toggle this setting to automatically configure the pins in this port to power and control external illumination.

Configuring Jobs

Configuration and deployment of FIS or MV jobs are facilitated using the Capture, Build, and Connect tabs.

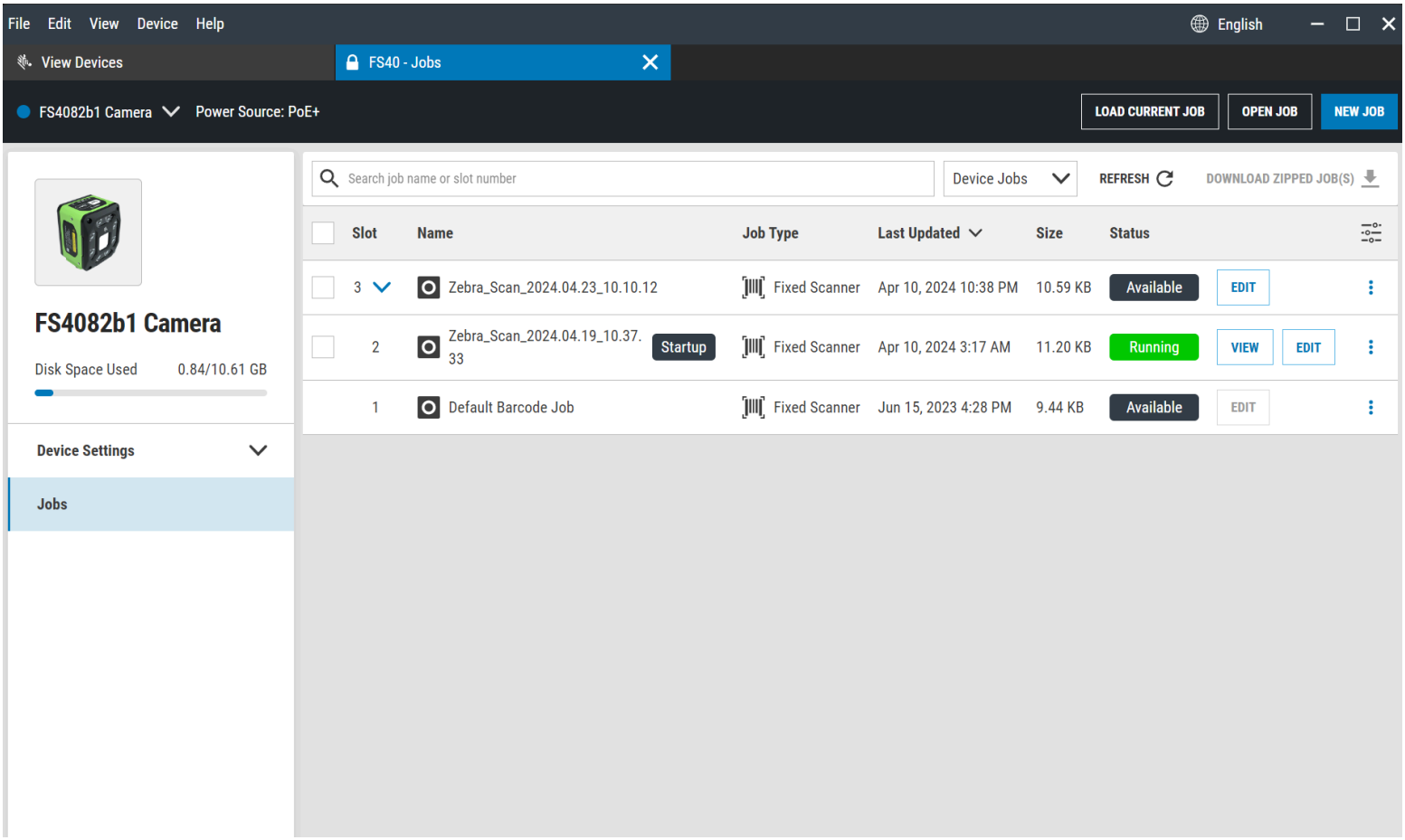


- **Capture** - configure the Job Mode and Source settings on the Triggers tab and imager settings on the Acquisition Settings tab.
- **Build** - depending on the toolset (FIS or MV), configure the settings and symbologies or tools used to complete a specific job.
- **Connect** - save and configure image settings and network connections such as Industrial Ethernet, Output Formatting, and GPIO Mapping.

Job Slot Management

Use the Jobs tab in Device Settings to manage job slots, rename, or delete jobs.

1. Go to **Device Details**
2. Select **Jobs** from the menu.

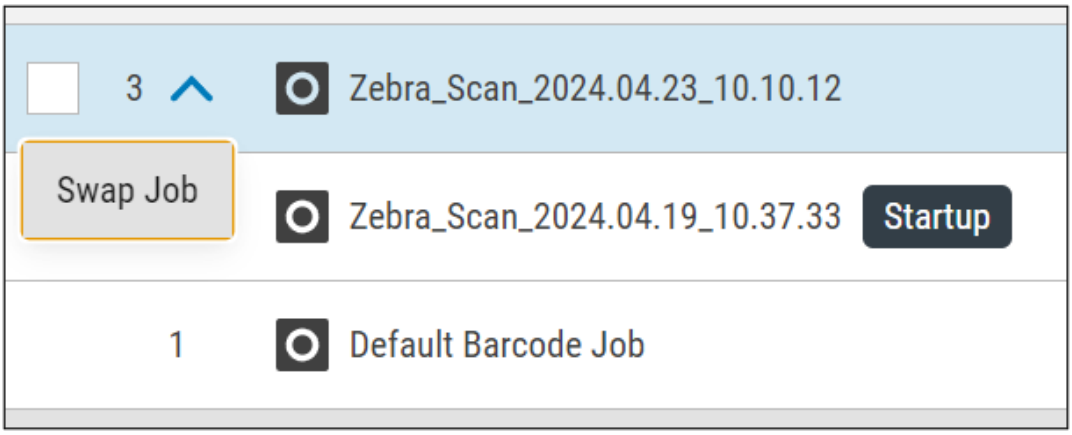


a) Startup - indicates the job that runs upon the camera's startup.

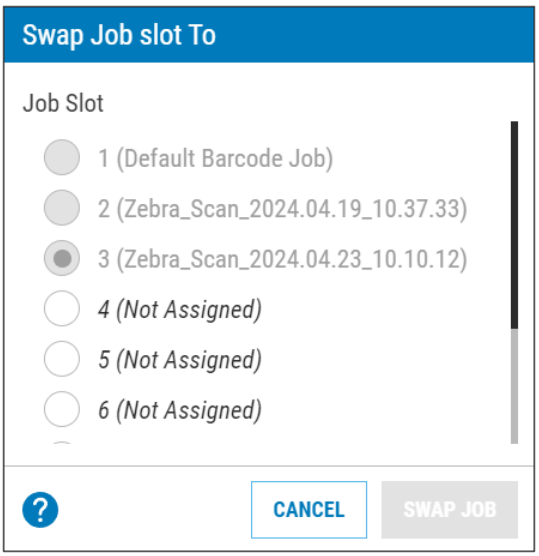
b) Active - indicates a job that is in edit mode.

c) Available - indicates a job that is deployed to the device but not running.

3. Click filter by slot, name, job type, last updated, size, or status.
4. Click **Edit** to edit the job in build mode.
5. Click ⋮ to set the job as running, clear the running jobs, set the selected job as the Startup job, clear the Startup job, delete, duplicate, or rename a job.
6. Click ⌄ and click **Swap Job** to swap the current job slot with another available slot.



7. Select the job slot to swap the current job with.



8. Click **Swap Job** to confirm the change.

Using Commands to Manage Job Slots

Use Telnet to communicate with the device and manage job slots by executing commands.

NOTE: Ensure that TCP/IP control is enabled using the Communications tab in Device Settings to use commands to manage job slots.

Load a job to a job slot:

```
job_slot_load <slot number>
```

Load a job to run at device startup:

```
job_slot_load <slot number> --startup
```

Swap a current job slot:

```
job_slot_swap <current slot number> <new slot number>
```

Set a job slot:

`job_slot_set <slot number>`

Get a job slot:

`job_slot_get`

Delete a job slot:

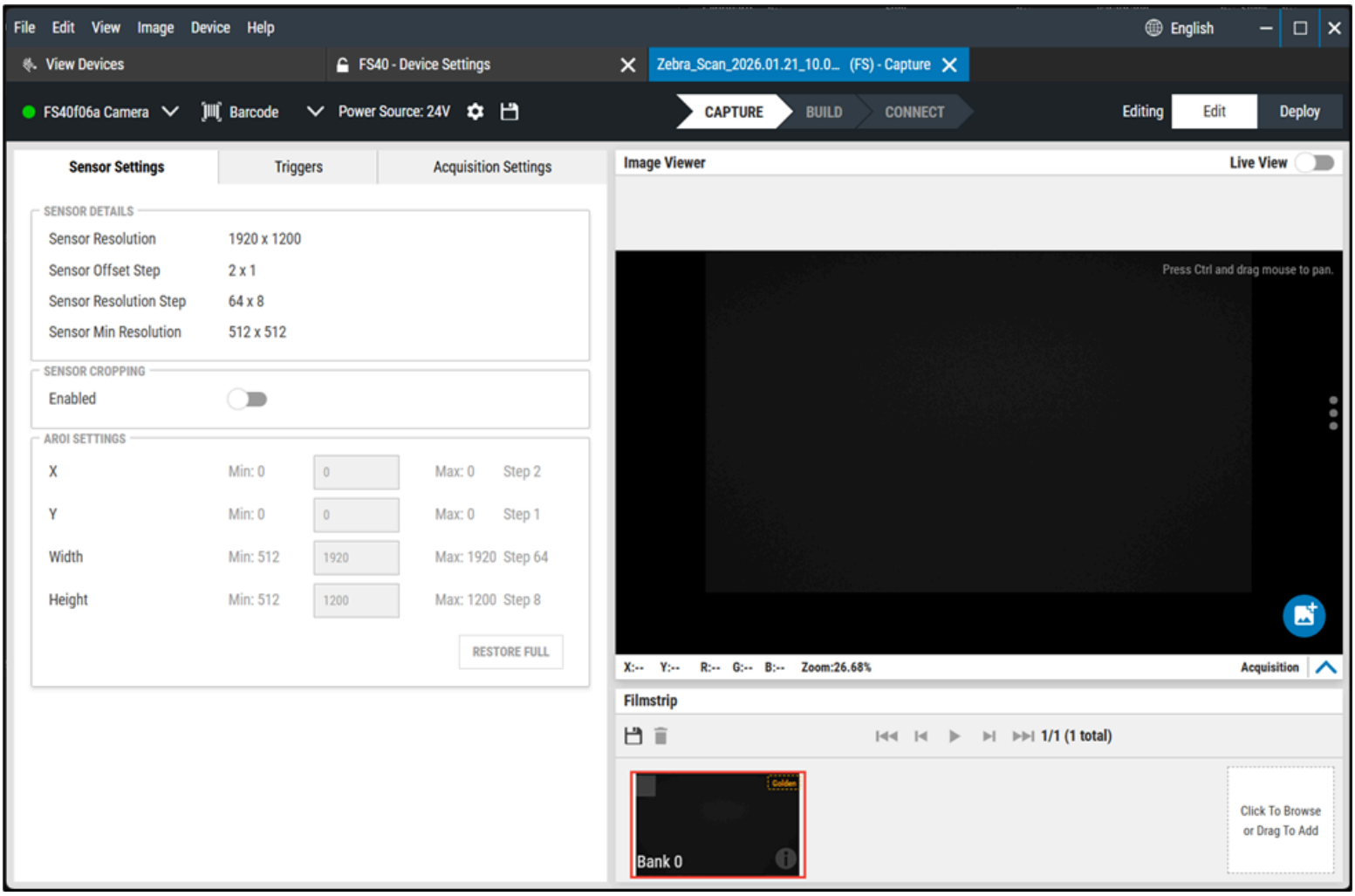
`job_slot_delete <slot number>`

List all current job slots:

`job_slot_list`

Capture

The Capture tab configures **Sensor Settings**, **Triggers** and **Acquisition Settings** for Fixed Industrial Scanning or Machine Vision jobs.



The sensor cropping feature allows for a flexible definition of the AROI leading to a reduction in acquisition time.

Define the AROI in two ways:

- By entering parameters in the **AROI Settings** section.
- By drawing a region directly in the **Image Viewer** window.

In the top-left corner of the Image Viewer, you will find the current values for **Acquisition time**, defining the maximum possible value for the **Exposure** and **Frame rate**.

When editing, remember:

- The AROI must be defined before switching to other tabs.
- After adding a tool in the **Build** tab, the AROI cannot be changed.
- The Long exposure option becomes unavailable.

Table 14 Sensor Settings

Item	Description
Sensor Details	This section contains important sensor parameters, including: • Sensor resolution • Offset step • Resolution step • Minimal resolution
Sensor Cropping	Enable to configure AROI settings. When enabled, the Image Viewer displays the Acquisition time and Frame rate.
AROI Settings	X – starting point on the x-axis with minimum, maximum value, and step Y – starting point on the y-axis with minimum, maximum value, and step Width – width resolution with minimum, maximum value and step Height – height resolution with minimum, maximum value, and step

Table 15 Trigger Settings

Item	Description
Source	Select the trigger source from GPIO, Device Trigger, Serial, PLC, TCP/IP Auto(Self), or Test Trigger.
Job Mode	Select the Job Mode between Single Shot, Level Continuous, Series (Burst), Presentation (Aggregate Mode), Periodic Single Shot, Continuous, Presentation

Table 15 Trigger Settings (Continued)

Item	Description
Show Image	Enable to configure Show Image settings. When enabled, the device sends the images back to Aurora Focus, which takes cycle time. To speed up job times, configure these settings to reduce or disable the frequency of image transmission.
Show Image Mode	Determine the method to updating shown images.
Update Every	Determine how frequently shown images are updated per inspection.
Triggered Image Buffering	Buffer images for triggers received while a job is in progress.
Maximum Image Queue Size	Determine the maximum number of images that can be queued before they are dropped from the queue.
Drop Mode	Determine the approach to drop images when the queue is full.

Acquisition Settings determine the configuration used to capture images with the device.

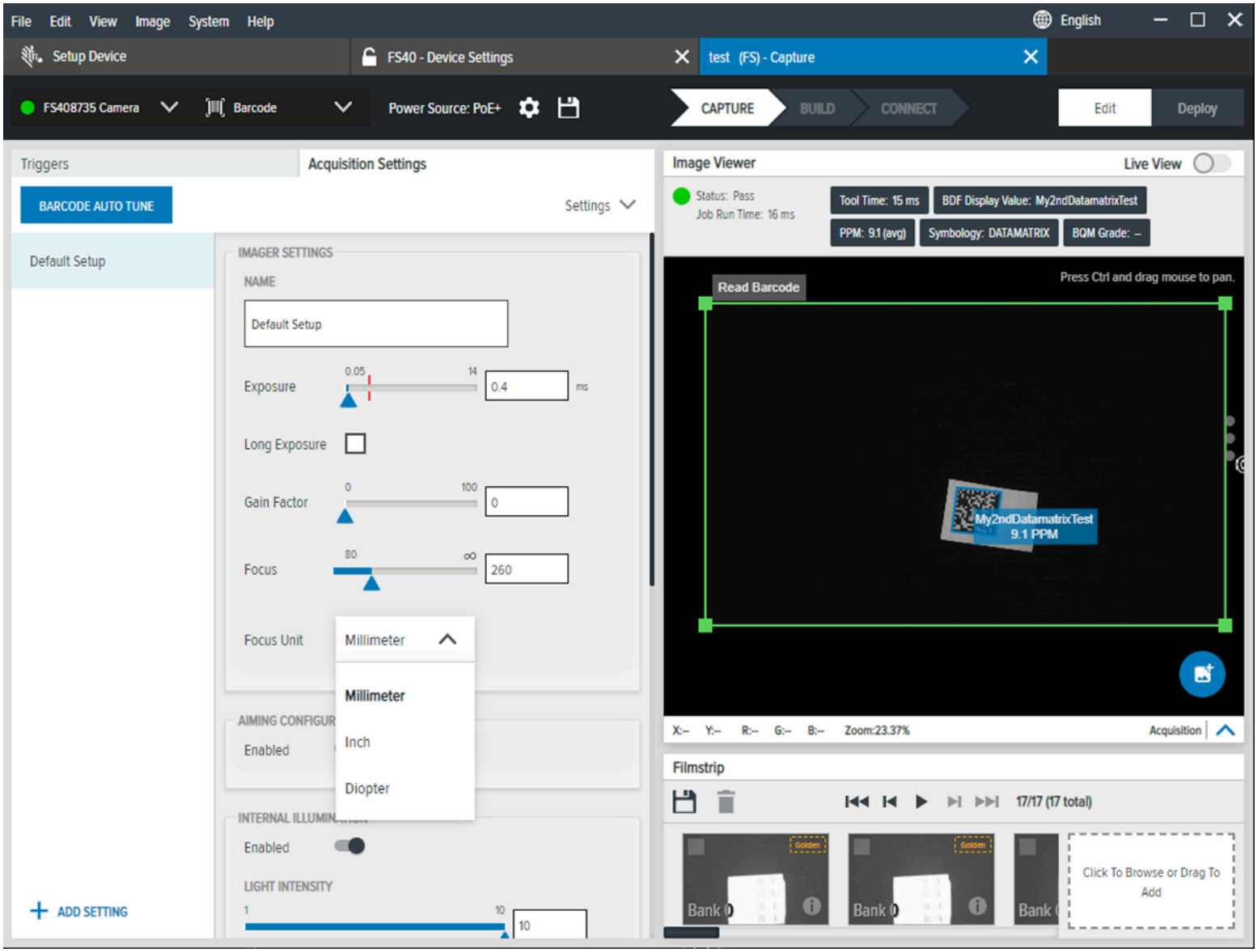


Table 16 Acquisition

Setting	Description
Barcode Autotune	Force an automatic tune procedure, configuring the camera to read the barcode or Datamatrix and adjust. **Exposure**, **Gain Factor** and **Focus**.
Autotune Settings	Open the dialog to adjust the Autotune Settings.
Image Perfect+ Setup	The setup list on the left is the possible configuration for ImagePerfect+.
Add Settings	Use this command to add a new setup to Image Perfect+
Name	Assign a name for the Image Acquisition Setup

Table 16 Acquisition (Continued)

Setting	Description
Exposure	Exposure time in ms.
Long Exposure	Activate this option to adjust the exposure time from 33 to 750 ms.
Gain Factor	Use this slider to adjust the sensor gain factor from 0 to 100.
Focus	Use this slider to adjust the focus length.
Focus Unit	Select Millimeter, Inch, or Dioptor to configure the focus measurement units on the device. **NOTE:** Depending on the hardware version of your device, **Millimeter** and **Inch** measurements are available in addition to **Diopter**. While the millimeter and inch distance values calculated in the application may differ slightly from the actual distance value, this does not impact the device's performance.
Aiming Configuration	Enabling this feature projects an aiming pattern onto the surface where the camera is pointing, helping to position the camera by identifying the focal point of the field of view.
Internal Illumination	This setting is only available with Unrestricted USB Power, PoE, and 24V external power. Select the Light Intensity (0-10V), color (red, white, green, IR), and quadrant (top, bottom, left, right) of the 360° LED light.
Light Intensity	Use the slider to increase light intensity by sliding to the right. • Minimum: 1 • Default: 4 • Maximum: 10
Red Light	• Top: enable this setting to use the top illuminator on the device. • Bottom: enable this setting to use the bottom illuminator on the device. • Right: enable this setting to use the suitable illuminator on the device. • Left: enable this setting to use the left illuminator on the device.

Table 16 Acquisition (Continued)

Setting	Description
External Illumination	Internal and External illumination can be activated simultaneously on FS42 devices while connected to a 24V DC power supply. Select the Light Intensity (0-10V).
Light Intensity	This controls the analog voltage pin on the Analog Output Pin. • Minimum: 1 • Default: 8 • Maximum: 10

See Also

Viewing Devices

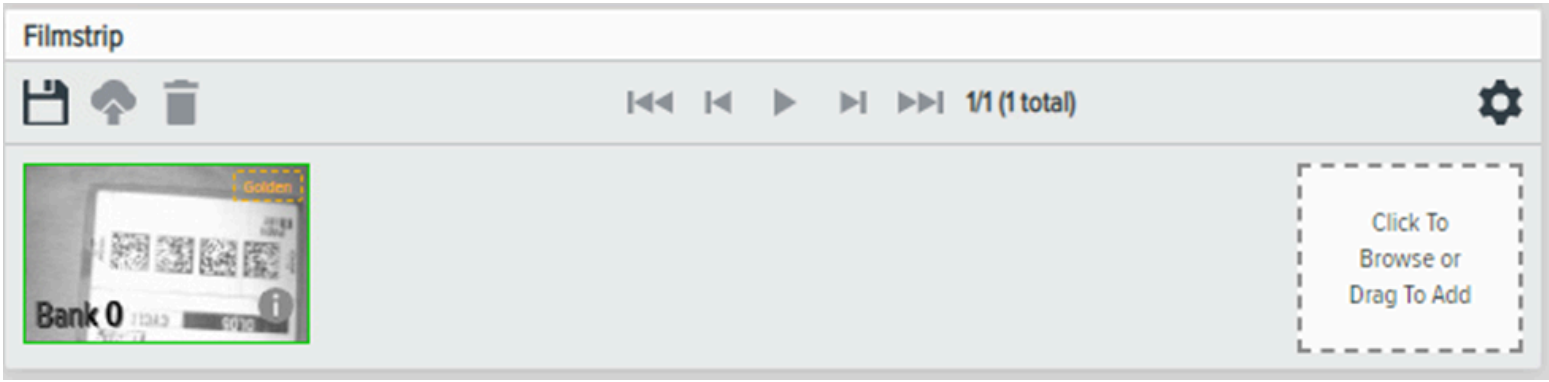
Trigger Settings

Different trigger settings are enabled based on the job mode.

- Single Shot - initiates a single image capture. Single Shot helps inspect an image.
- Level Continuous - initiates an image capture on a level trigger for objects in motion. This is helpful for use cases involving a conveyor belt.
- Series (Burst Mode) - initiates a series of image captures. Bust mode is helpful for use cases that require capturing a series of images based on a user-defined number of trigger intervals and frequencies.
- Periodic Single Shot - initiates a single image capture after a user-defined period of time. It is helpful for debugging.
- Continuous - initiates a series of captures rapidly. Continuous is helpful for barcode scanning applications.
- Presentation Mode - illuminates the scan window when the sensor detects motion. This mode is helpful for kiosk and retail point-of-sale applications.
- Presentation (Aggregate Mode) - triggered the same way as Presentation Mode and aggregates all the barcodes decoded during the active job timeout

Using the Filmstrip

Drag images onto the Filmstrip from your PC to add to the image bank.

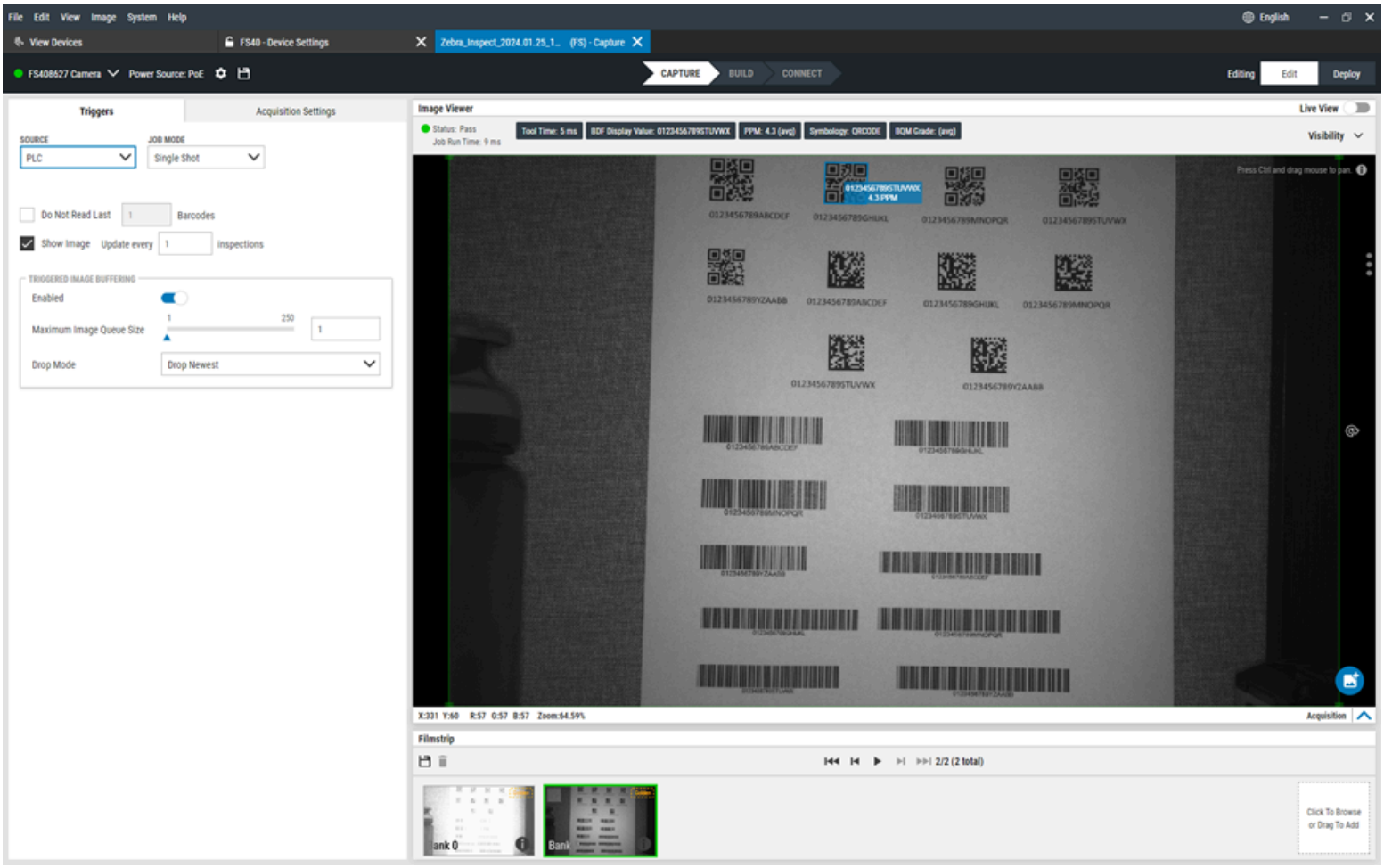


PLC Trigger Modes

The device supports different use cases with single shot, level continuous, series (burst), periodic single shot, continuous, and presentation trigger modes.

Single Shot

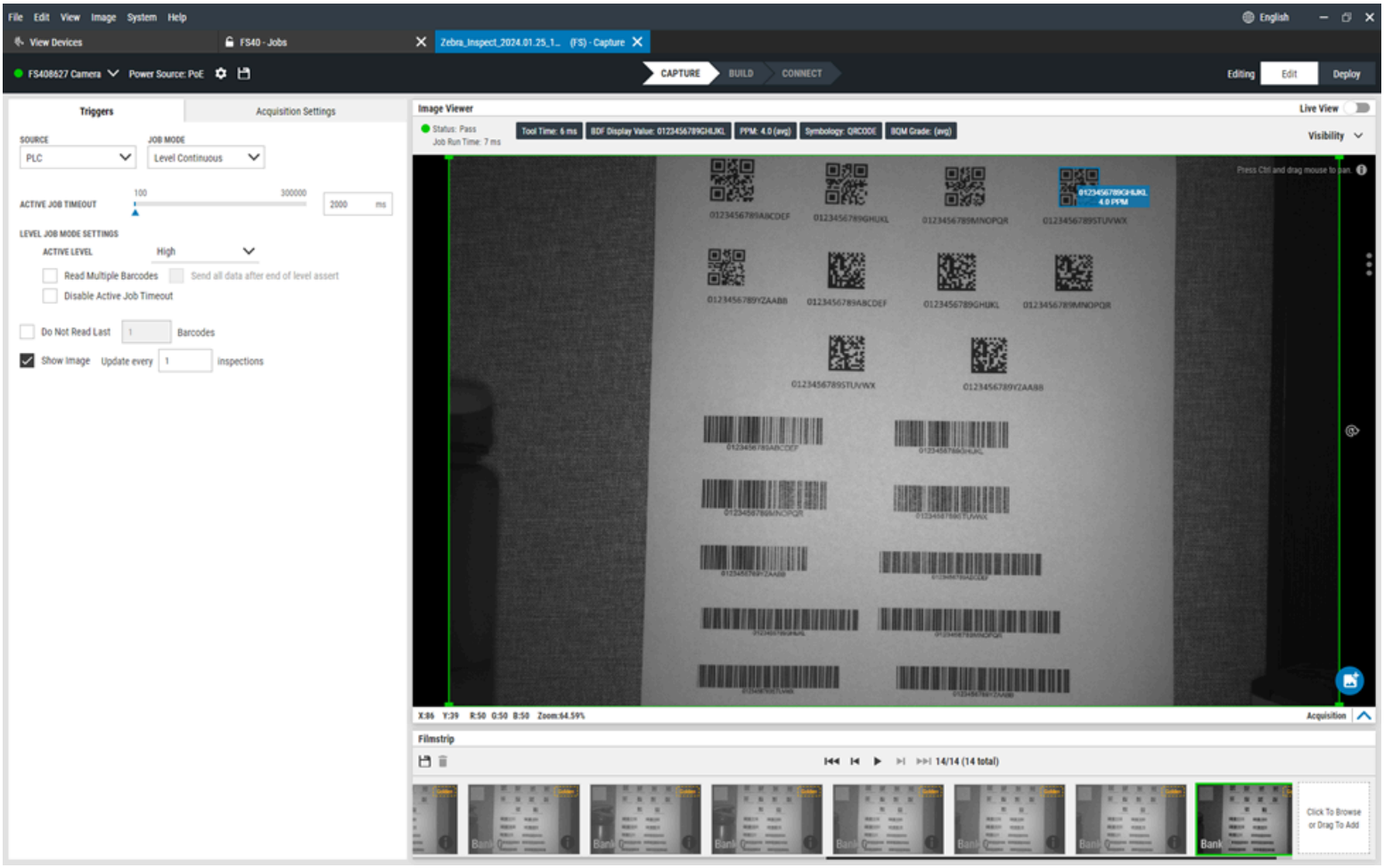
When the job is in Single Shot trigger mode, the device runs one job on each trigger and decodes the barcode.



Toggle the trigger bit from 0 to 1 to perform a trigger. The job stops when the trigger is complete. Toggle the trigger bit to run the job again. Toggling the trigger bit from 0 to 1 executes the job once.

Level Continuous

Level Continuous initiates an image capture on a level trigger for objects in motion. This trigger mode is particularly helpful for use cases involving a conveyor belt. This mode consistently captures the image for the duration specified in the Active Job Timeout setting. When that duration expires, the job stops capturing the image.

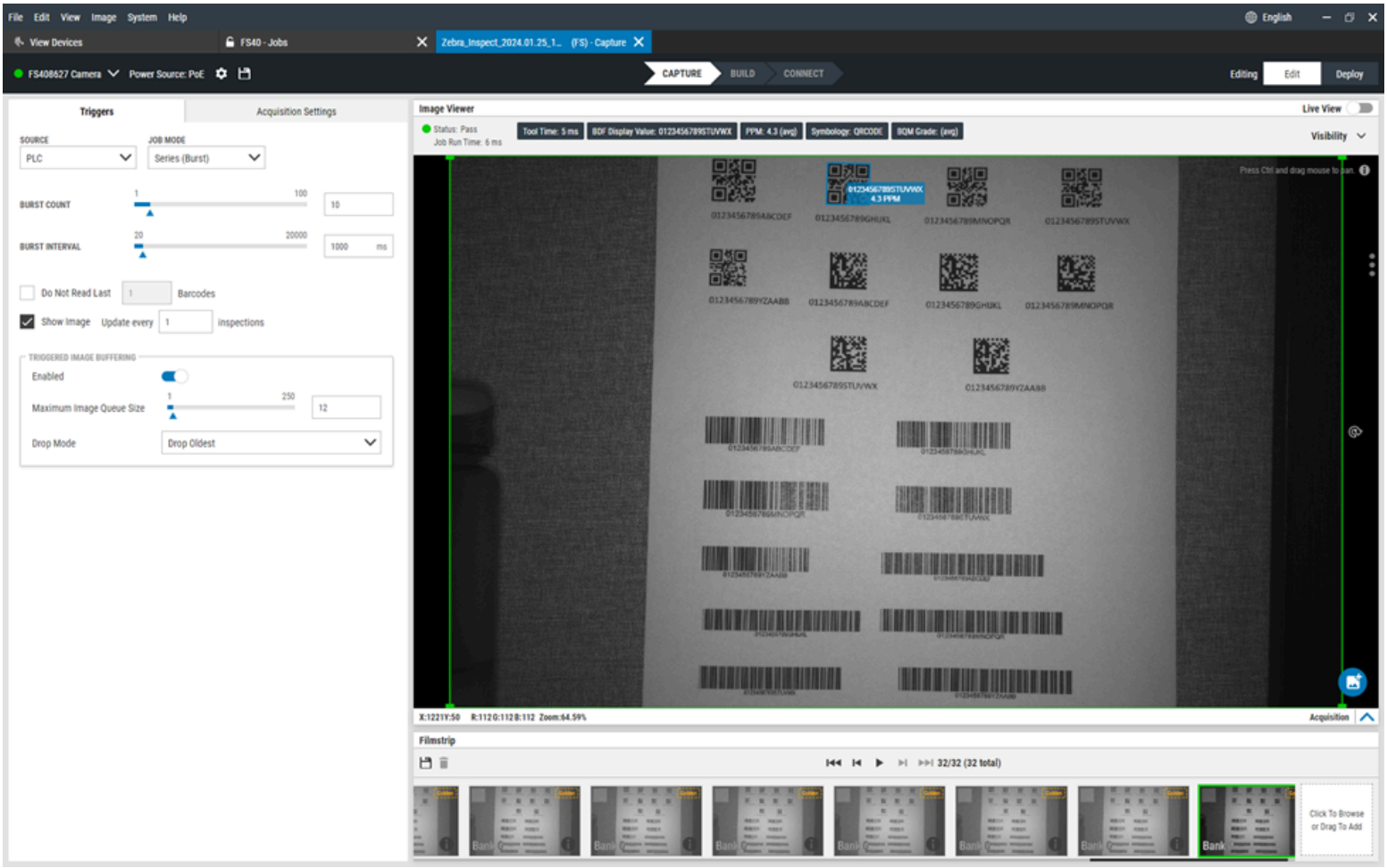


To trigger the job for the first time, the trigger bit state should be toggled from 0 to 1. The job runs for the time specified in Active Job Timeout and stops. Toggle the trigger bit state from 0 to 1 to trigger the job for the first time. Toggle the trigger bit from 0 to 1 before the Active Job Timeout to manually stop the job. Toggle the trigger bit from 0 to 1 again to trigger the job again.

NOTE: Toggling the trigger bit from 1 to 0 does not affect the job.

Series (Burst)

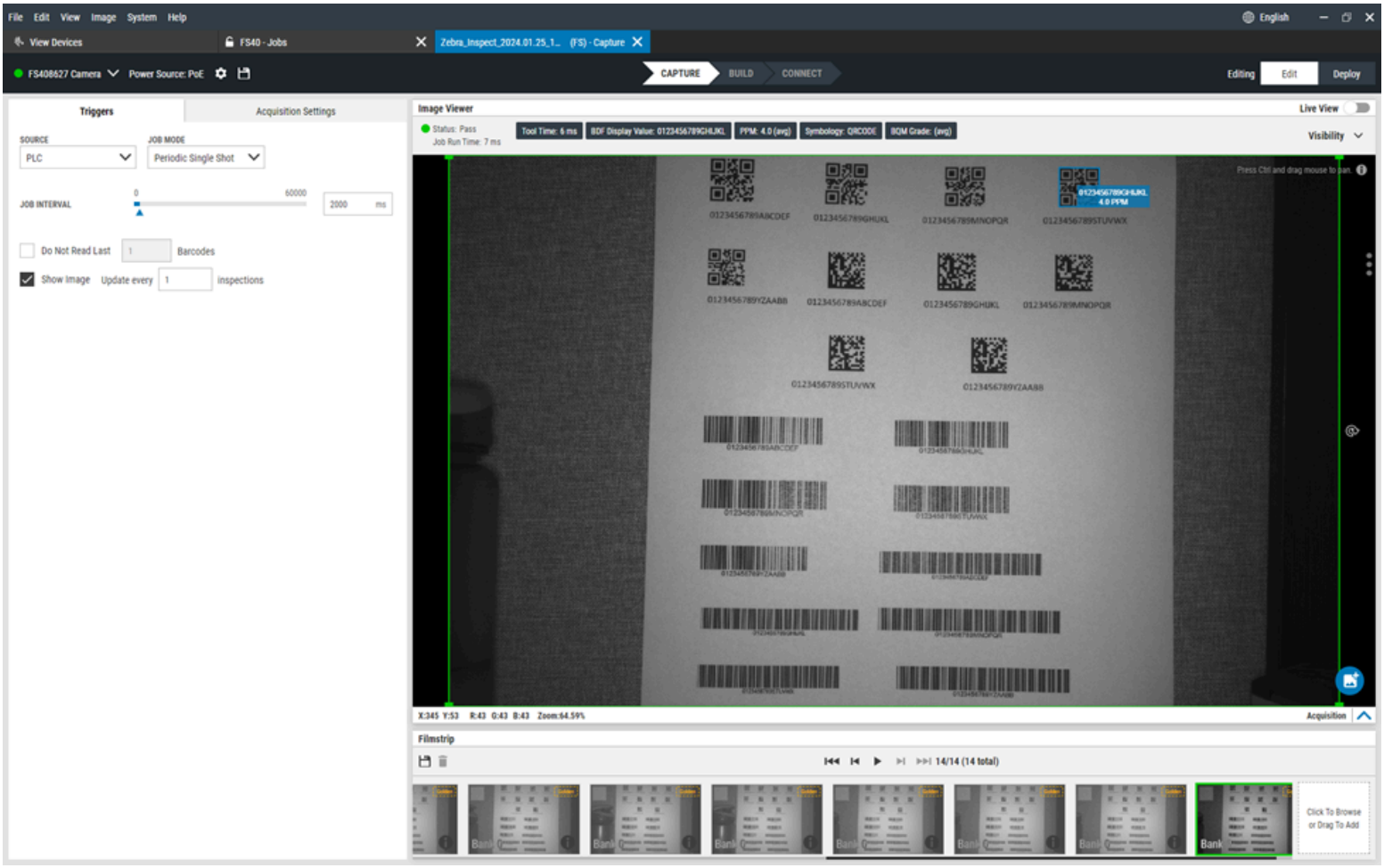
Burst mode initiates a series of image captures, which is helpful for use cases that require capturing a series of images based on a specified amount of trigger intervals and frequency.



In this mode, the first trigger event (toggling the trigger bit from 0 to 1) starts the burst sequence. Consecutive trigger events do not have any effect until the previous burst sequence is complete. The following trigger event (toggling the trigger bit from 0 to 1) restarts the burst sequence.

Periodic Single Shot

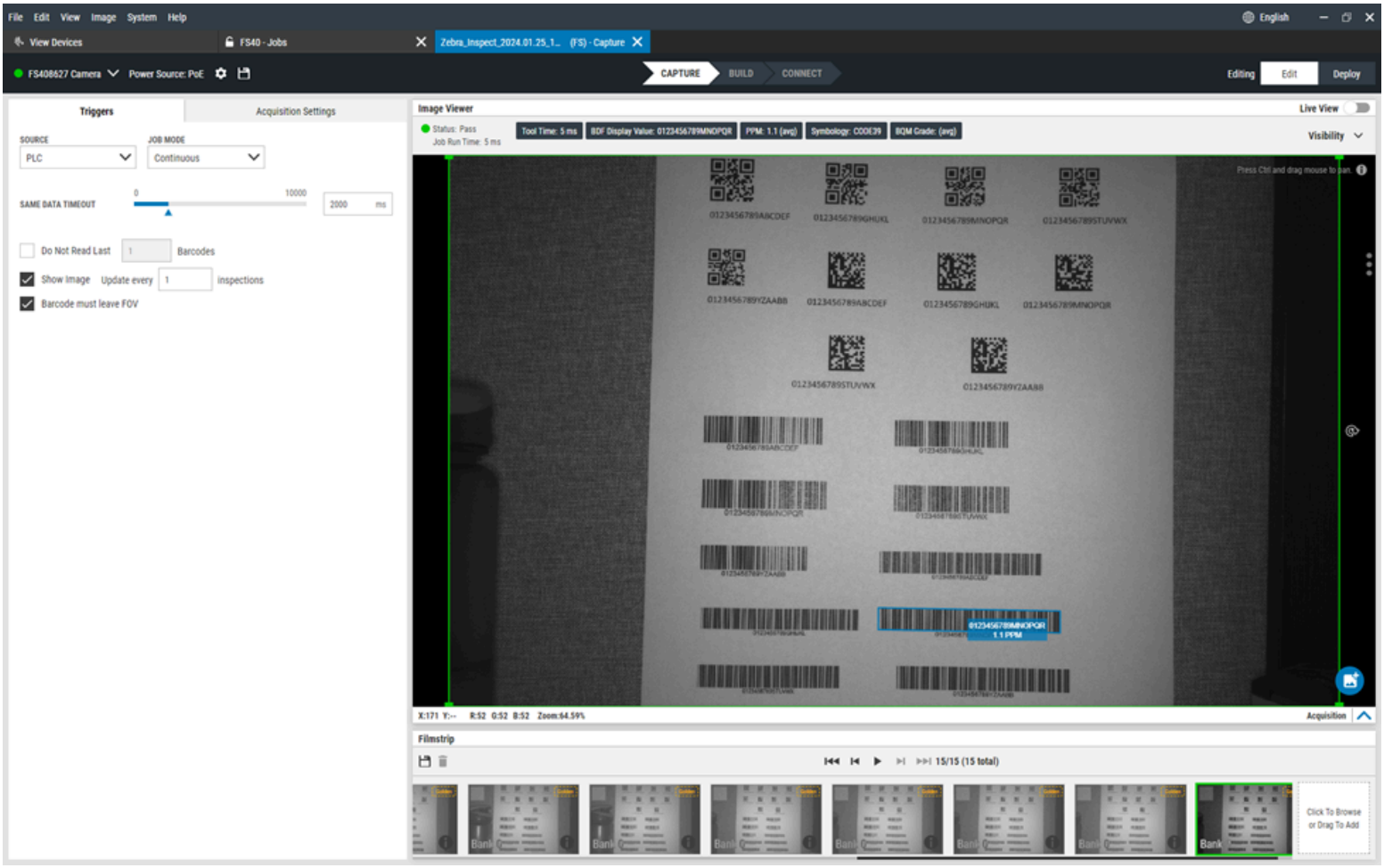
Periodic Single Shot mode initiates a single image capture after a specified time. This is helpful for debugging use cases.



Periodic Single shot maintains the triggering job periodically after a specified period. Send the trigger again to stop the job in progress. While using PLC, the first trigger (toggling the trigger bit from 0 to 1) starts the job, and a second trigger event (toggling the trigger bit from 0 to 1) stops the job.

Continuous

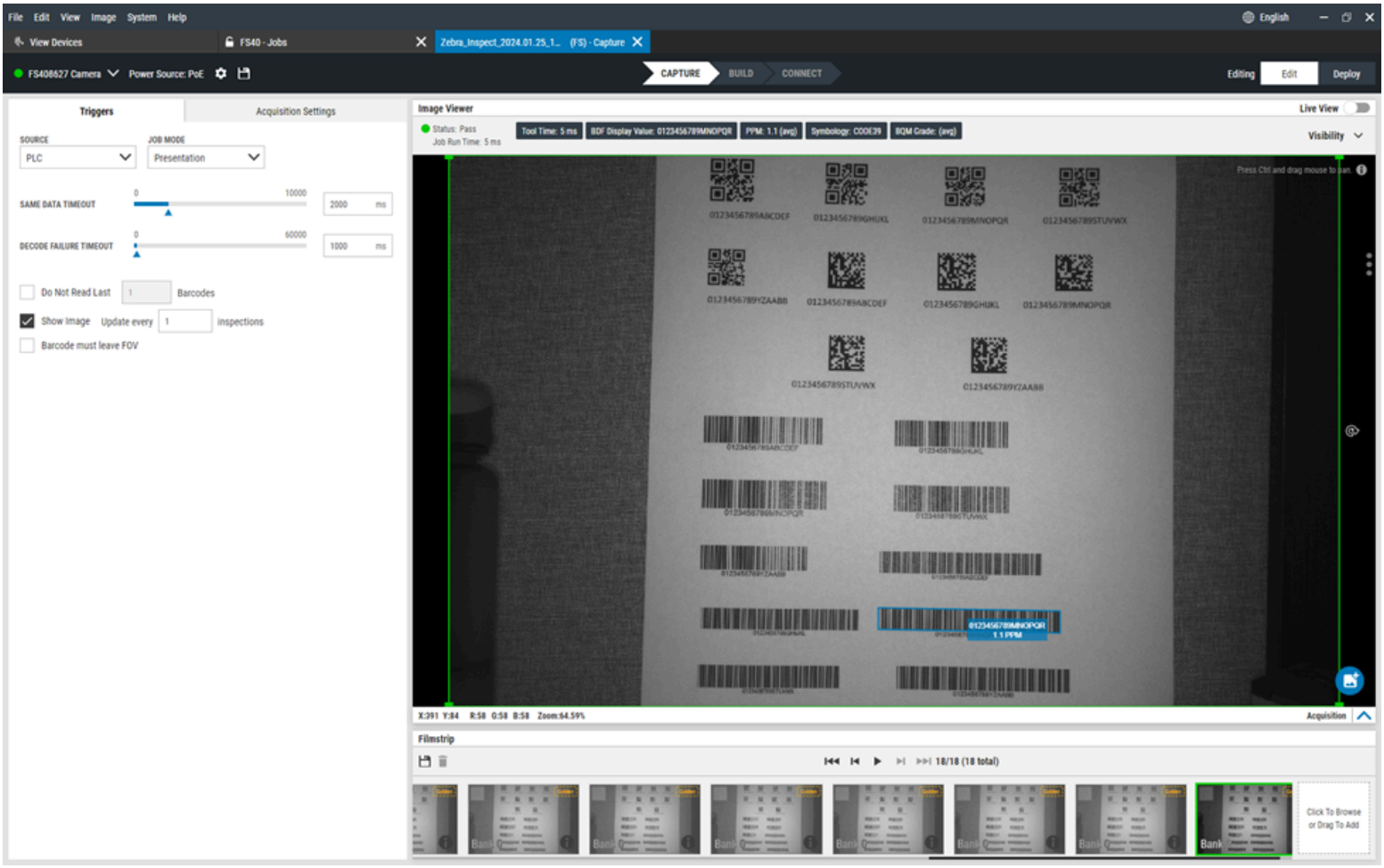
Continuous mode initiates a series of image captures rapidly. This mode is helpful for barcode scanning applications.



While in Continuous Mode, the PLC behavior is the same as in Periodic Single Shot mode.

Presentation

Presentation mode illuminates the scan window when the sensor detects motion. This mode is helpful for kiosk and retail point-of-sale applications.

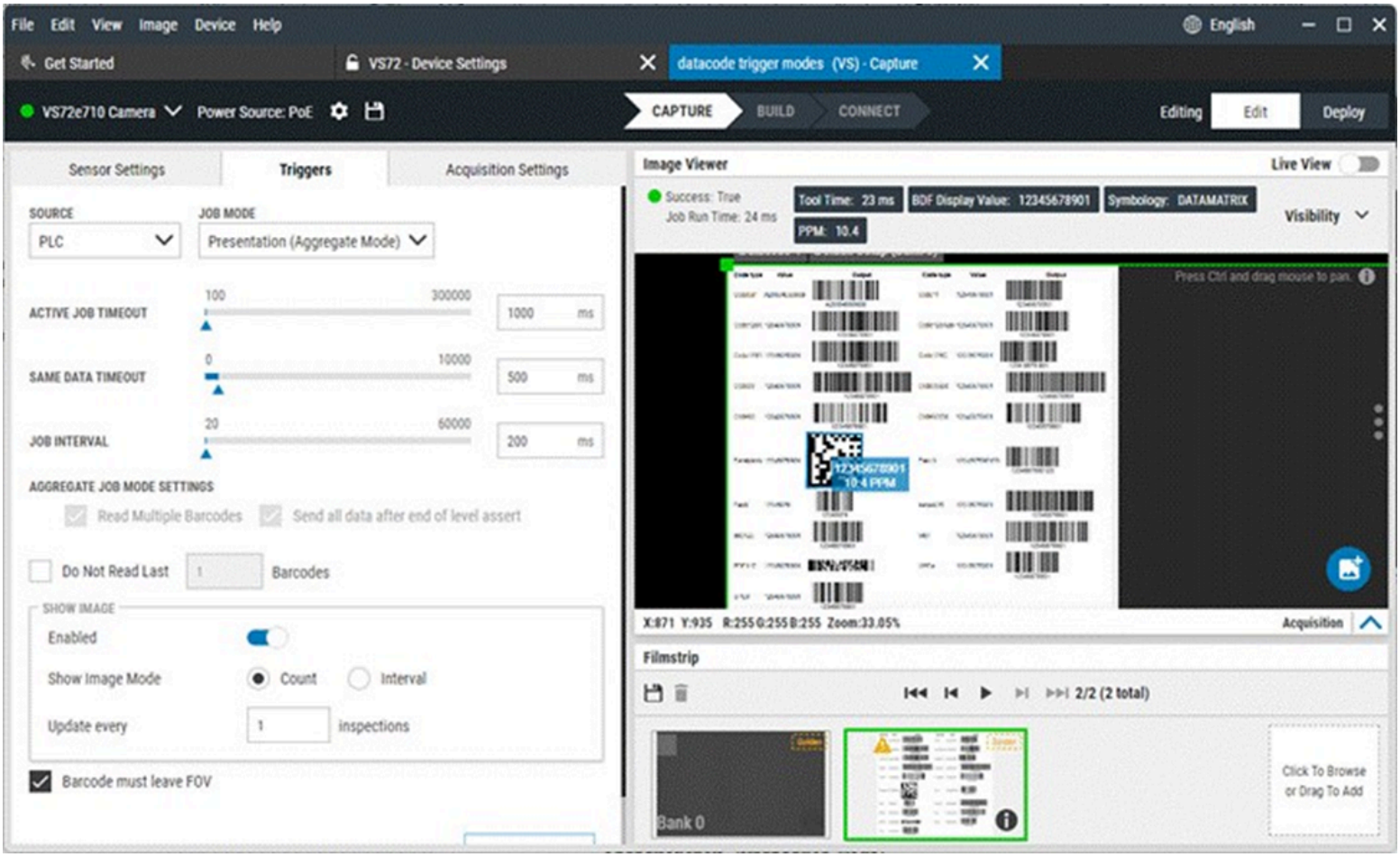


The first trigger event (toggling the trigger bit from 0 to 1) starts Presentation mode. Send the second trigger event (toggling the trigger bit from 0 to 1) before scanning the barcode to cancel Presentation mode.

NOTE: Presentation mode stops after the barcode successfully decodes and restarts when the next trigger event occurs.

Presentation (Aggregate Mode)

Presentation Aggregate mode is triggered the same way as Presentation mode. It aggregates all barcodes decoded during the active job timeout.

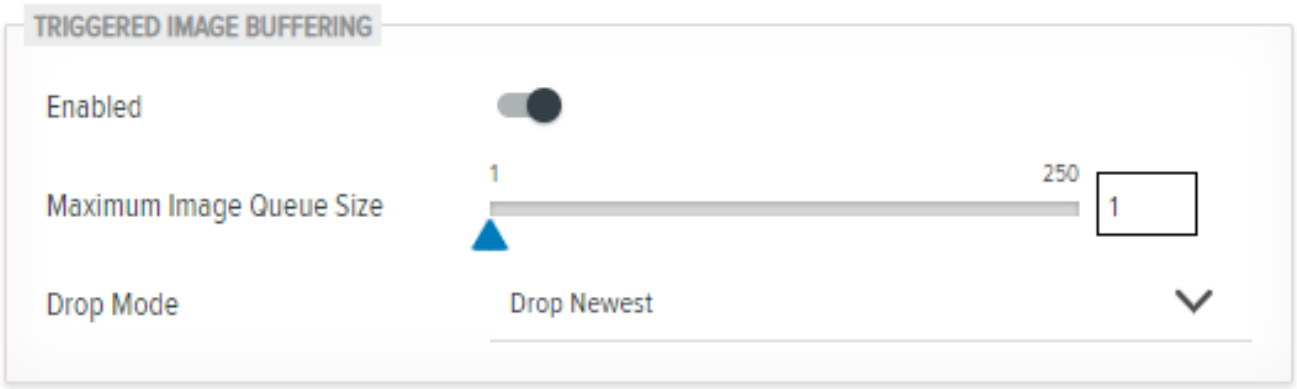


Triggered Image Buffering

When Triggered Image Buffering is enabled, there is an image processing event and one buffer event between each trigger occurrence.

Increase the **Maximum Image Queue Size** to configure the size of the image queue.

Select the **Drop Mode** from the menu to determine when images should no longer be included in the queue.

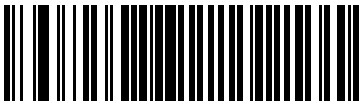

TRIGGERED IMAGE BUFFERING
Enabled
Maximum Image Queue Size
1
250
1
Drop Mode
Drop Newest

Using Autofocus Barcodes

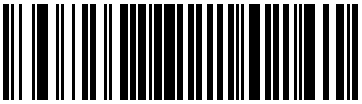
Use calibration barcodes to set the device's focus settings automatically.

NOTE: This process only adjusts the focus setting on the device and does not adjust exposure or gain.

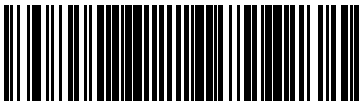
Bring one of the following calibration barcodes into the device field of view:



5s delay



8s delay



13s delay

1. Read one of the autofocus barcodes to start the focus-tuning process based on the selected delay (8s, 13s, or 18s).

NOTE: Observe the status LED flash every 500ms during the delay, providing time to bring the test barcode into the field of view to tune the focus on it.

2. During this delay, bring the desired test barcode into the field of view for the autofocus process to use as a reference. It is recommended to tune the focus of the device with a barcode type that is commonly decoded in your use case.

NOTE: Do not use the calibration barcode as the test barcode to calibrate focus against.

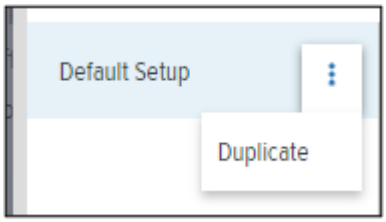
3. After the delay, the autofocus process starts and continues for 5-15 seconds.
4. Listen for the confirmation beeper sequence indicating the focus calibration process is complete.

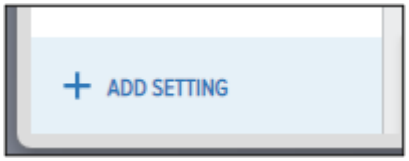
Using ImagePerfect+

ImagePerfect+ is an intuitive functionality used to grab additional images from the same trigger using a set of Acquisition Settings. It is helpful in acquiring images with different shutters, gain, focus, or different illumination configurations.

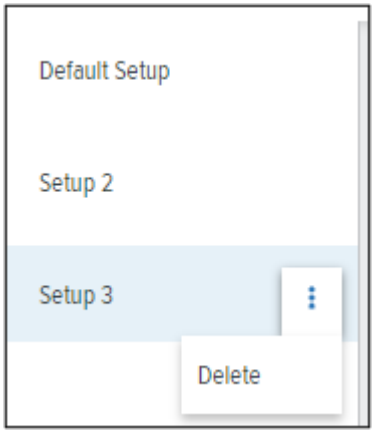
Single Shot and Periodic Single Shot are supported for use with ImagePerfect+.

Click **Default Setup** to clone the current configuration or click **Add Settings** at the bottom of the page to create a new configuration.





Delete a configuration by clicking the ellipses:



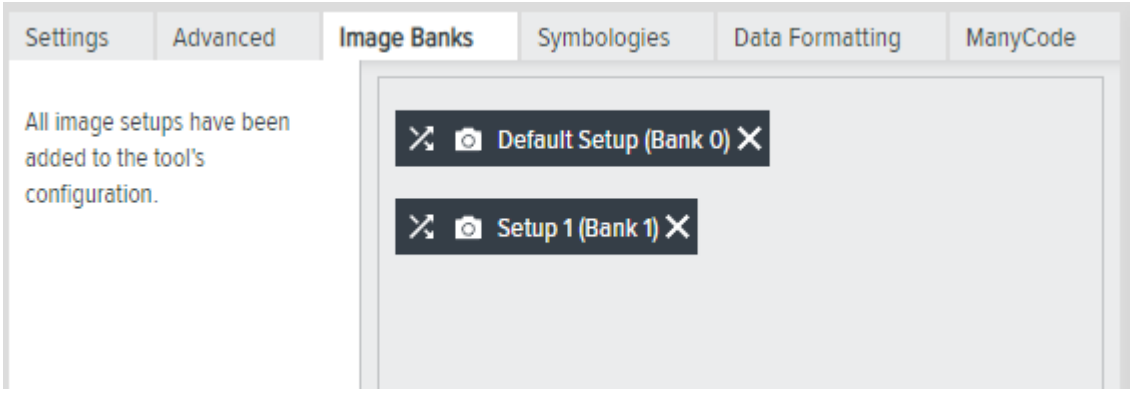
NOTE: FS10/xS20 devices support a maximum of 3 Acquisition Settings.

NOTE: xS40 and xS70 devices support a maximum of 16 Acquisition Settings.

Saving ImagePerfect+ Images

The smart camera saves only images processed by the decoder.

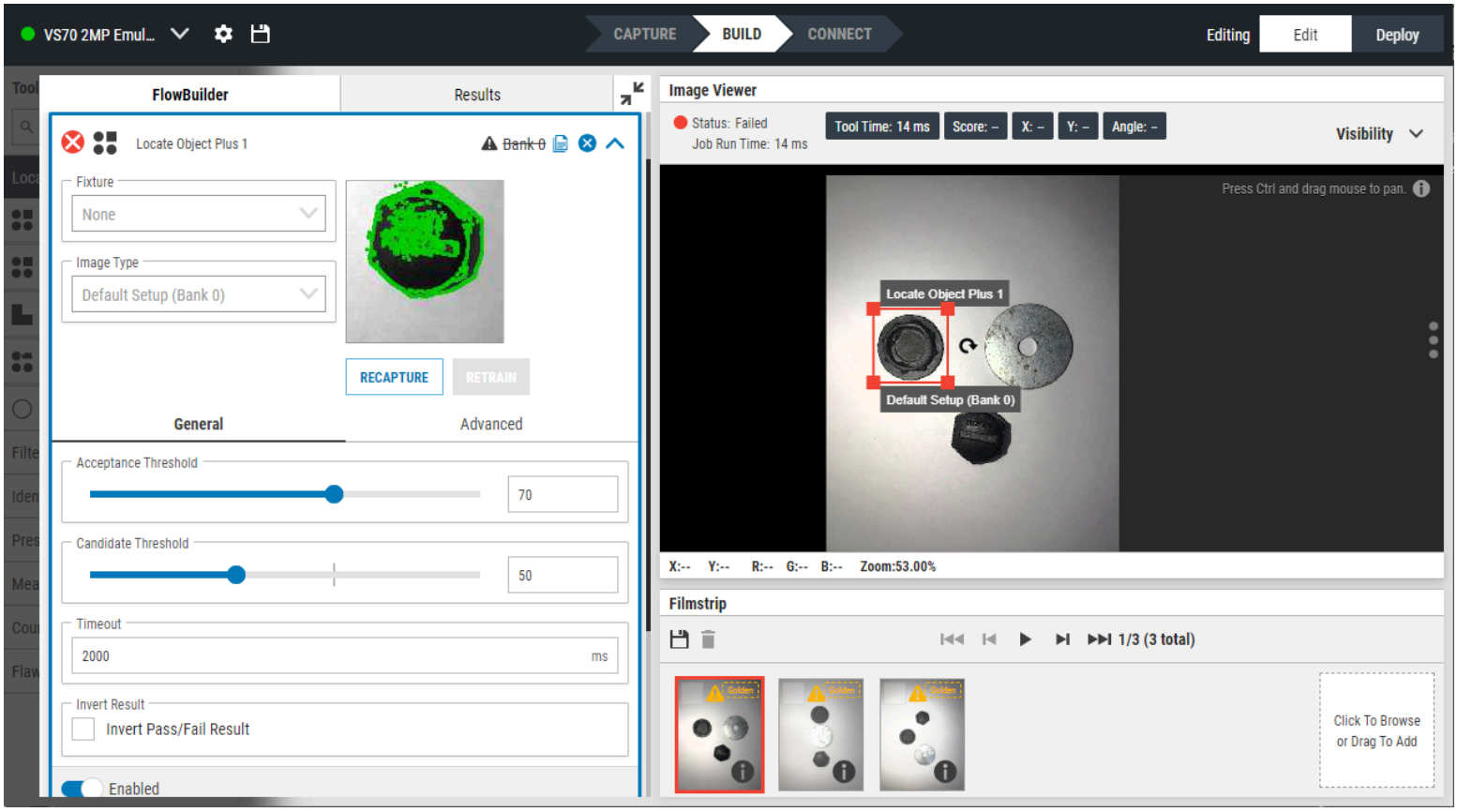
For the FTP to save bank_0 and bank_1, it is important to add both banks while configuring in **Build > Image Banks.**



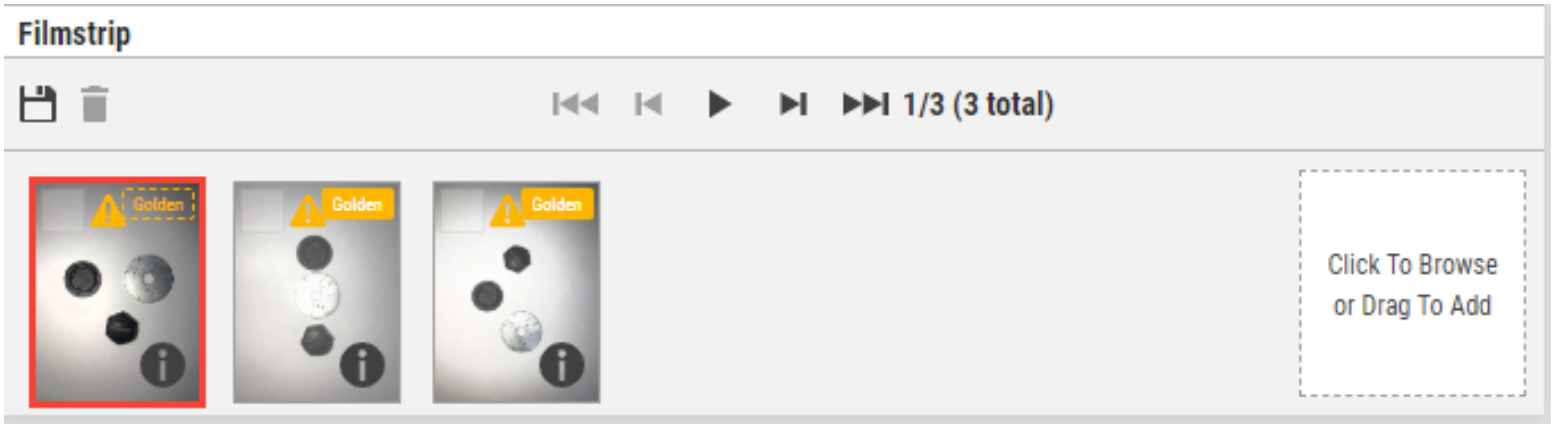
Using Golden Image Compare

Use Golden Image Compare to inspect two images from the Filmstrip simultaneously.

1. Acquire a frame using the capture button in the bottom right corner of the canvas.
2. Select the frames to set as Golden Image candidates by clicking the transparent yellow rectangle icon in the top right corner of each frame in the filmstrip.

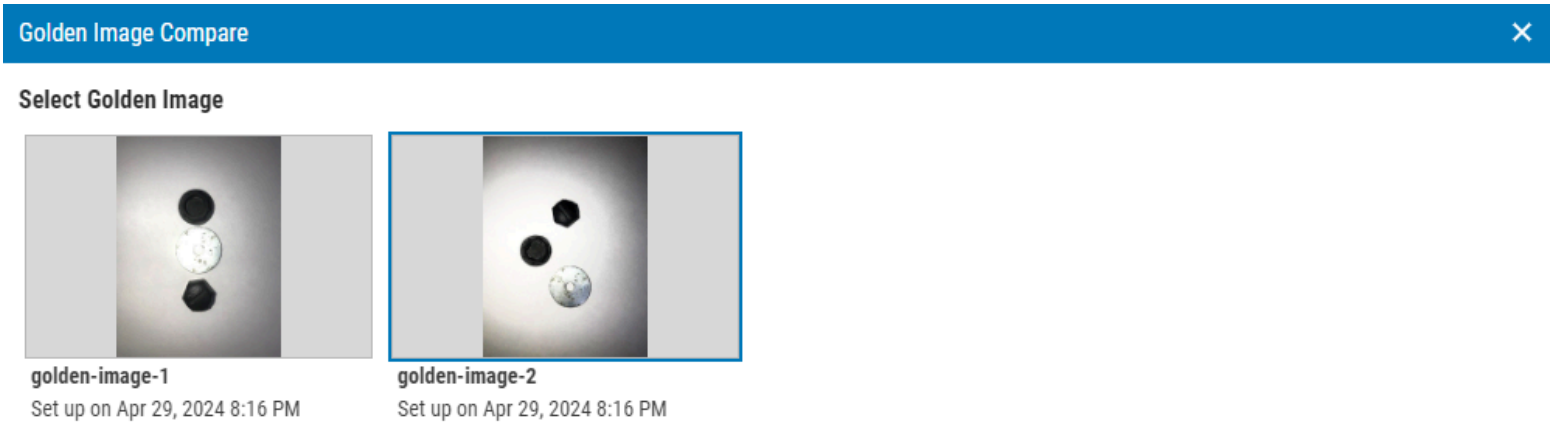


3. Next, capture a new image or select frames in the filmstrip to compare to the golden image.

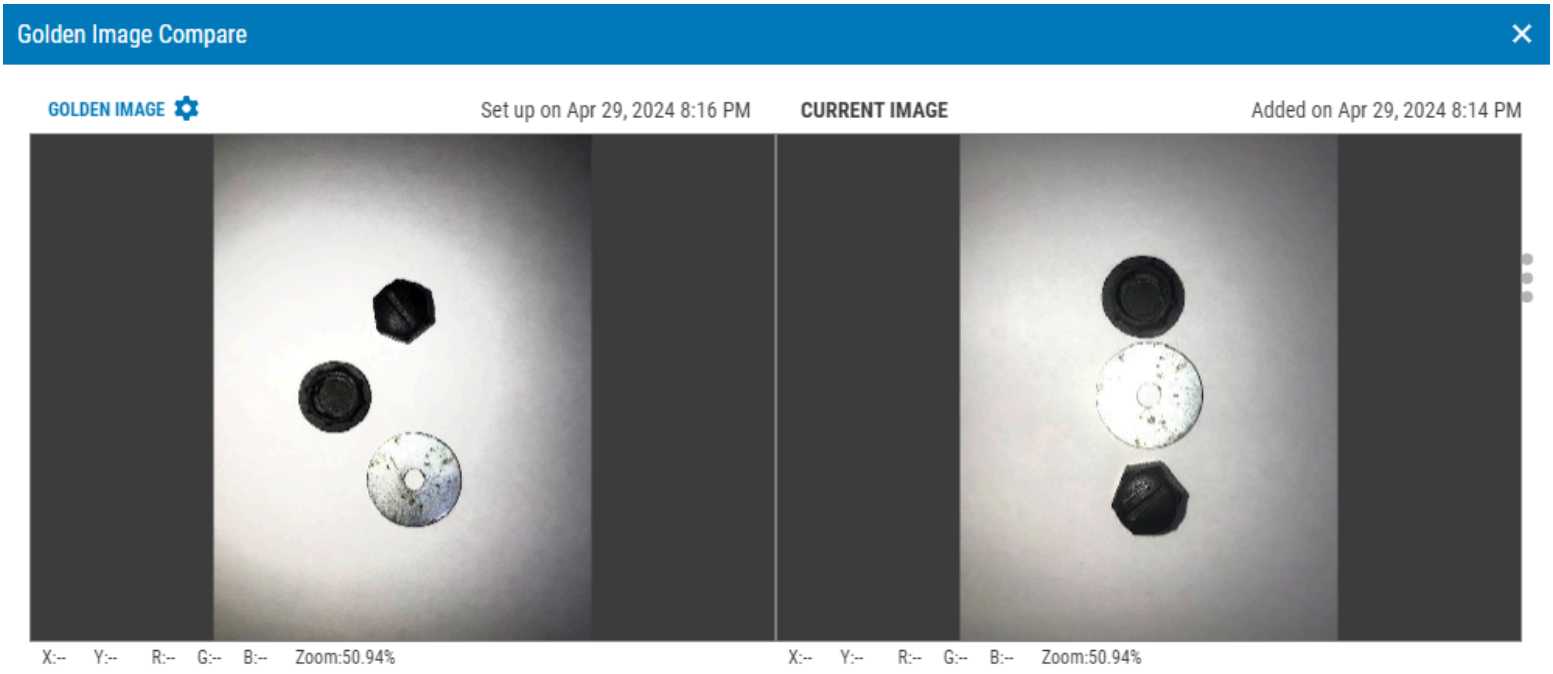


4. Hover over the three dots on the right of the canvas to view the dialog box menu and select the yellow Golden Image compare icon at the bottom of the menu.

5. Select the Golden Image candidate to compare to the current image in the canvas and click **Next**.



6. Observe the Golden Image Compare window to simultaneously inspect the Golden Image and the current image in the canvas side by side.



7. Hold the CTRL button on the keyboard while scrolling and zoom in or out using the scroll wheel on the cursor.

NOTE: Scroll to zoom in or out and inspect the same area of each image side-by-side. At the bottom of each image, use the window where the cursor is positioned to view a given pixel's XY position values and RGB color values.

Build

The **Build** tab facilitates job configuration and deployment for FIS and MV tools.

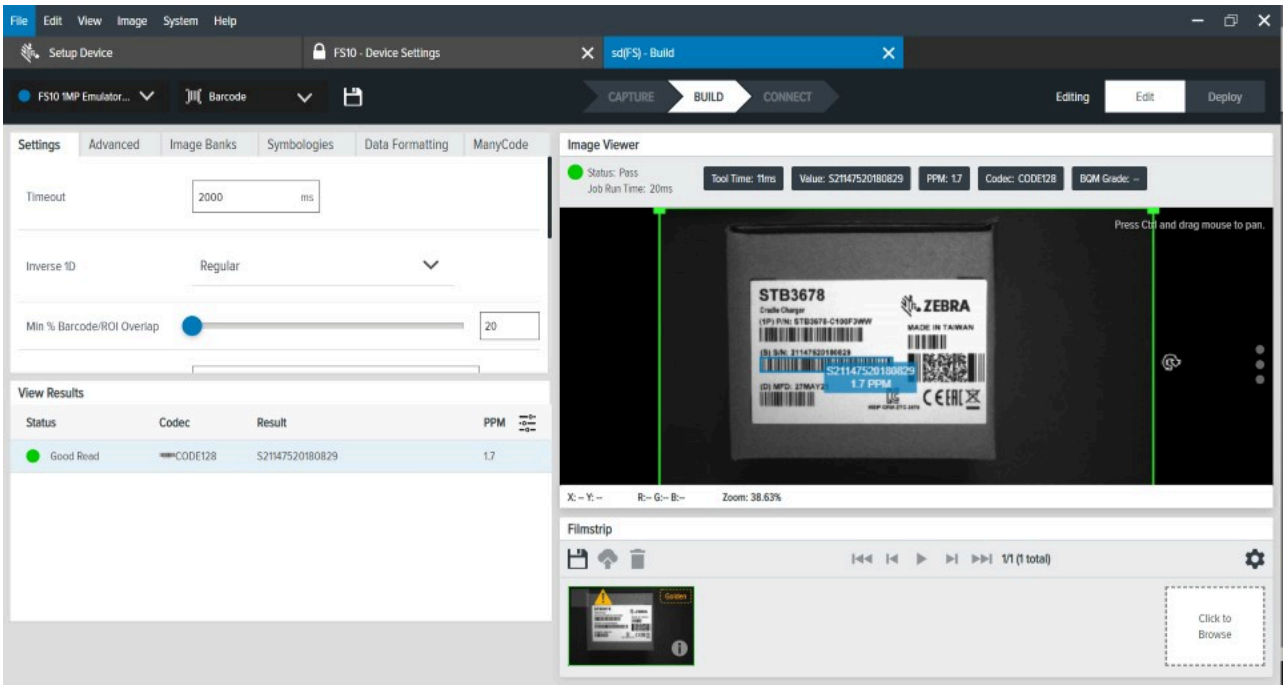


Table 17 Build Settings

Settings	Description
View Results	View the results of recent jobs.
Image Viewer	Observe the tool's analysis of the image.
Filmstrip	View the series of images captured by the device or upload a previously captured set of images.
Deploy	Run the job as configured in FlowBuilder.

Barcode Quality Metrics

Barcode Quality Metrics (BQM) are used to verify the quality of your printed barcode, so you know you can count on it to perform as expected in its intended setting. Enabling BQM reports an overall grade for the decoded barcode and grading based on various sub-components.

NOTE: The BQM mechanism in Aurora Focus implements the ISO 15415/ISO 15416 standard. Per ISO specifications, the optical reference arrangement requires high resolution, such as an effective resolution with a minimum of 10 pixels per module in width and height. For additional information, refer to ISO 15415.

NOTE: BQM supports 1D (Code 128 and Code 39) symbologies developed in accordance with ISO/IEC 15416:2016 and 2D (Datamatrix) symbologies developed in accordance with ISO/IEC 15415:2011. BQM current does not support QR codes.

Zebra BQM measures the barcode quality in the captured image, not the barcode itself. BQM score stability depends on image quality stability. To calculate the quality of the barcode, measure it in a strictly controlled manner using an image acquisition procedure.

BQM in Aurora Focus uses coarse grading. For example, the score step size is 1, per the ISO specs. A specific score of 2.9 is graded as 2 (C), although it is close to 3 (B).

BQM Best Practices

NOTE: While ISO specs recommend a minimum of 10PPM, achieving a PPM of 15 or greater is recommended to ensure consistent BQM results.

Follow the best practices outlined in this section to ensure optimal BQM performance.

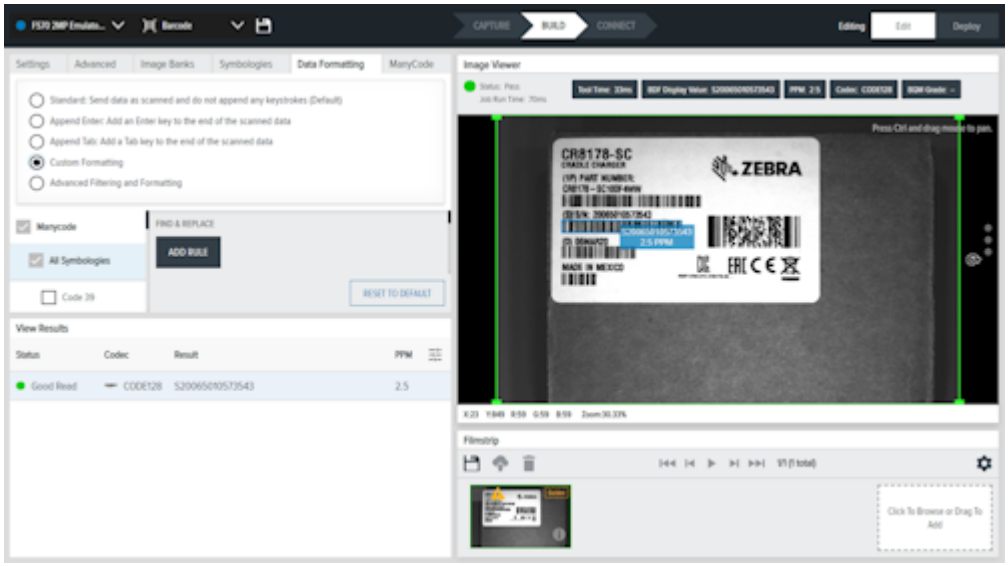
- Consistent lighting with high contrast, little to no gradient, and clear focus is critical to achieving optimal BQM results. Maximum pixel intensity should be between around 160 and 200 with reasonably good contrast. Ensure there are no defects or glare on the image; no specular spots should be present.
- Consider using a polarized light filter accessory to eliminate glare. A clean and consistent barcode is essential for BQM grading; a polarizer can be helpful to eliminate glare that could compromise the barcode image.
- Position the barcode close to the center of the Field of View (FOV) to minimize any optical distortion and ensure that the barcode is parallel to the camera plane on the device so the edges are parallel to the image border.

Custom Formatting

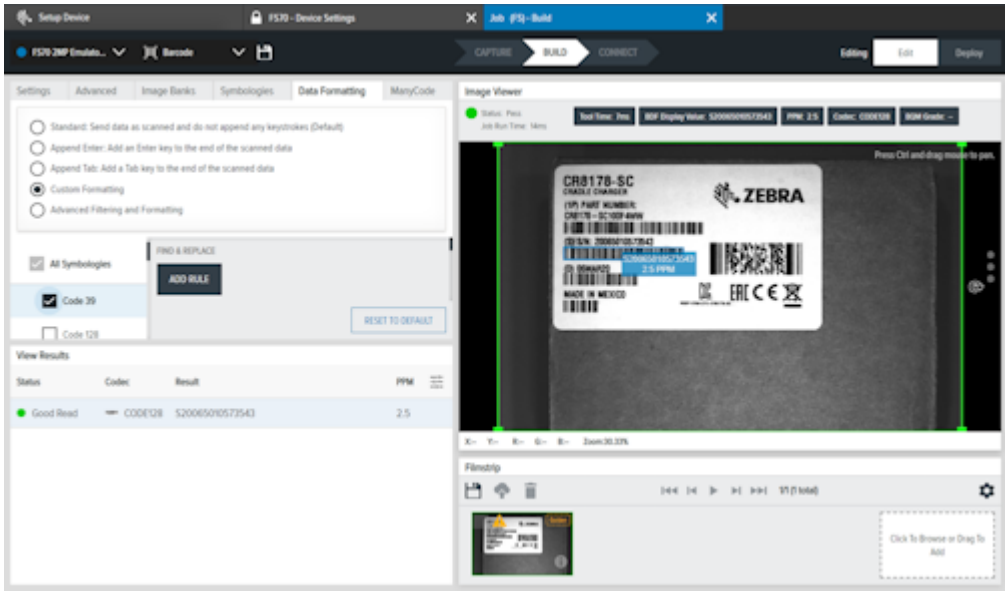
Use Custom Formatting to retrieve specific data by adding rules and delimiters.

To configure **Custom Formatting**, access the **Data Formatting** tab from the **Build** tab.

1. Select the **Custom Formatting** radio button.



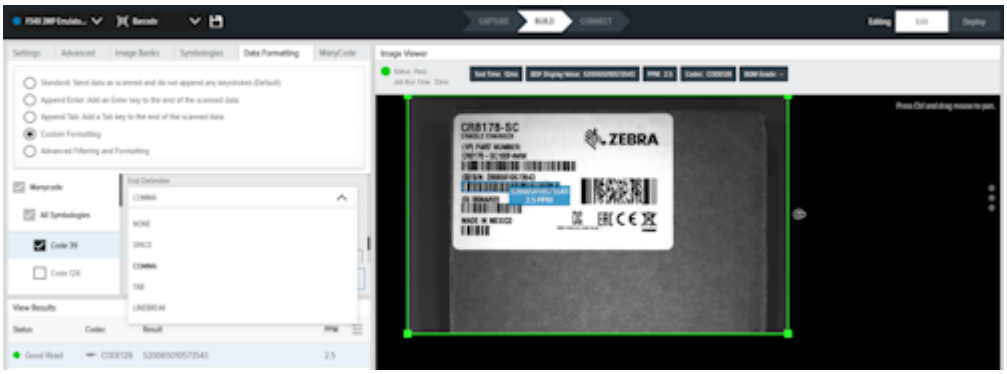
2. Select a symbology from the **All Symbologies** list.



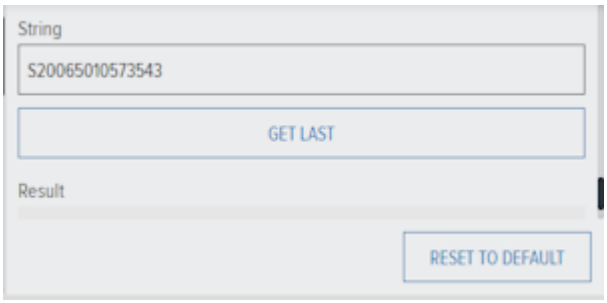
3. Depending on the symbology, click **Add Rule** to add a find and replace rule.
4. Click next to **Find** to add a data type to identify.
5. Next, click next to **Replace With** to designate a data type to replace the data type identified in the Find form field.
6. Click next to the **Prefix** form field to add a data type. Follow the same steps for **Data** and **Suffix.**

7. Select a **Data Type Delimiter** from the drop-down.

8. Select a **End Delimiter** from the drop-down.



9. Click **Get Last** to retrieve the **String** or **Result**.



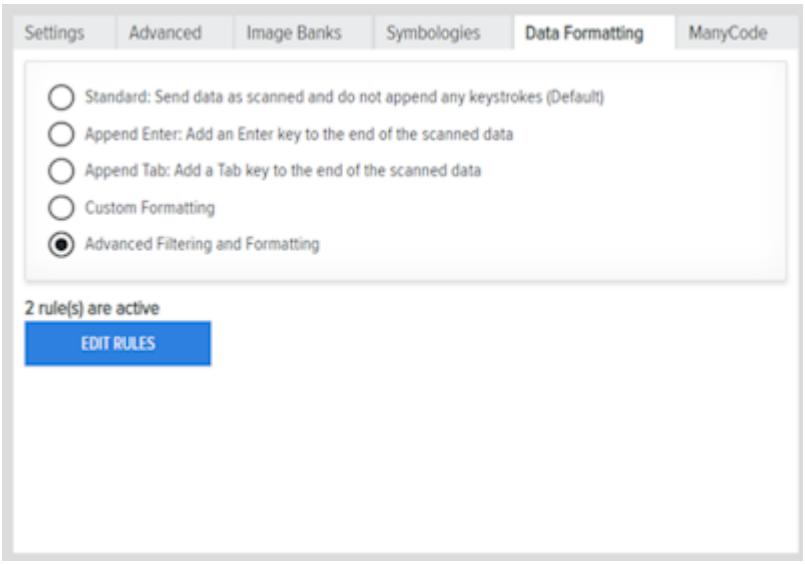
Advanced Filtering and Formatting

The following functionalities are not supported in Aurora Focus:

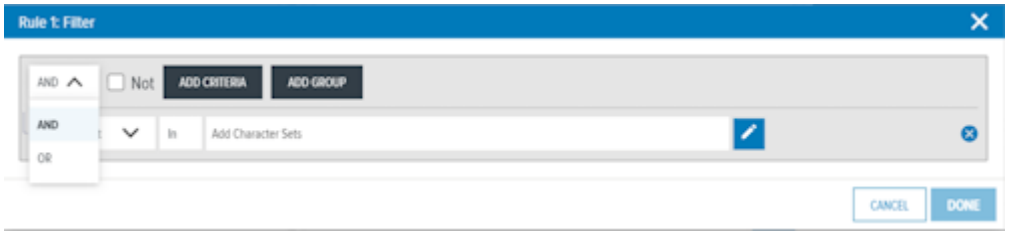
- Non-standard, customer-specific check digit calculations
- Location based Pattern Match requirements based on barcode positional dependencies such as Code 128 is on the left of a UPC
- Interjection of time delays between transmitted barcode data
- Non-ASCII values

To configure Advanced Filtering and Formatting settings in Aurora Focus:

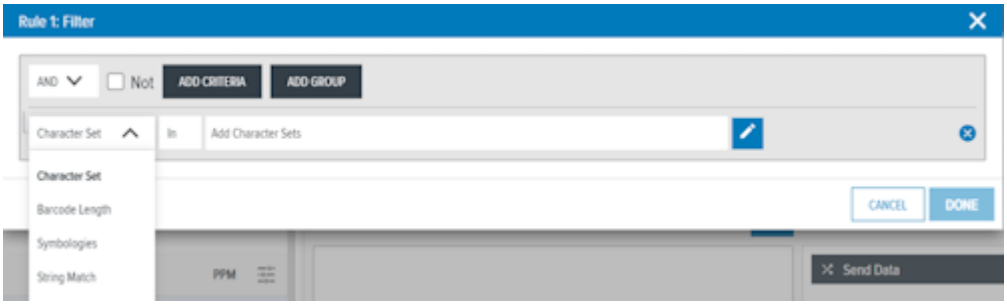
1. Select the **Data Formatting** tab to access **Advanced Filtering and Formatting**.
2. Select the **Advanced Filtering and Formatting** radio button.
3. Click **Edit Rules** to create a new rule or edit an existing rule. Rename the rule if necessary.



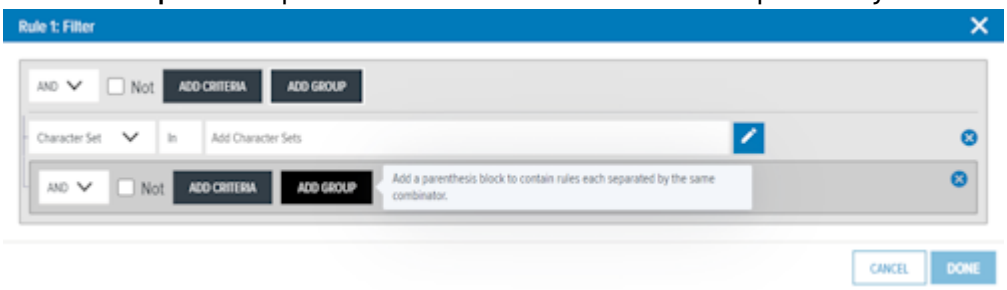
4. To add a **Filter**, click **Edit** and select an **And** or **Or** operation from the drop-down menu, or select the **Not** checkbox to invert the result.



   a. Click **Add Criteria** to add a condition to the rule from the drop-down menu.



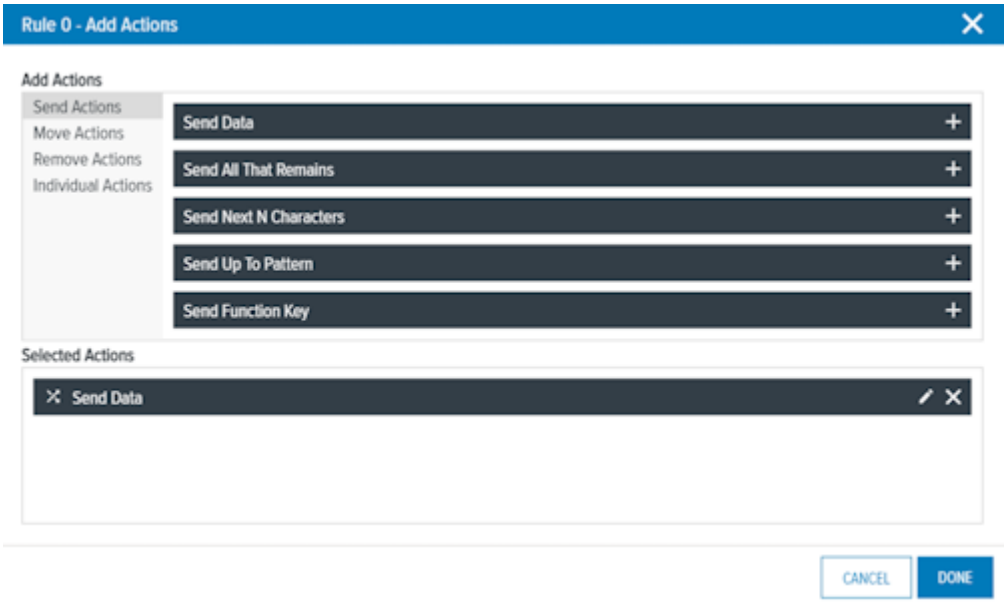
b. Click **Add Group** to add a parenthesis block that contains rules separated by the same combinator.



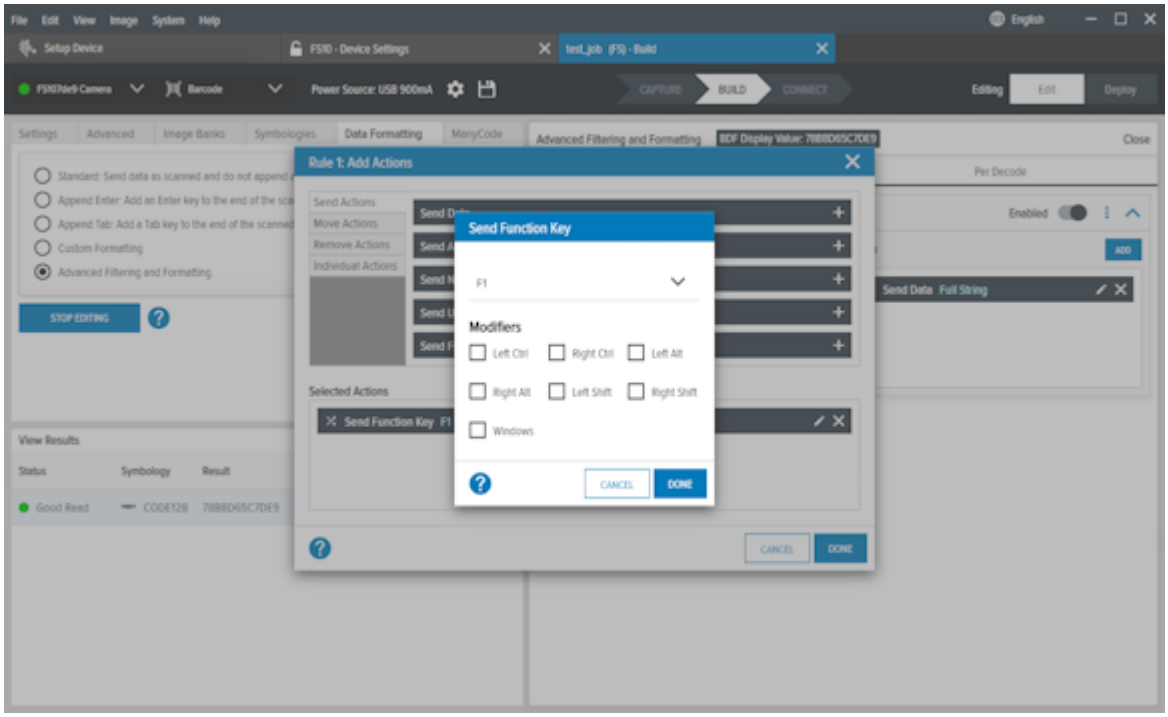
5. To add an **Action**, click **Add**.

   a. Select a type of **Action** (send, move, remove, or individual).

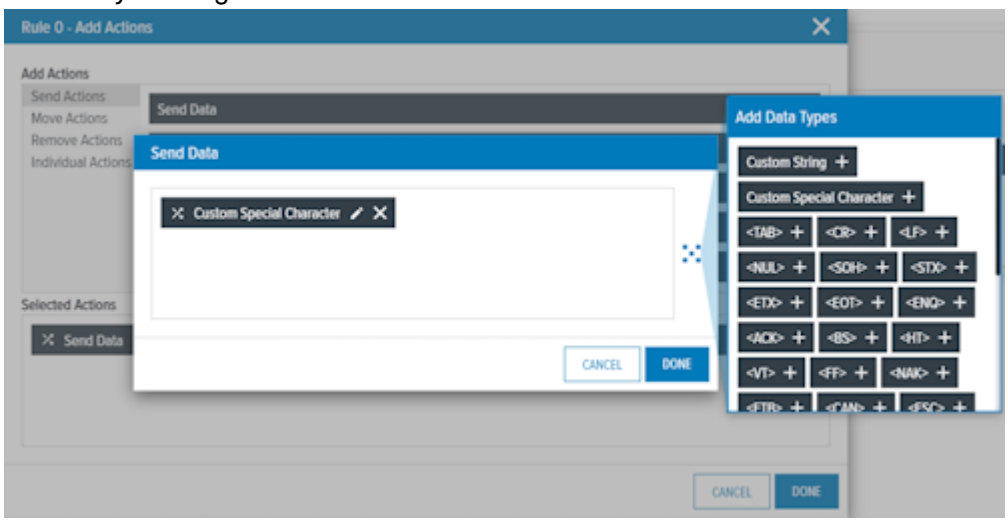
   b. Click + to specify an **Action** to add to the list of selected actions by clicking the plus sign.



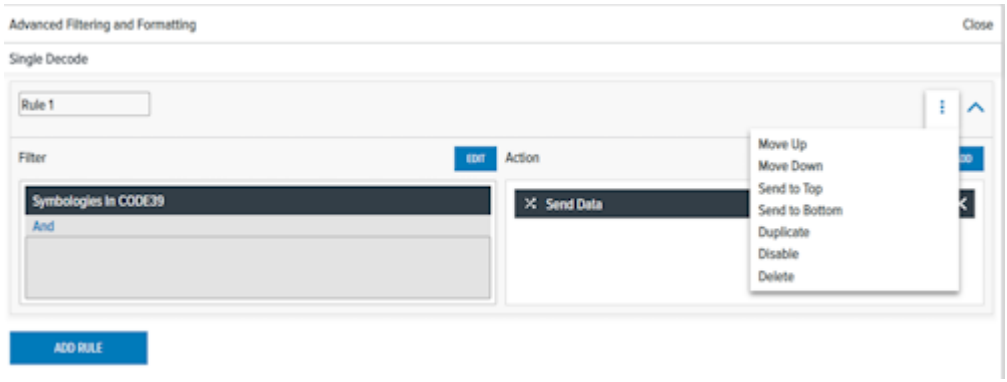
   c. Use **Send Function Key** to configure which function key to send as the keystroke for the HID output interface. The range of keys is from F1 to F24. Add modifiers when applicable, including Left Ctrl, Right Ctrl, Left Alt, Right Alt, Left Shift, Right Shift, and Windows.



d. Edit the action by clicking **Edit** on the selected action.



6. After a **Filter** and **Action** are identified for the rule, and rules are managed by clicking the ellipses.



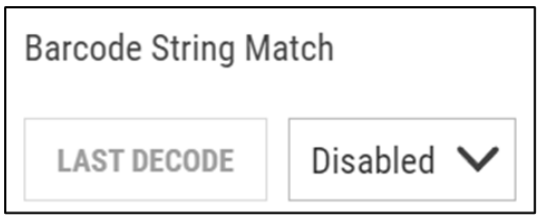
Using Match String

Use Match String to determine if the code has a specific string present.

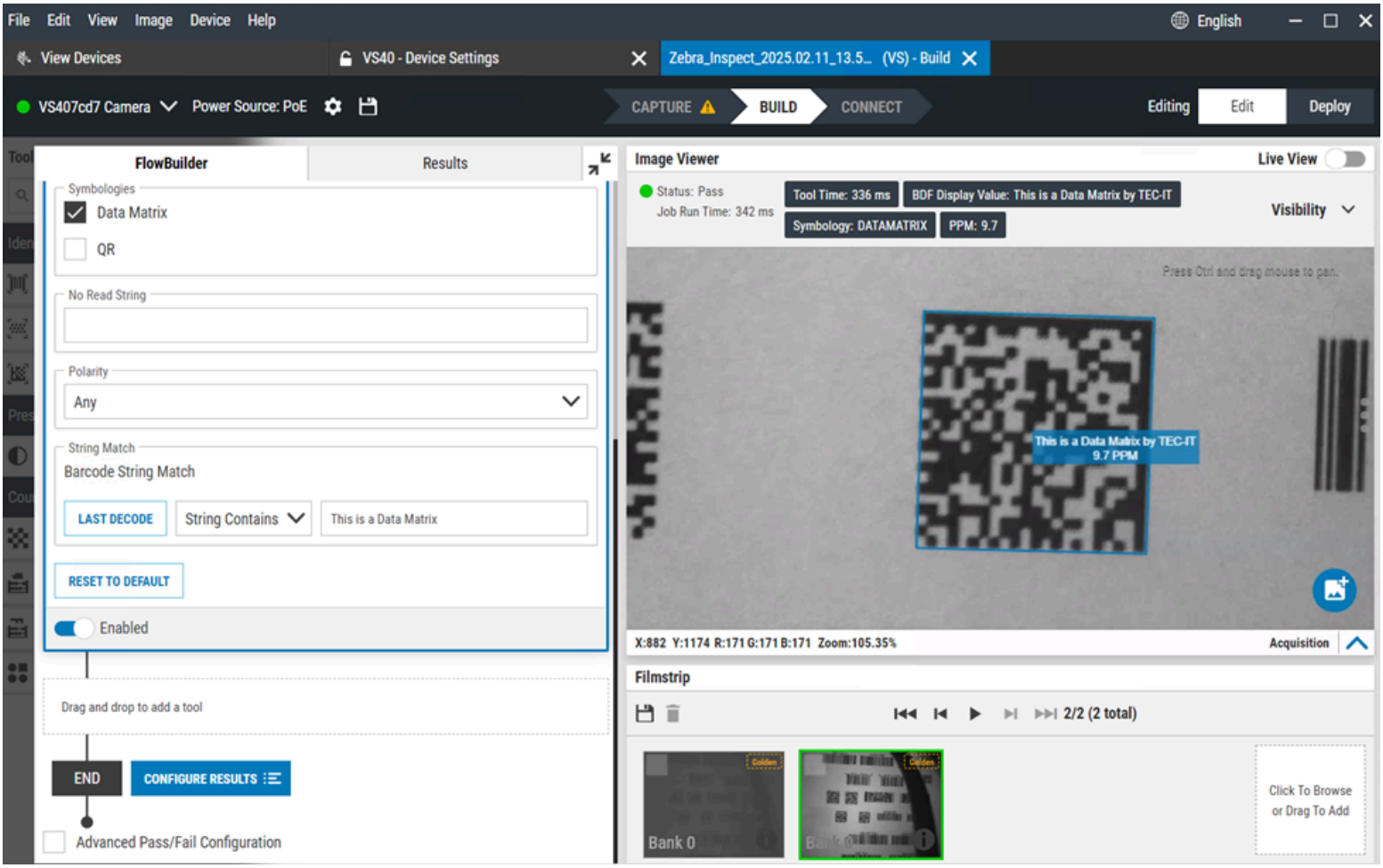
Match String supports the following tools:

- Datacode
- Deep Learning OCR
- Read Barcode

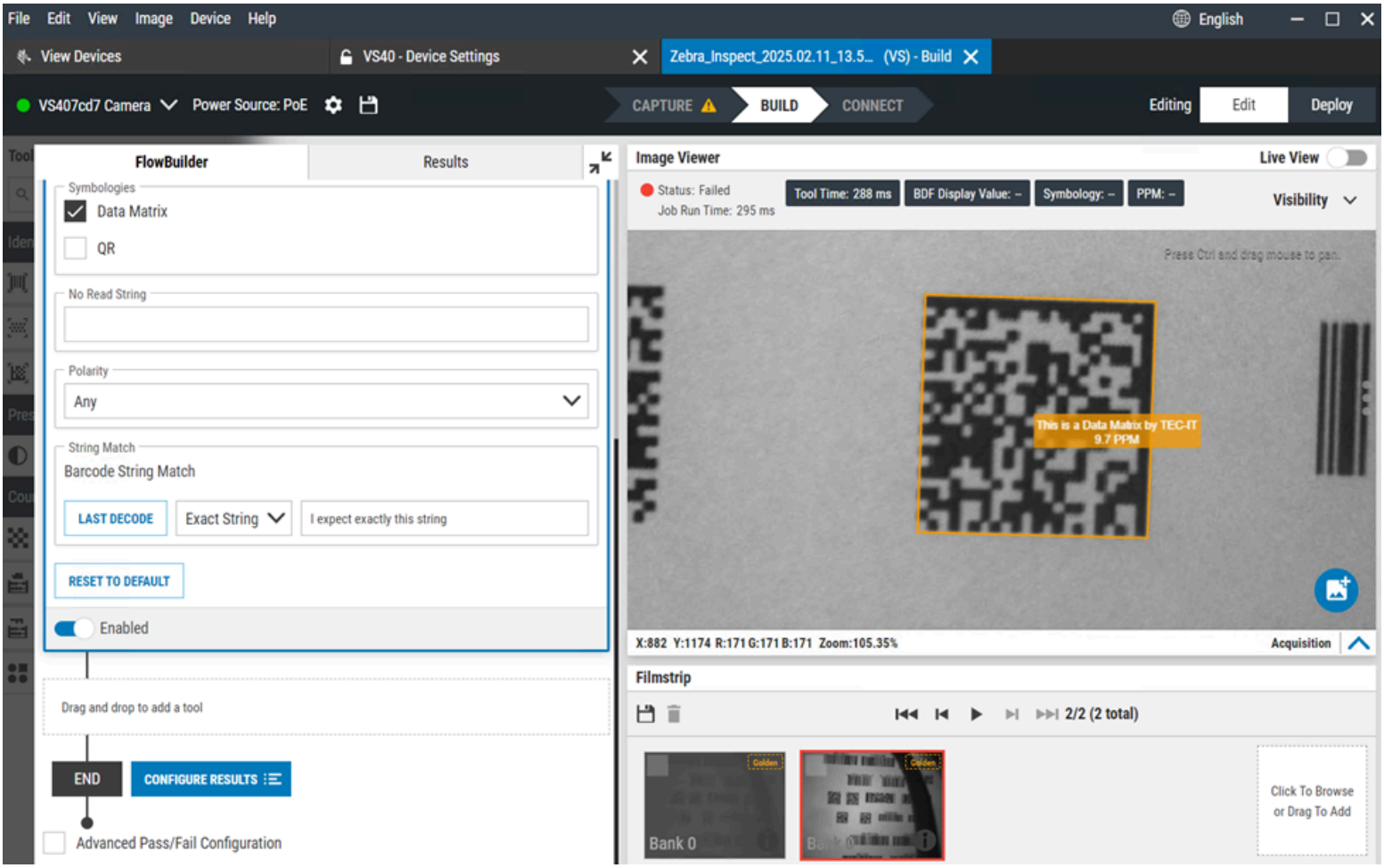
For cases where match string functionality is not necessary, ensure **Barcode String Match** is disabled.



In cases where the data code contains the given string, the job passes.



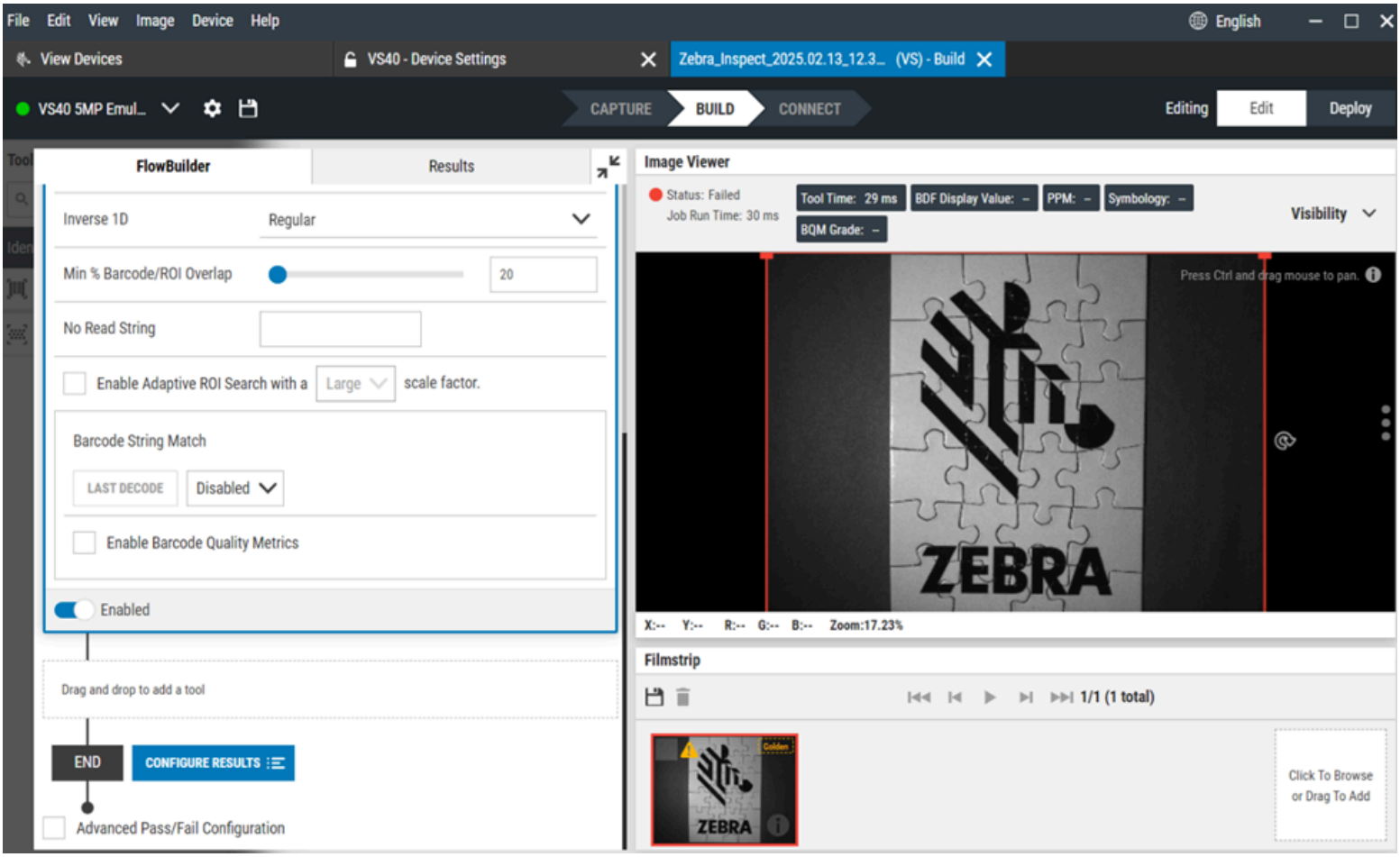
In cases where the data code does not contain the given string, the job fails.



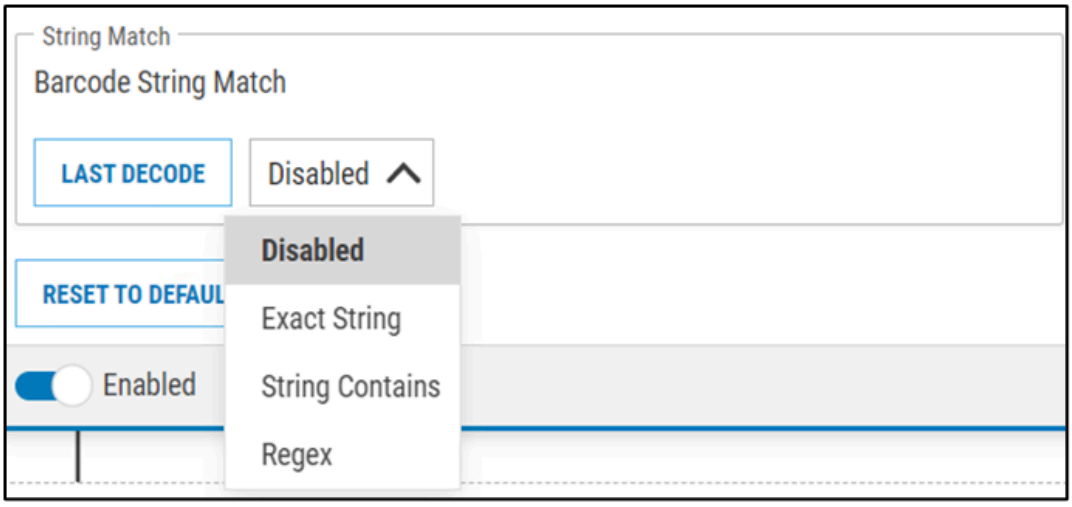
Using Set and Get Match Strings

Match Strings are configurable in Aurora Focus for Datacode, Deep Learning OCR, and Read Barcode tools.

1. Add the tool in **FlowBuilder** using the **Build** tab to view match string options.



2. Select the **Barcode String Match** mode from the drop-down menu.



- Disabled (Default) - no match string is used, and every string is accepted.
- Exact string - the result must match the exact string provided.
- String Contains - the result must contain the given string.

- Regex - result acceptance follows the given regex rule.

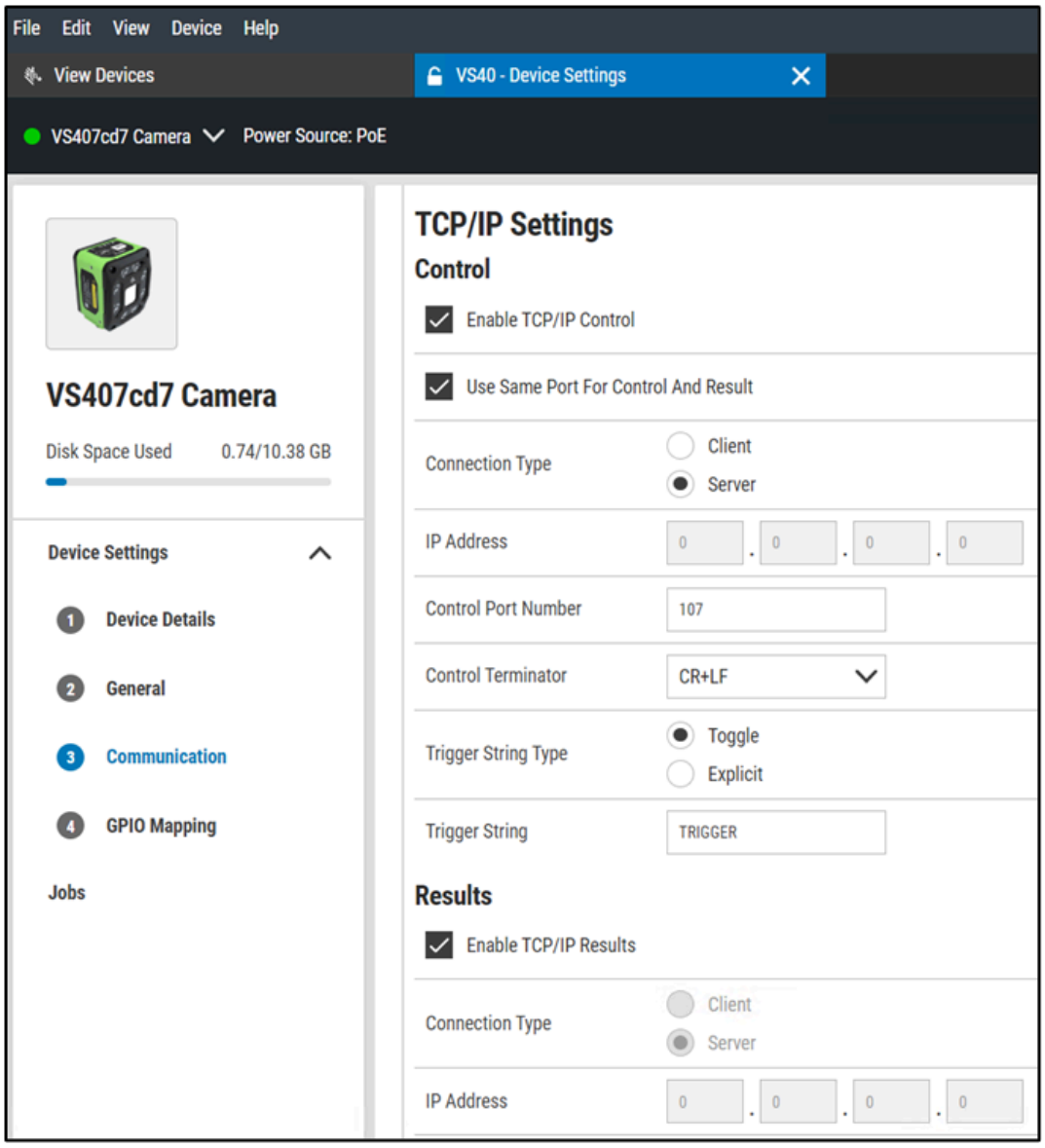
NOTE: Barcode String Match mode is disabled by default and the match string value is empty. In this case, the tool accepts all read codes.

3. Click **Last Decode** to use the last successful read string as the current match string value.

Command Channels

Match strings are set by channels that support controlling devices that use commands such as TCP/IP or Serial Port.

NOTE: Ensure that **Enable TCP/IP Control** and **Enable TCP/IP Results** are enabled in **Communication** settings.



Enabling **TCP/IP Results** The command results must be read from the channel. Enabling this setting is recommended even in cases where the return value is unnecessary for the set command because it can return useful information about possible errors.

NOTE: You cannot get a match string value if this option is disabled.

Available Commands

Parameters are preceded with two strings that inform what action should be performed and which job parameter it applies to.

The first string informs whether the command should get **GetJobParameter** a parameter or set **UpdateJobParameter** a parameter as the value. The second string informs which job parameter should be updated (in this case, use **matchstring**).

Provide the parameters in the following command line format:

```
--parameter_name parameter_value
```

For string parameter values, use double quotes:

```
GetJobParameter matchstring --toolname “Datacode 1”
```

For boolean parameter values, use true or false:

```
UpdateJobParameter matchstring --value "XYZ" -- persistent true
```

For enum parameter values, enter the parameter value without quotes:

```
GetJobParameter matchstring --format plain
```

In cases where arguments do not take any value, only provide the parameter name:

```
GetJobParameter matchstring --help
```

Get Match String Commands

If an optional argument is not provided, use the default value.

NOTE: Skipping a required parameter results in command failure.

Get Match String command:

```
GetJobParameter matchstring [optional parameters]
```

Parameters:

Table 18 Get/Set Command Parameters

Parameter	Argument	Optional	Description
`help`	no argument	Yes	Produce a help message with a description of all commands. If provided, other arguments are ignored, and only the help output is returned.
`toolname`	string (**empty string** by default)	Yes	Friendly name of a tool. If provided, the result will contain only data for the given tool (and only if the given tool is found and supports a match string). Otherwise, the result contains data for all tools.
`format`	enum (**plain** / base64)	Yes	The format used to present the match string in the result.

Example input:

```
GetJobParameter matchstring --format plain
```

Example result:

```
[GetJobParameter result] Get match string result: ["GetMatchStringEntries",
[{"MatchMode":"STRING_EXACT","MatchString":"ABC","Toolname":"Datacode 1"}]]
```

Set Match String Commands

If an optional argument is not provided, use the default value.

NOTE: Skipping a required parameter results in command failure.

Set Match String Command:

```
UpdateJobParameter matchstring --value “match string value” [optional
 parameters]
```

Table 19 Set Match String Command Parameters

Parameter	Argument	Optional	Description
`help`	N/A	Yes	Produce a help message with a description of all commands. If provided, other arguments are ignored, and only the help output is returned.

Table 19 Set Match String Command Parameters (Continued)

Parameter	Argument	Optional	Description
`toolname`	string (**empty string** by default)	Yes	Friendly name of a tool. If provided, the match string is applied only to the tool with the given tool name; otherwise, it is applied to all tools supporting the match string. **NOTE:** If this parameter is omitted or an empty string, other parameters are applied to all tools supporting the match string.
`format`	enum (**plain** / base64)	Yes	Format of provided match string. Setting the format to base64 makes the value parameter to be interpreted as base64. **NOTE:** This applies only to the value you insert in the command and doesn't influence the way of storing match strings internally by Aurora Focus.
`persistent`	bool (**true** / false)	Yes	If true, setting the match string will be persistent (saved in the database). Otherwise, it will only be applied to a currently deployed job.

Table 19 Set Match String Command Parameters (Continued)

Parameter	Argument	Optional	Description
`match_mode`	enum string (**unknown** / disabled / string_contains / string_exact / regex)	Yes	Match mode: • unknown - either previous mode from job is used or string_contains, if it was not set; • disabled - every match string is accepted, regardless of set value; • string_contains - result must contain given match string in any place; • string_exact - result must exactly match the given match string; • regex - result acceptance follows the given regex rule.
`value`	string	No	Match string value. If the format parameter is set to base64, then the value is interpreted as base64; otherwise, it is plain text.

Example Input:

```
UpdateJobParameter matchstring --value "XYZ"
```

Example Result:

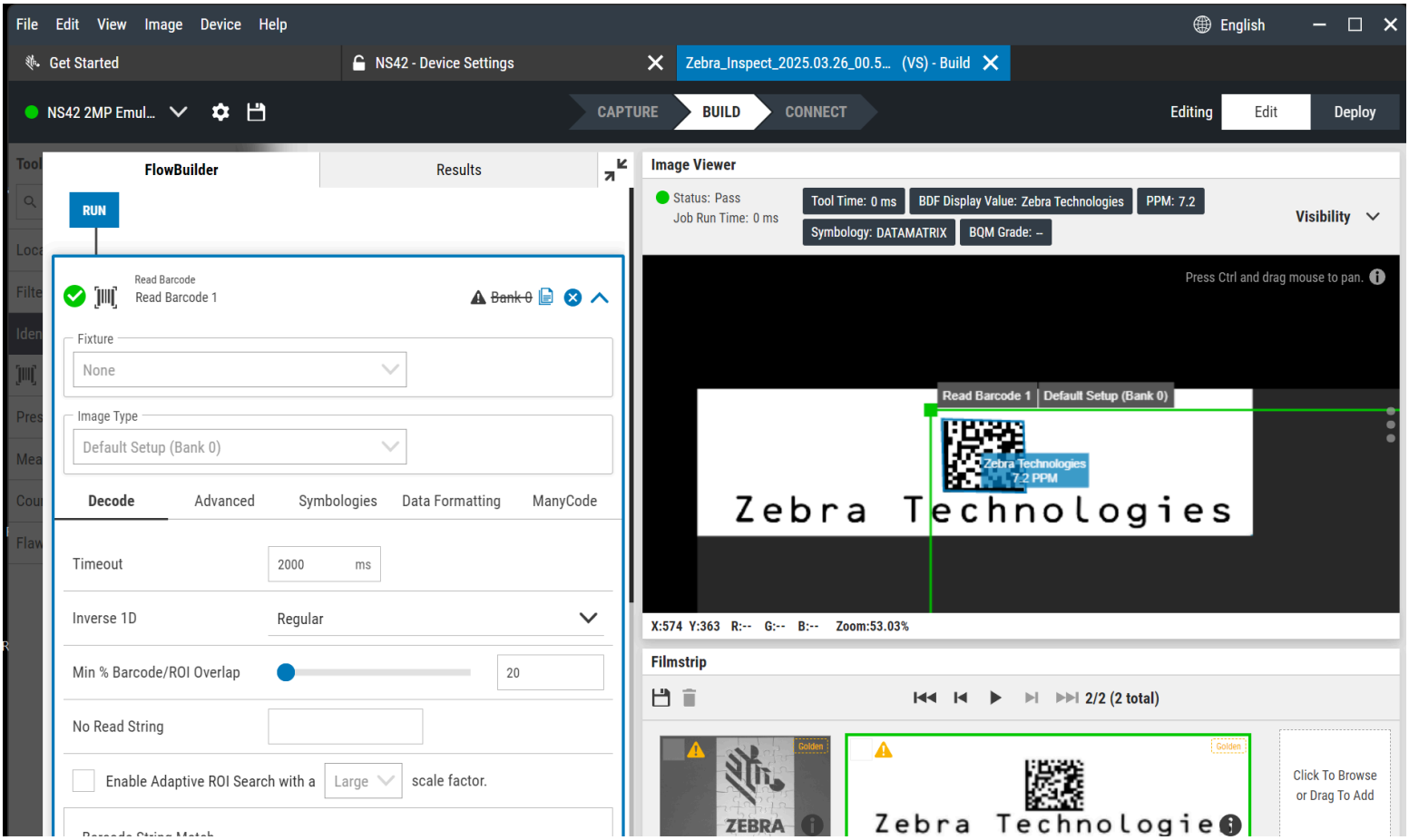
```
[UpdateJobParameter result] Set match string, success
```

Using Fixturing Tools

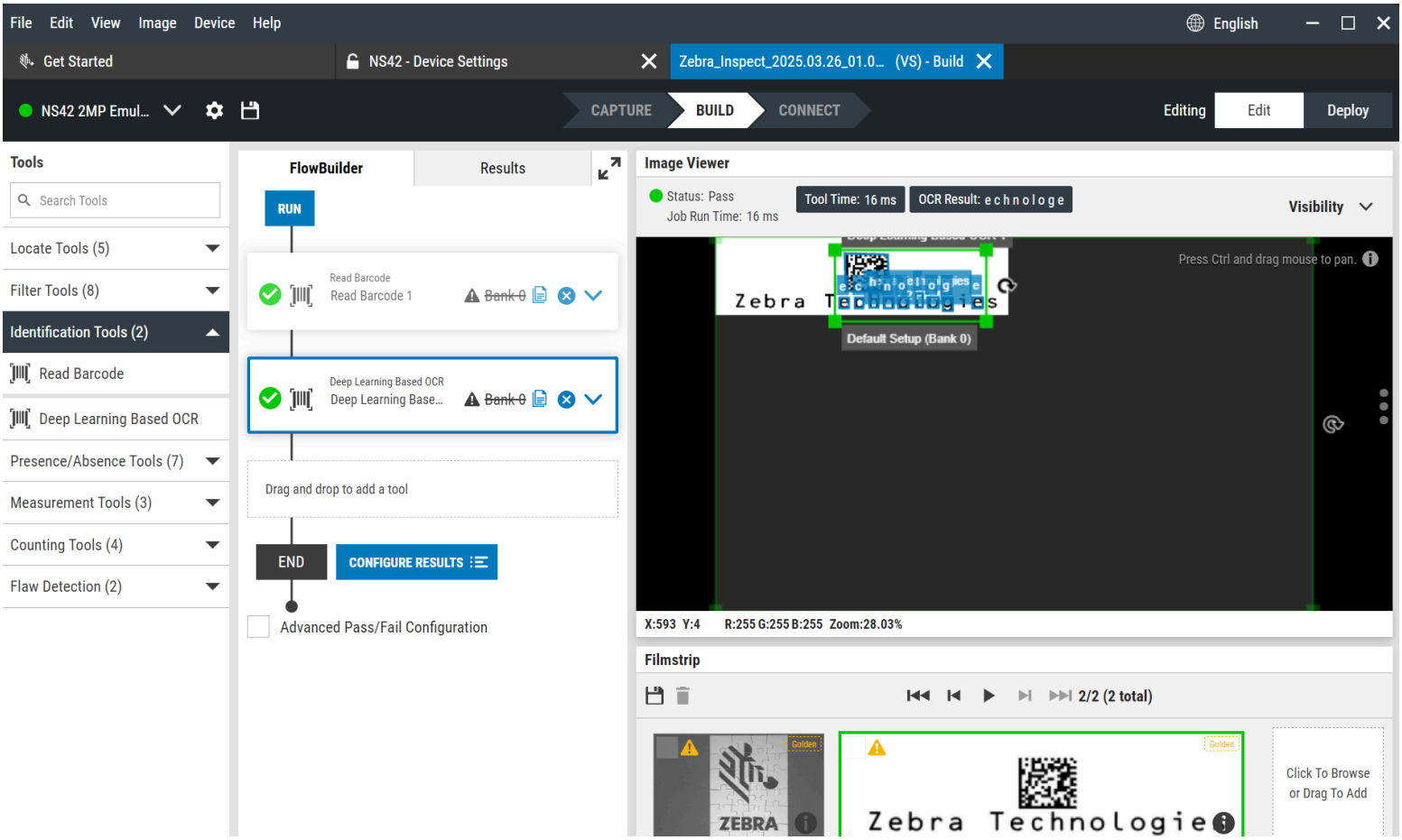
Use fixturing tools to focus on a specific symbology in environments where codes are processed rapidly and may be presented in different orientations.

NOTE: All 1D and 2D symbologies are compatible with fixturing tools.

1. Create a new job.
2. Add the Read Barcode tool to the FlowBuilder.

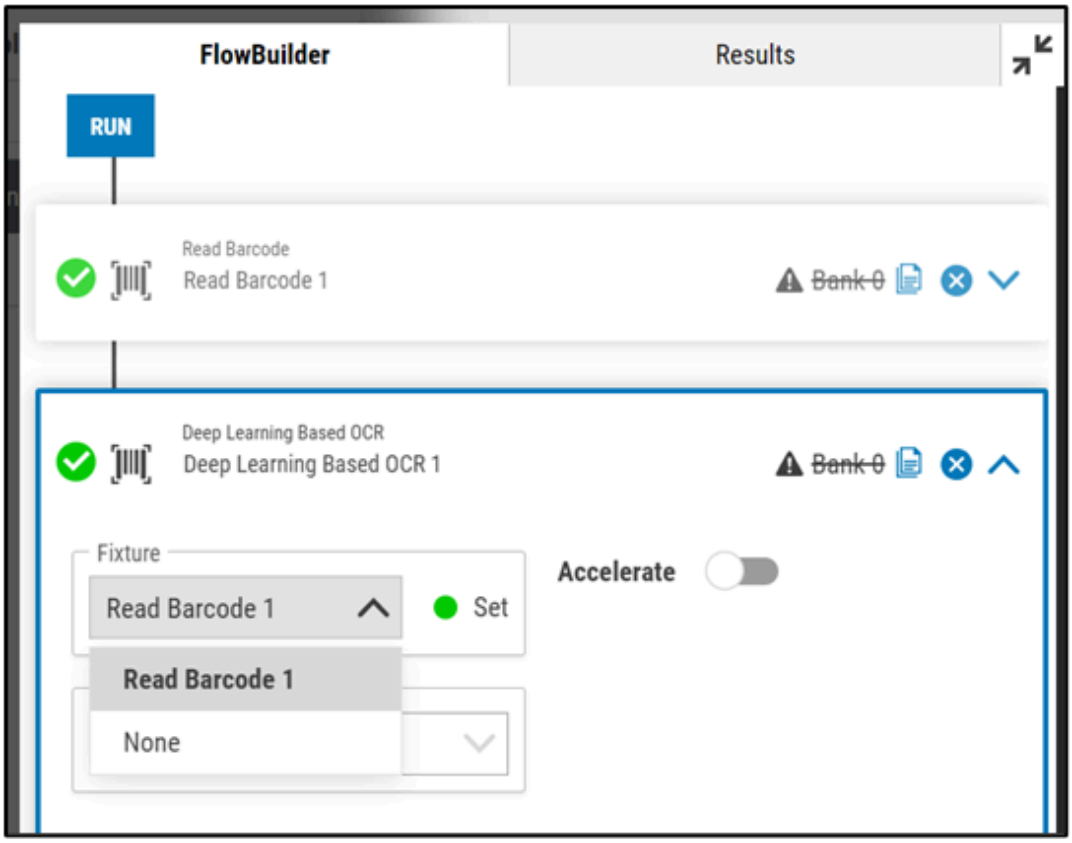


3. Add an additional tool, such as Deep Learning OCR.



4. Ensure that Manycode is disabled before using fixturing.

5. Select the first tool (Read Barcode) from the Fixture drop-down list.

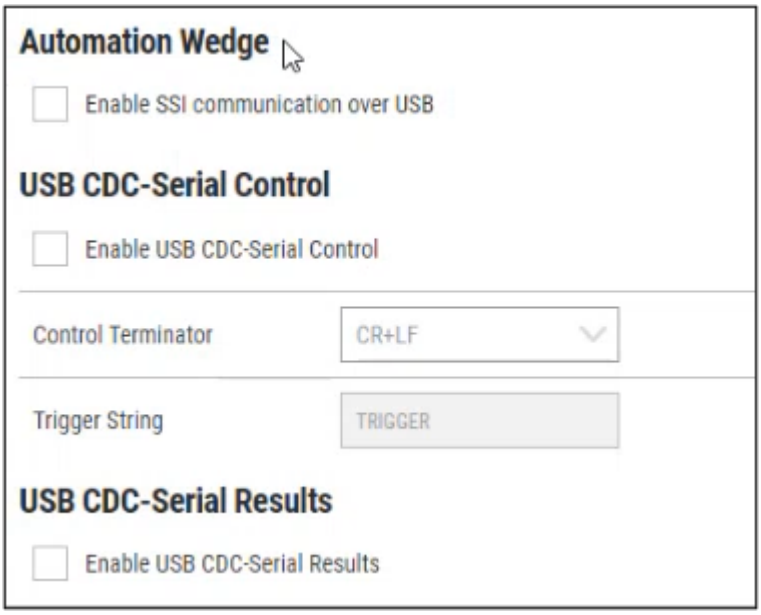


6. Deploy the job and move the test image. Observe the Deep Learning OCR tool's ROI following the Barcode tool results.

Using Automation Wedge

Automation Wedge transmits data captured by the smart camera to an application running on a Zebra mobile computer. Use Automation Wedge to enhance productivity by streamlining the process of collecting and utilizing data from physical sources.

1. Enable Unrestricted USB power when using USB for communication and power. This is a required step.
2. Connect the USB cable from an FS40 or FS42 device to the USB power on the docking station.
3. In Aurora Focus, navigate to the **Communication** section in **Device Settings**.
4. Click **Enable SSI communication over USB** under **Automation Wedge** in the **USB Settings** section.

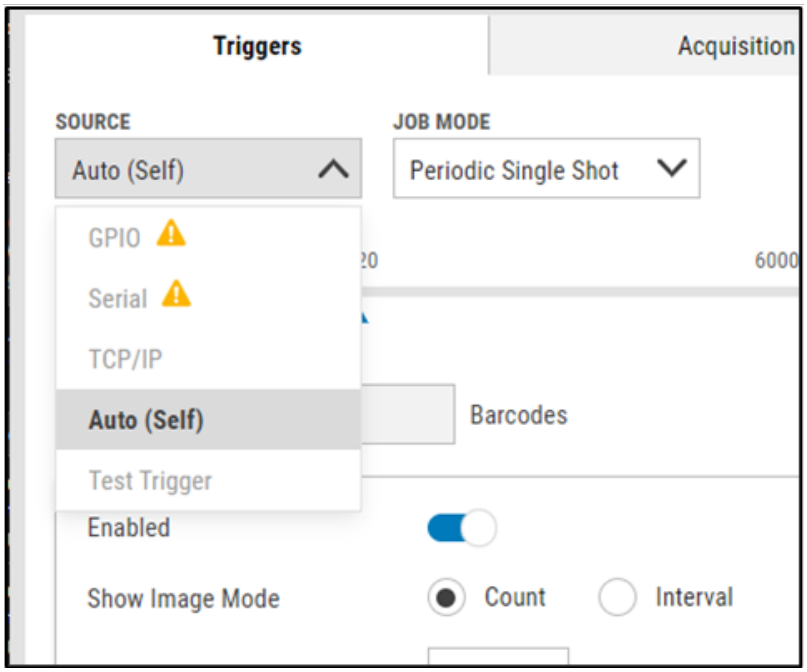


NOTE: SSI mode can only be enabled if HID results, CDC results, and CDC Control options are disabled.

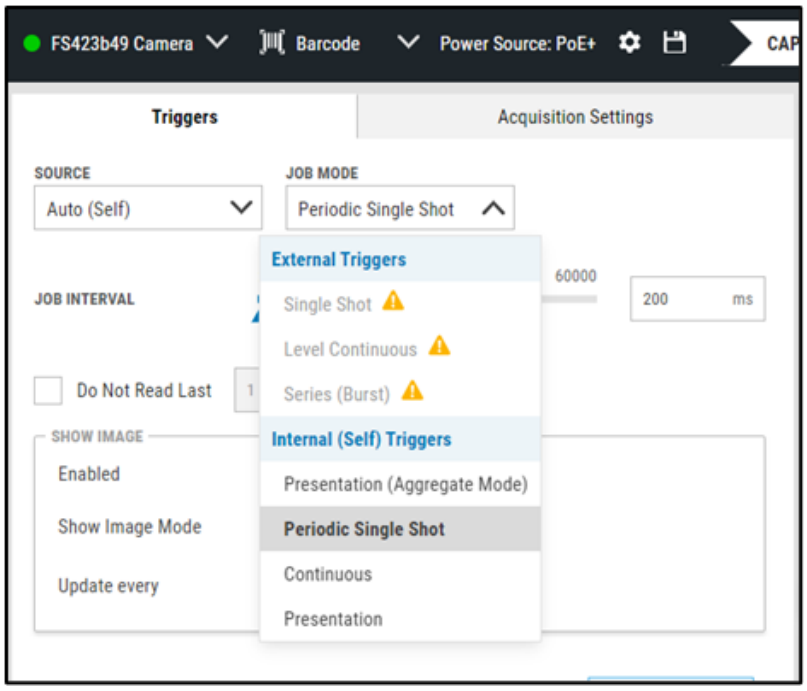
5. Navigate to job settings and configure. **Source** and **Job Mode** settings from the drop-down menu on the **Triggers** tab.

NOTE: Only auto modes are supported in job settings when SSI mode is enabled. The job starts when the device receives the enable command.

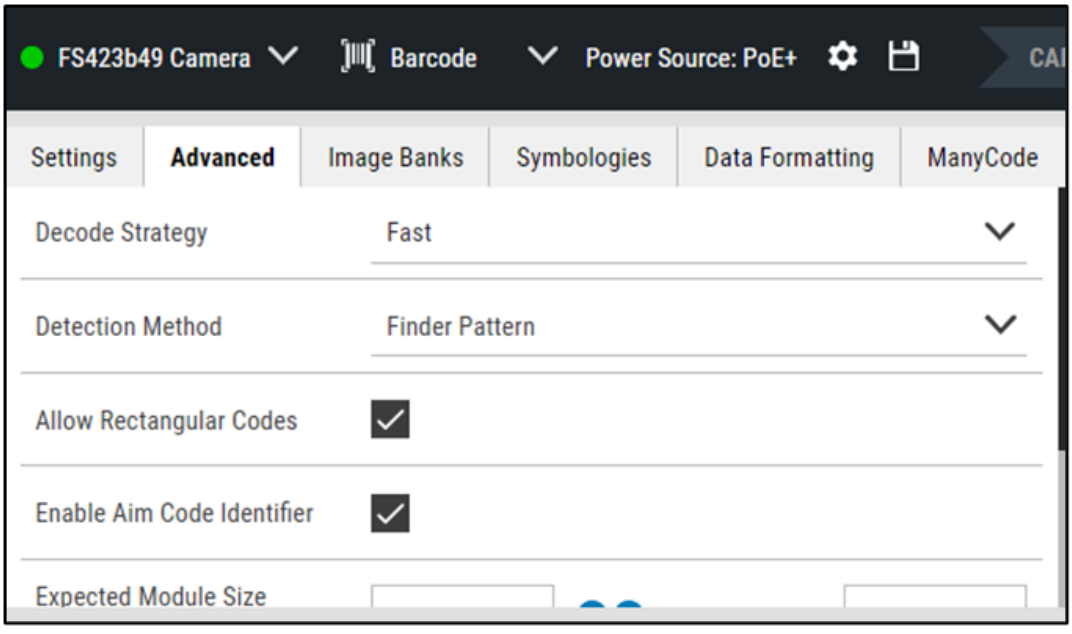
- Supported Trigger Sources: Auto/Self



- Supported Trigger Modes: Presentation, Continuous, and Periodic Single Shot



6. Ensure that **Aim Code Identifier** is enabled when creating a new job in SSI mode.



NOTE: An **Aim Code Identifier** is required for decoded data to be transmitted correctly. If it is not enabled, the first three characters may get removed from the barcode data sent to the device.

Zebra Automation Wedge Solutions

The following Zebra devices can be implemented with Fixed Industrial Scanners to form an end-to-end Automation Wedge solution.

NOTE: Supported Zebra Mobile Computers must have Android 11 or higher and support Zebra USB Scanner within the DataWedge Demo application.

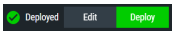
Table 20 Zebra Automation Wedge Solutions

Fixed Industrial Scanners	Mobile Computers	Cables	Docking Stations	Power Supplies
FS10, FS40 and FS42	TC73 Mobile Computer	DL Line Cord (CBL-DC-388A2-01)	TC73 Workstation Cradle (CRD-NTC7X-1SNWS-01	AC/DC Power Supply (PWR-BGA12V50W0WW)

Deploy Mode

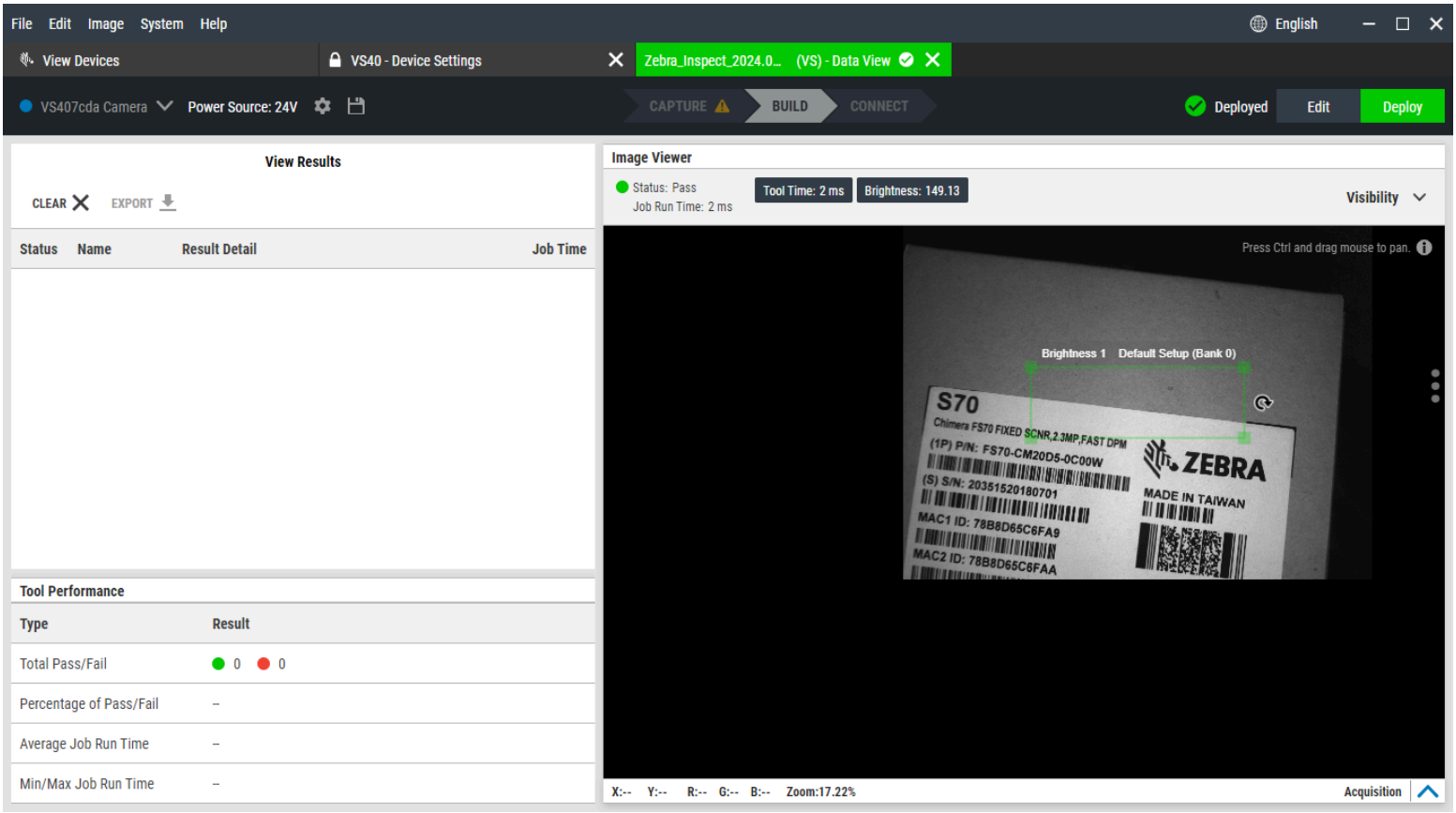
Use Deploy mode to view the job results and decode summary.

Click **Deploy** while in **Edit** mode 

to enter **Deploy** mode 

and view job results and decode summary for read count, total pass/fail, tool time, and quality information.

NOTE: The Filmstrip is not available while in Deploy mode.

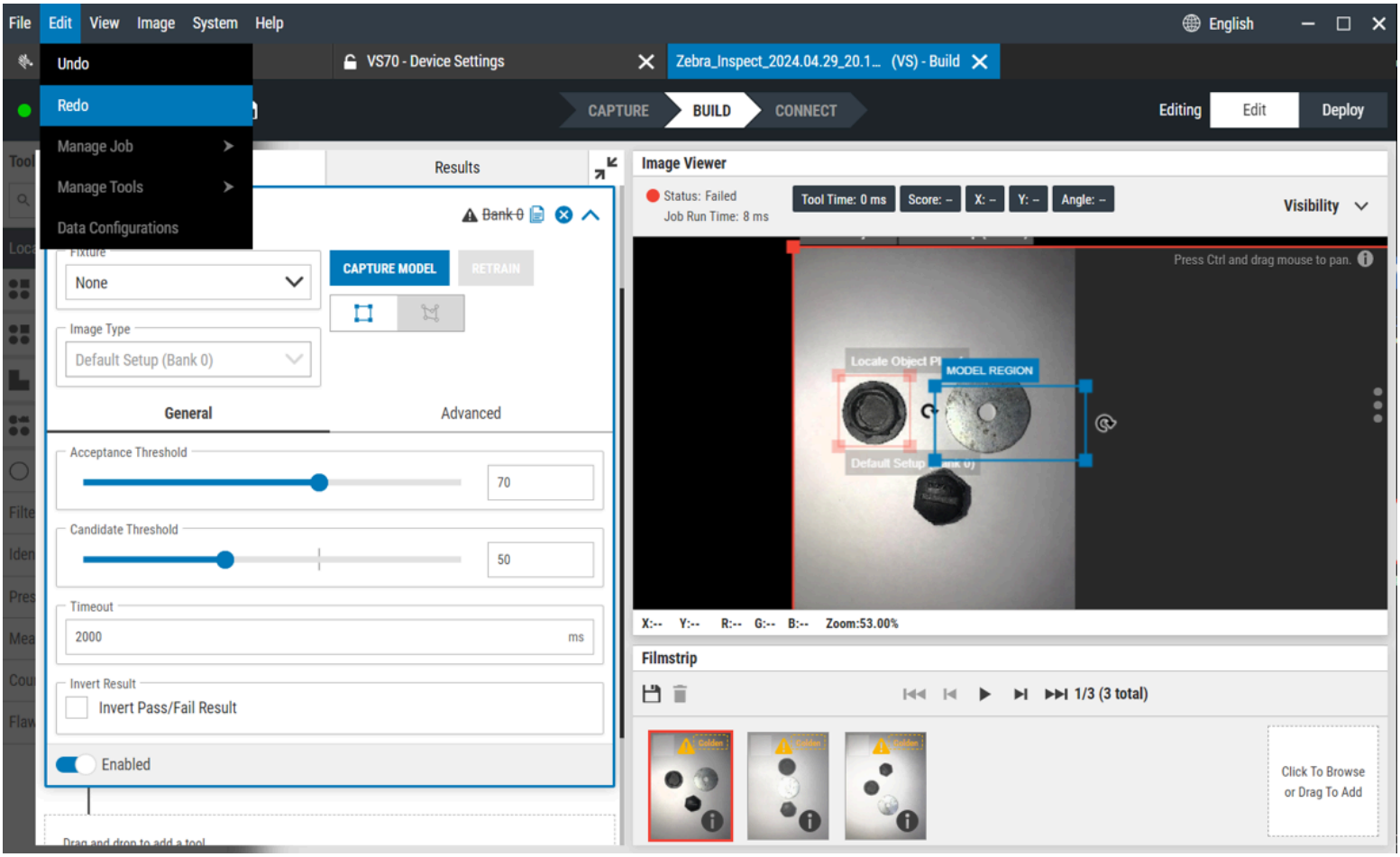


Using Undo/Redo

Use Undo or Redo to go back to the previous step in a job deployment or configuration.

Click **Edit** from the top menu and select **Undo** or **Redo** to revert back or forward while building a job.

Zebra Aurora Focus Software Overview



Connect

The Connect tab provides access to Industrial Ethernet, Output Formatting, Script Formatting, GPIO mapping, and Interface configuration.

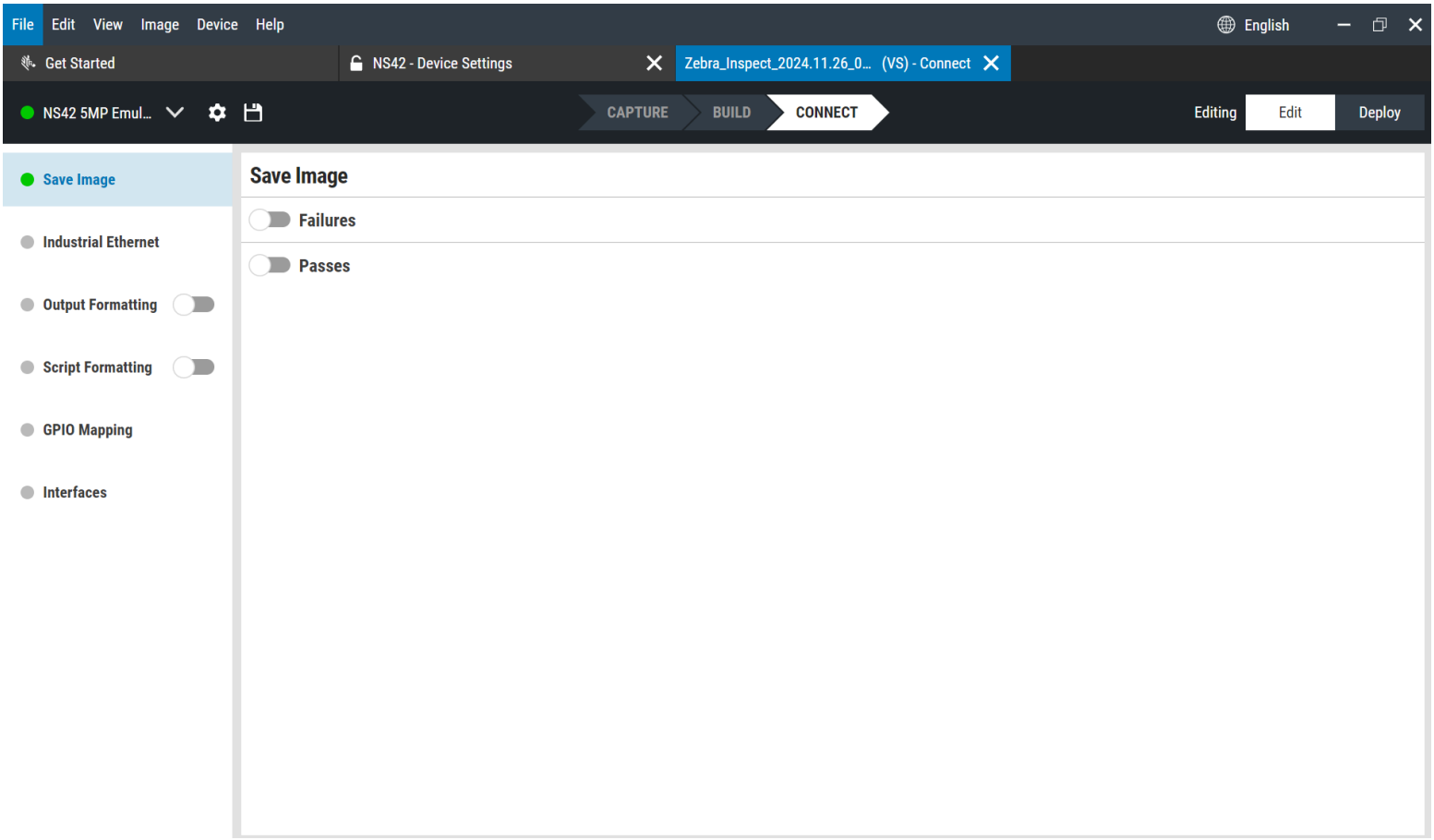


Table 21 Connect Settings

Setting	Description
Save Image	Enable save settings for Failures or Passes.
Industrial Ethernet	Configure User Control Data and Results data, add tools or jobs for configuration, and review the message sample.
Output Formatting	Enable Output Formatting to customize the formatting for all or specific results, including Serial, TCP/IP, and HID, by enabling filters or delimiters.
Script Formatting	Enable Script Formatting to open the JavaScript editor and create scripts for specific use cases.
GPIO Mapping	Configure GPIO Mapping by clicking **Edit Device Settings**.

Table 21 Connect Settings (Continued)

Setting	Description
Interfaces	Configure HID or RS-232 interfaces by clicking **Edit Device Settings**.

NOTE: Click ⚙ to configure advanced settings.

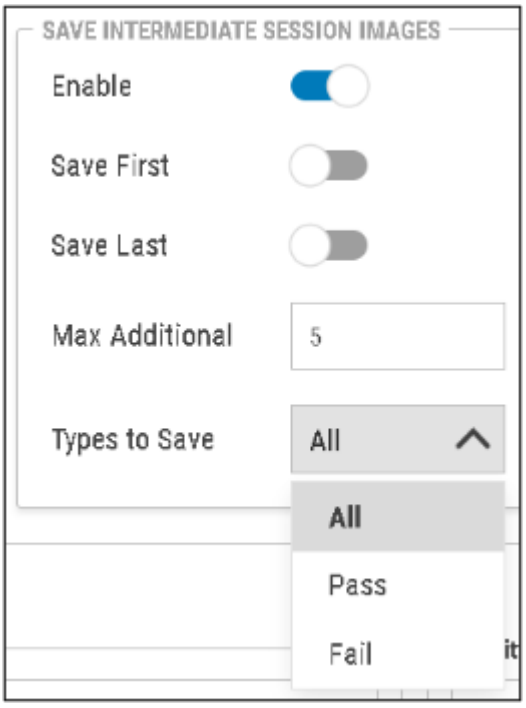
Image Saving

Use Image Saving to save reads, no reads, and intermediate session images.

1. In the **Sessions to Save** section, specify if the images should be saved for pass, failed, or both job runs.

NOTE: Use **Sessions to Save** to understand the outcome of a job. If multiple frames are processed for a single job.

NOTE: Enable **Save Immediate Session Images** for further analysis or troubleshooting. Use this option for trigger modes that acquire multiple images during a single inspection, such as continuous, presentation, and level continuous. Saving immediate session images may impact performance during high inspection rates. It should be implemented for troubleshooting by selecting only frames that failed to detect the code and/or limiting the maximum number of additional images to a low number.



NOTE:

2. Enable **Save Result JSON** to save JSON files for each saved inspection result.

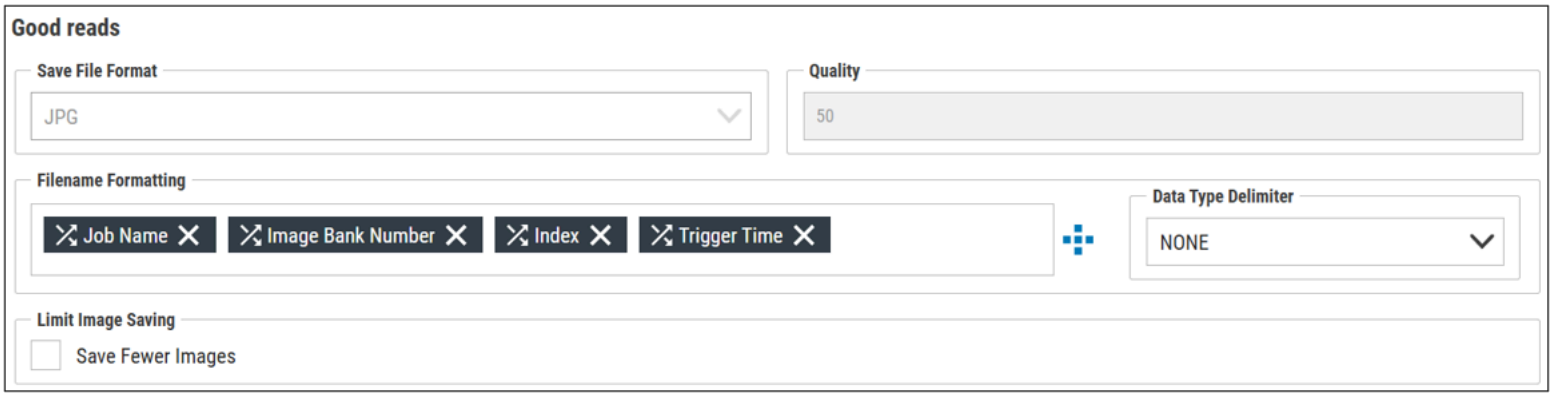
Configuring Saved Image Settings

Both No Reads and Good Reads use the same process to configure the file format for saved images.

1. Select the **Save File Format** from the dropdown.
2. Define the **Quality** of saved image filenames.

NOTE: Quality defines the compression level of saved JPEG images. Higher-quality numbers result in improved image clarity and larger file sizes. Lowering this value lowers image quality, resulting in a lower file size.

3. Determine the filename formatting convention by clicking ⊠ to remove or ✣ to add data types.
4. Select a **Data Type Delimiter** to add between each data type from the dropdown.
5. Enable **Limit Image Saving** to save fewer images and reduce memory and disk usage.



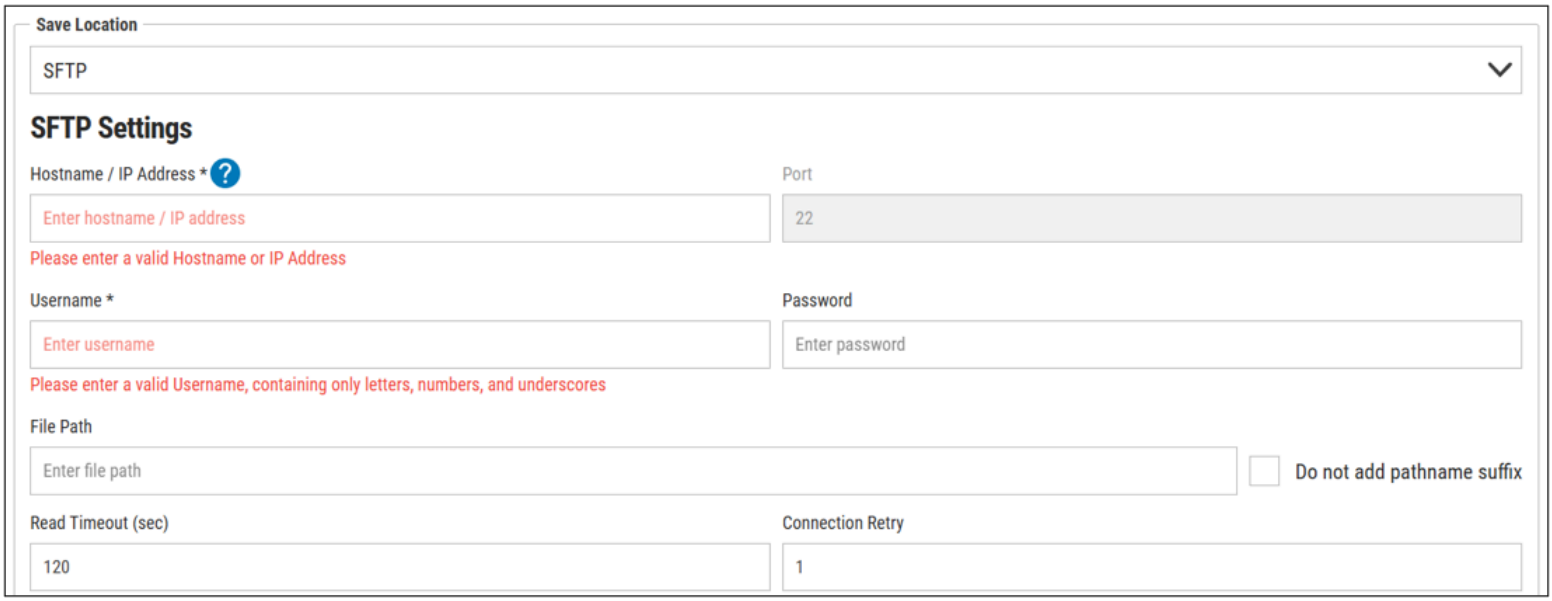
Add the following data types:

- Job Name - add the name of the current job.
- Image Bank Number - include the image bank number the job is in.
- Index - include the position of the job in the index.
- Image Time - include a timestamp of when the image was taken.
- Trigger Time - include a timestamp of when the image was triggered.
- Trigger Image Index - include the position of the triggered image in the index.
- Trigger ID - include the ID of the trigger.
- Barcode Read Count - include the number of barcode reads.
- Custom String - include a custom string (include the string when selecting this data type).

SFTP/FTP Settings and Best Practices

Configure SFTP/FTP settings including hostname, port, username, password, file path, read timeout and connection retry.

Select the Save Location (SFTP/FTP or on the device) for the images and populate the form fields with the required information.



NOTE: Storage on the server the device does not work as expected when special characters are present in the SFTP/FTP username.

The following restrictions on special characters in the SFTP/FTP username and FTP filename configuration are in place to ensure proper naming conventions are maintained.

- SFTP/FTP username:
  - Only upper and lower case letters, numbers, and underscores are permitted in the username.
- Linux filenames:
  - Avoid the following characters:
    - * ? | & > < .. : ; ~ !
- FTP filenames:
  - Avoid the following characters in the FTP path:
    - : * ? | < > " &

NOTE: Use JavaScript formatting to override the default file name formatting for saved images and manage the output data. For additional information on developing custom scripts using the JavaScript editor, refer to the Aurora Focus JavaScript Developer's Guide.

Byte Swap Configuration

Configure a byte swap in the Ethernet/IP protocol to toggle byte order and ensure the connected PLC correctly interprets string data.

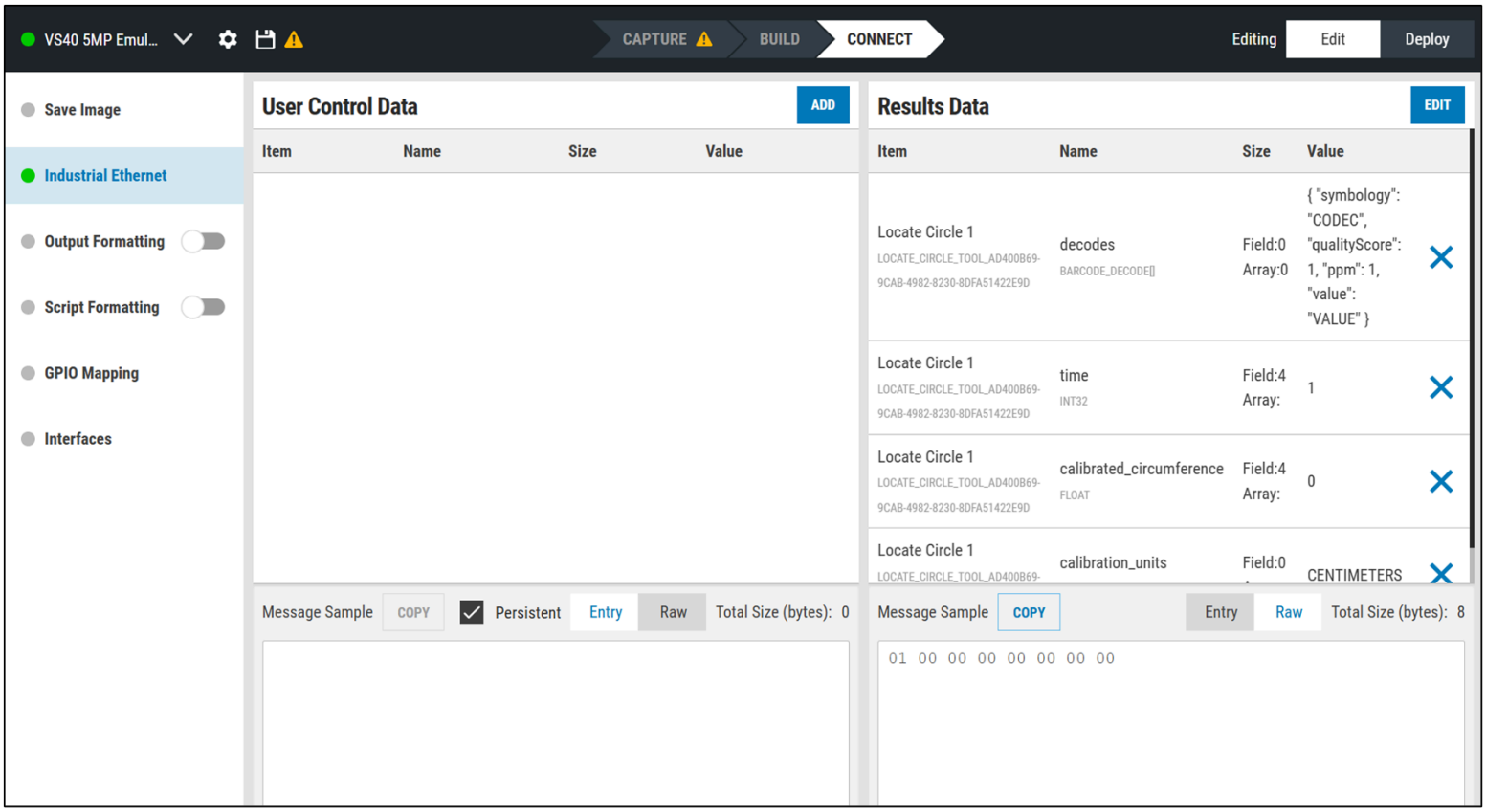
NOTE: The default setting maintains the camera's native byte order (no swap), ensuring backward compatibility.

NOTE: In Aurora Focus, the Industrial Ethernet boolean fields have a fixed size of 1 byte. This can cause misalignment of received data on the PLC side when a boolean field follows a float or an integer field, both of which are 4 bytes in length. In order to avoid misalignment of data, boolean fields should be added as the last field in the Industrial Ethernet data configuration.

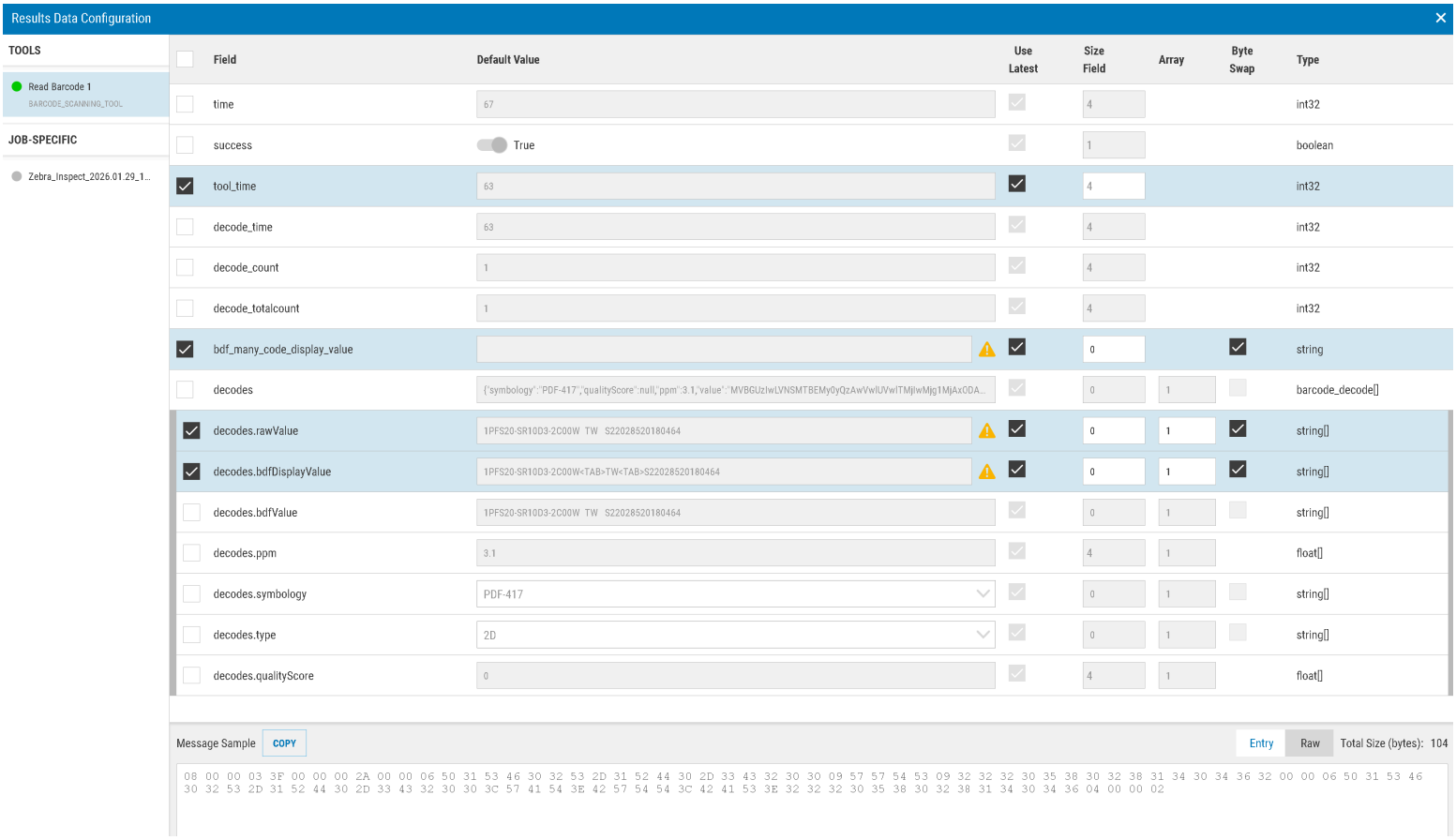
1. Configure the PLC Protocol by selecting Ethernet/IP or Modbus TCP in Device Settings.

NOTE: A manual reboot is required for PLC protocol changes to take effect.

2. Go to the **Connect** tab and click **Edit** in the **Results Data** to edit the Results Data Configuration.



3. Select two fields or more to **Byte Swap**.



4. Click **Submit**.

Output Formatting

Use Output Formatting to customize data output from identification and Machine Vision Tools. Access Output Formatting using the **Connect** tab to adjust delimiter settings for each tool or job.

1. Navigate to the **Connect** tab and click **Output Formatting**.

NOTE: Standard identification tools (Read Barcode, Read DPM, Read DPM & Barcode, and Datacode) will have their Decodes.formattedOutputValue output. If there is more than one standard identification tool, the additional tool's Decodes.formattedOutputValue will be appended to the existing result data. The output result will be in the order of the tools on the flowbuilder. For example, barcodeTool1barcodeTool2barcodeTool3

NOTE: Machine Vision Tools will add the overall job Success field (Pass/Fail) to the result output. If a standard identification tool is in the job, the job Success field is added to the front of the output result with a comma separator between the job success and barcode data.

For MV tools only: Pass

For MV tools and standard identification tools: Pass,barcodeTool1barcodeTool2barcodeTool3

2. Slide the toggle to the right to enable **Output Formatting** and customize the output result format.

3. Click the specified tool under **Tools** to add a **Results Field** or **Delimeter** to a **Prefix, Data,** or **Suffix** category.
4. Click the specified **Job** to add a **Results Field** or **Delimeter** to a **Prefix, Data,** or **Suffix** category.
5. Select a **Data Type Delimeter** from the dropdown menu.
6. Click **Copy** to copy the **Message Sample** to clipboard.

FTP File Saving

1. Download babyftp (free FTP solution): pablosoftwaresolutions.com/html/baby_ftp_server.html.
2. Navigate to Settings and define the home directory for files to be saved to.

NOTE: Babyftp does not have an option for credentialed access (SFTP).

3. Use the following credentials:

- Username = anonymous
- Password = <blank>

Determine the IP address of the host computer where the FTP resides.

1. In Aurora Focus, navigate to the **Connect** chevron, then **Save Image**.
2. Select **FTP** under **Save Location**.
3. Click the **Gear** Icon
4. Enter the IP address of the host PC.
5. Use the following credentials:
   - Username = anonymous
   - Password = <blank>
6. To save to a specific subdirectory on the FTP server, enter the following format:

```
/xyz/ (this will write to the folder identified below)
```

```
<FTP home directory>/xyz/
```

```
C:\Users\RMQ783\Desktop\babftp\xyz\
```

Save.

NOTE: Include a tilde (~) when providing the file path, for example, ~/myFilePath/

Deploy a job to observe the Pass/Fail images in the expected folder.

NOTE: For example, failed images are saved in the following directory after a Failed job run: Desktop\babyftp\VS407a8e\result\asf\fail. Each job run generates a .jpg and a JSON file.

FTP Naming

The FTP file name is generated from different fields.

For example:

result_FtpJob_bank_0_2023-06-26T03-04-13.004479+00-00

- Parameter File Name Prefix as configured in the **Connect** tab.
- JobName
- bank_0 that identifies the first image acquired by ImagePerfect+
- The image timestamp with time representation up to microseconds.

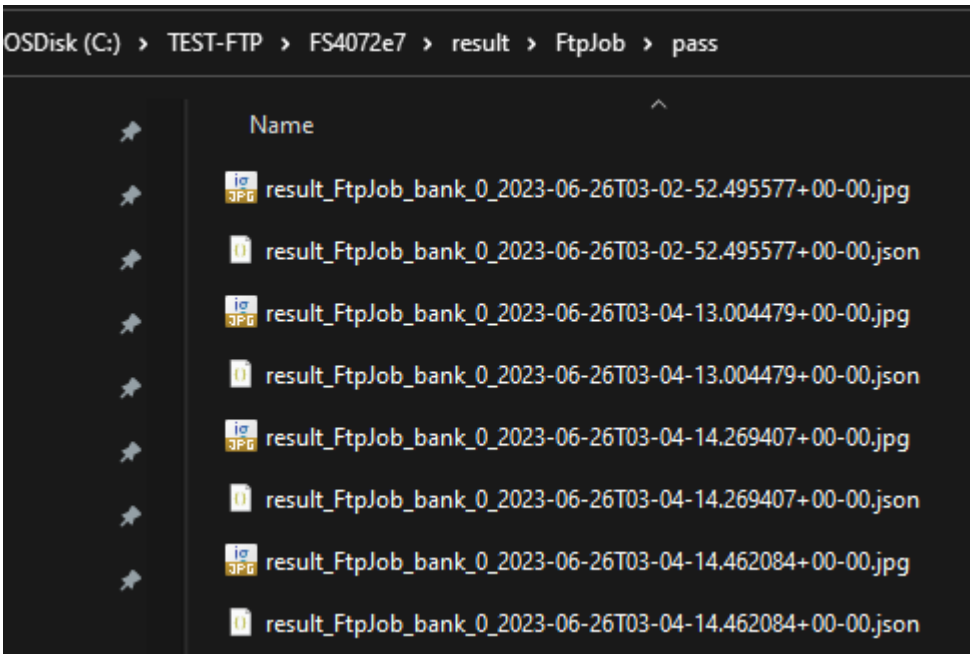
It is recommended to align the image name with the setting name in **Acquisition Settings**.

FTP Folder Structure

The default FTP settings can generate a complex folder structure.

The file path is formed by:

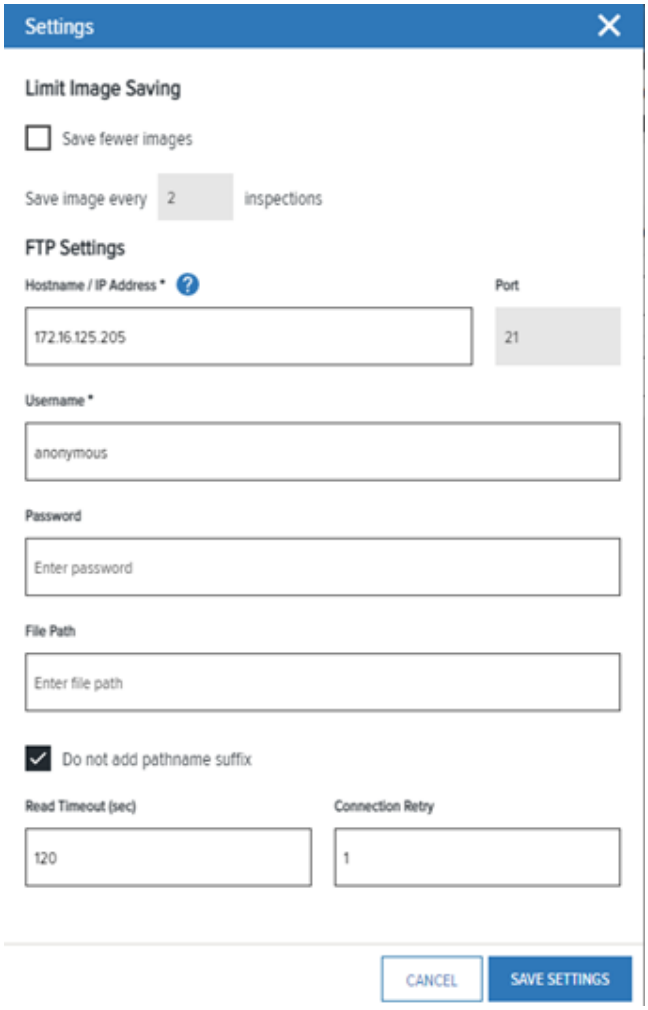
- Host Name (as configured in **Communication > General > Host Name**)
- result
- JobName
- Pass or Fail (based on Good Read and No Read events)



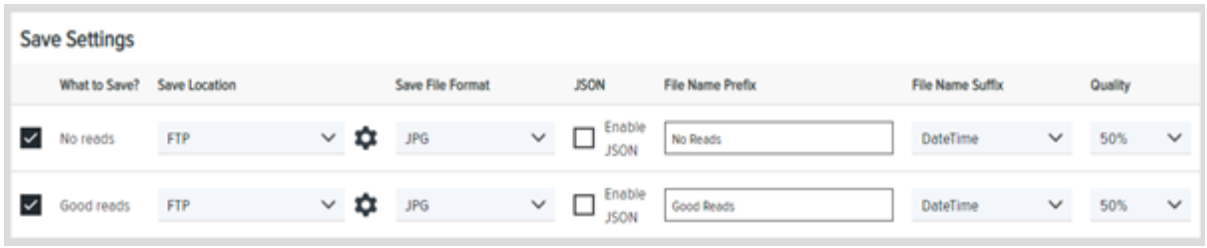
Applying a Simplified FTP Folder Structure

If the default folder structure is not necessary, disable the default folder structure and apply a simplified structure.

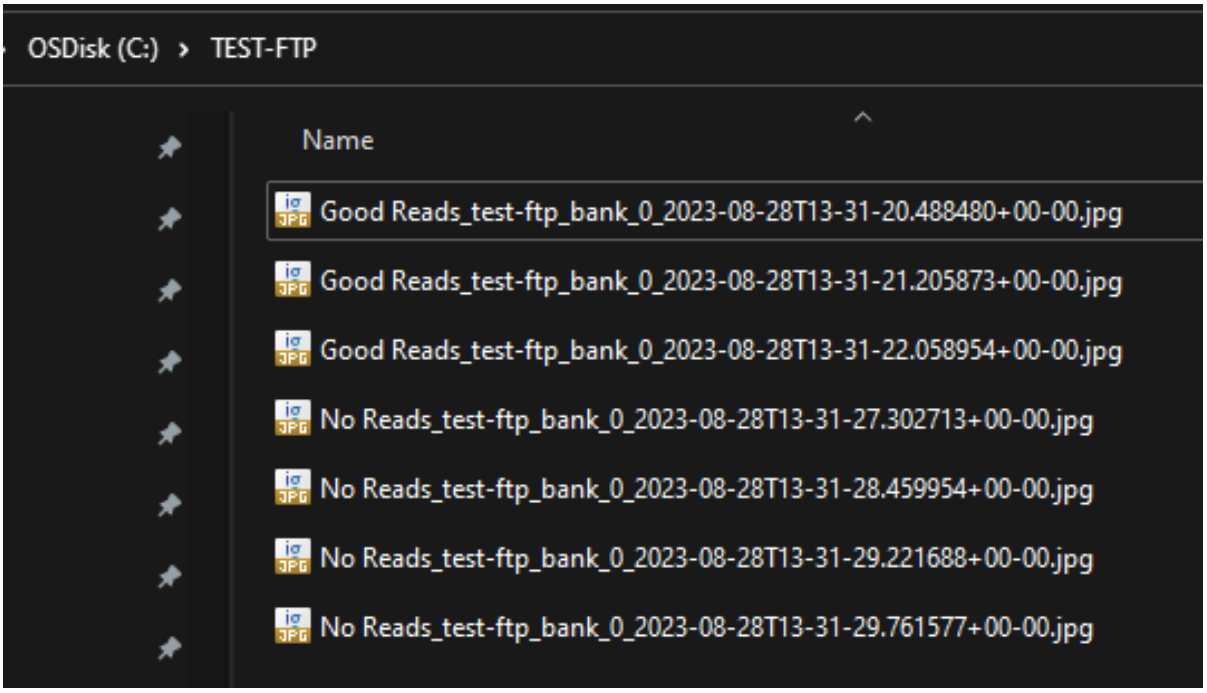
1. Enable **Do not add pathname suffix** in the **Settings** dialog for the FTP connection.



2. To save pass or fail images only and avoid the JSON file, use the **Connect** chevron and deselect the **Enable JSON** checkbox.



The following figure displays the resulting folder content. Use File Name Prefix to separate Good Reads from No Reads without using the folder structure.

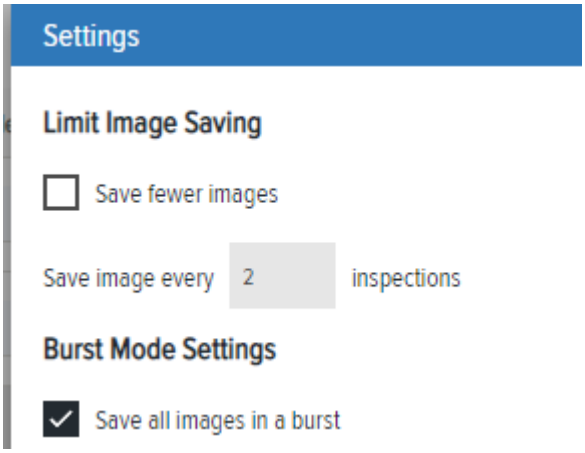


Saving a Series of Images in Burst Mode

Images collected using Series or Burst mode are saved to the FTP server.

Get the **Burst Count** and disable Stop after successful inspection to retrieve the same number of images each time.

In the FTP configuration dialog, configure **Burst Mode Settings**.

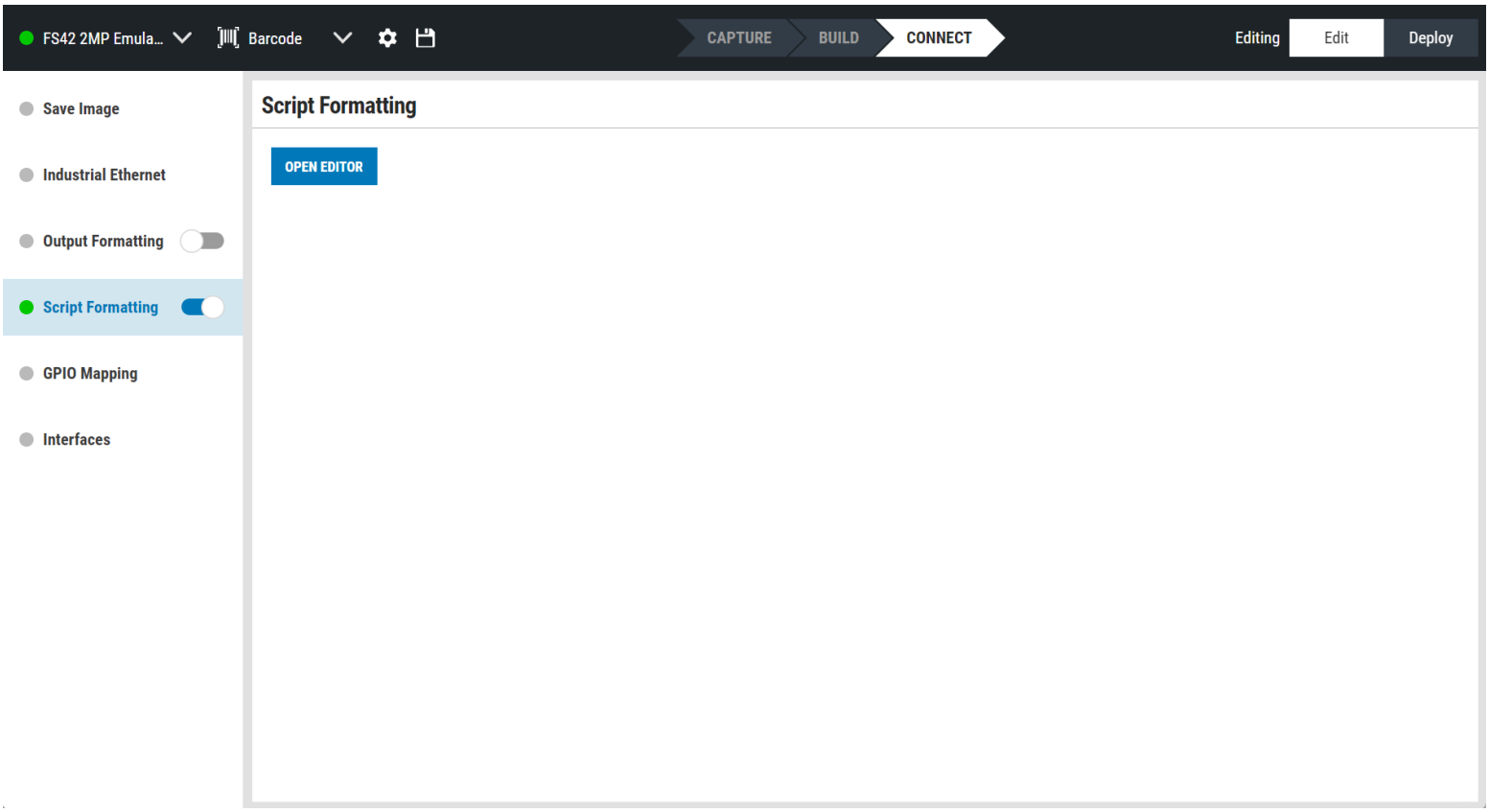


JavaScript Formatting

Access **Script Formatting** from the **Connect** tab.

NOTE: JavaScript Formatting is supported on Aurora Focus version 9 and above. For additional information on developing custom scripts using the JavaScript editor, refer to the Aurora Focus JavaScript Developer's Guide.

Enable **Script Formatting** from the menu and click **Open Editor** to open a new editor window.



Using the JavaScript Editor

The Aurora Focus JavaScript Editor consists of menu bar, control bar, code editor and console.

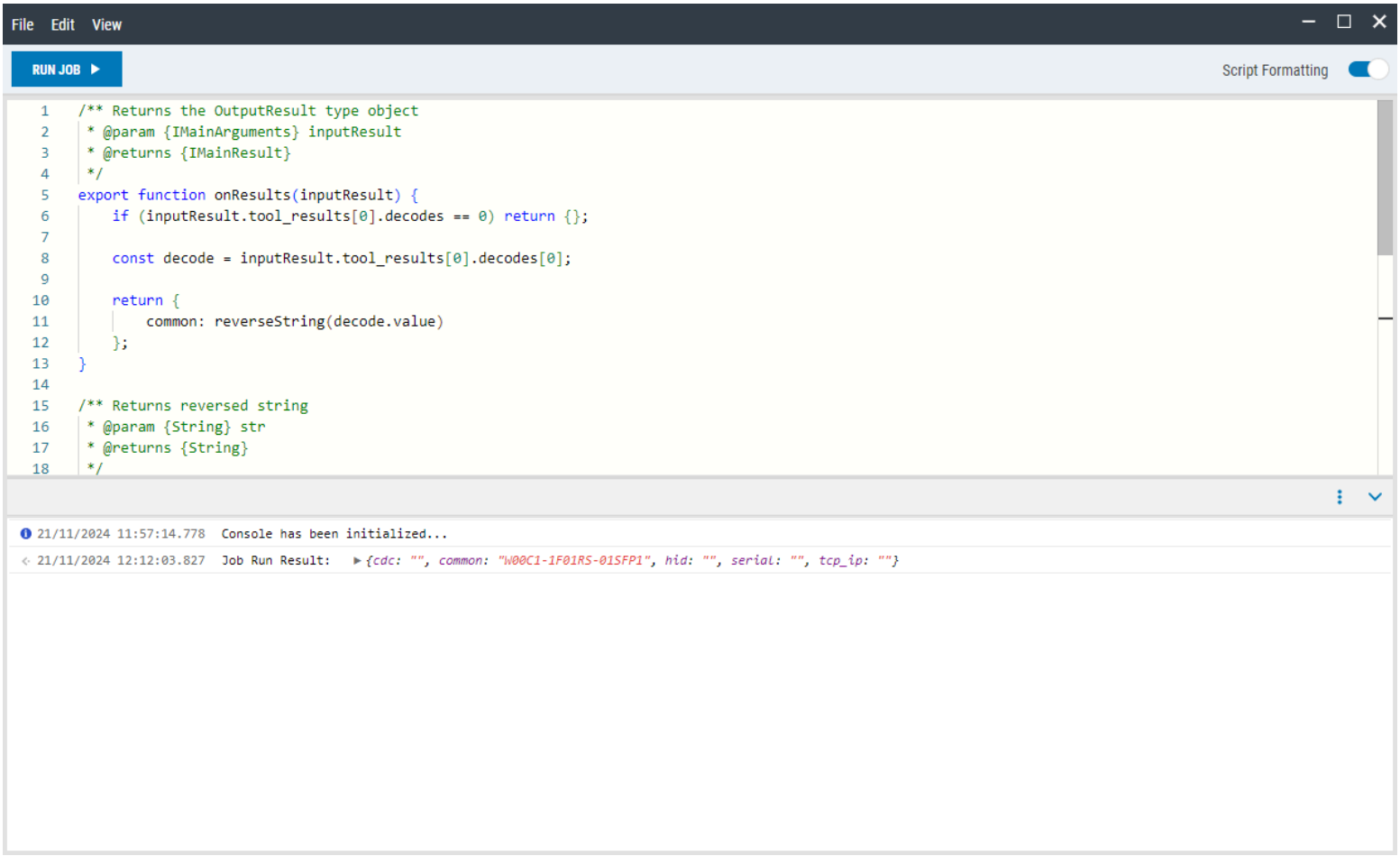
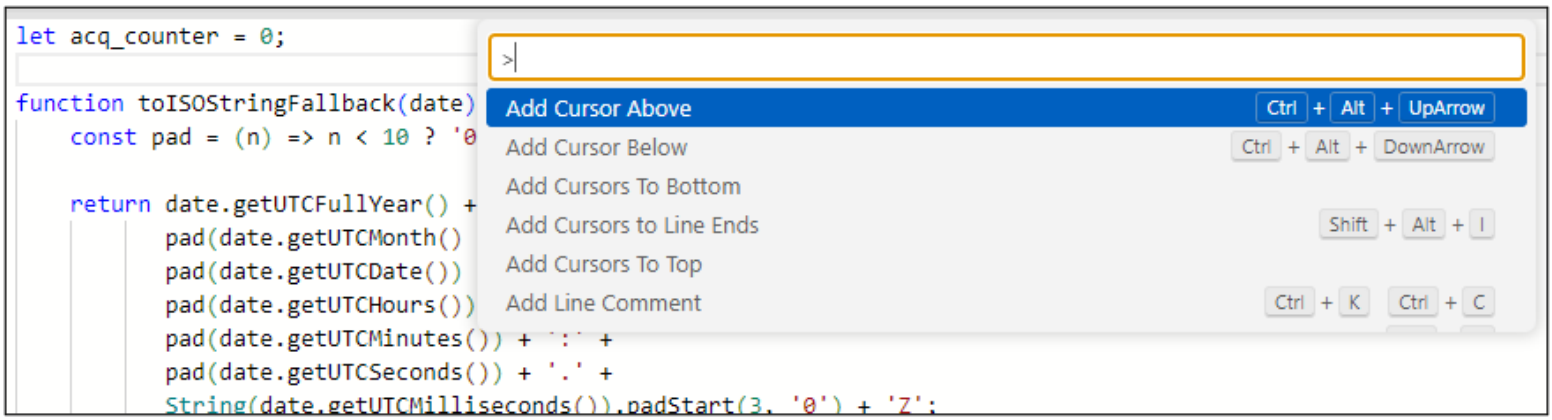


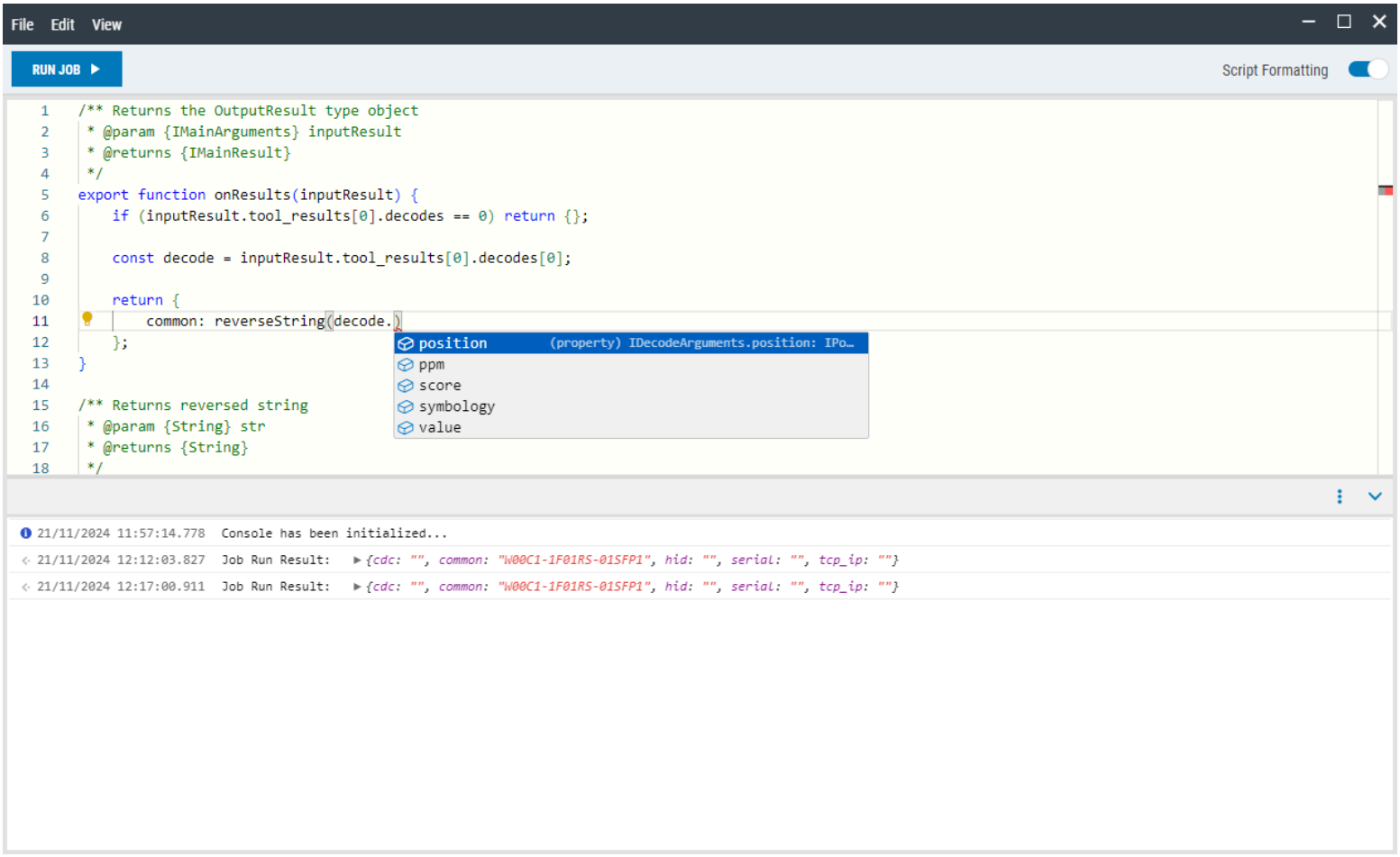
Table 22 JavaScript Editor UI Elements

	UI Element	Description
	Top Navigation Menu File Edit View	The top navigation menu provides access to **File**, **Edit**, and **View** functions for the application. • Open an existing file with JavaScript code and save JavaScript to an external file from the **File** menu. JavaScript by default is saved in job file when saving a whole job or deploying a job to the device. • Use templates and snippets to quickly develop new JavaScript code that is specific to your use case from the **Edit** menu. • Adjust editor font size from the **View** menu.
	Run Job RUN JOB ▶	Click **Run Job** on the control bar to validate test code in the editor by running a job on an image in the Filmstrip in Aurora Focus. You can also trigger a job by pressing F5 while in the editor.
	Script Formatting Script Formatting	Disable JavaScript formatting by using the **Script Formatting** toggle in top-right corner of the editor window.

Press F1 while inside the editor for a list of all functionalities.



Use IntelliSense for script recommendations to enable code-completion. When a function has JSDoc documentation, the editor assumes its variable types and provides a recommendation for object properties. IntelliSense is also useful for standard JavaScript interfaces like JSON or Date. It is recommended to keep JSDoc comments for the `onResult` function. This is necessary to receive recommendations for input and output objects.



The console below the code editor displays logs. The log icon and color change depending on the log category. For example, warnings are on a yellow background, errors and result object.

Table 23 Console Log UI Elements

UI Element	Description
	Save the console logs to a file or clear the console.
	Minimize the console.

NOTE: Only the latest 100 entries display on the console, older ones are automatically deleted.

The code editor is powered by Monaco Editor that uses Visual Studio Code and integrates functionalities such as formatting, undo/redo, renaming symbols and find & replace. The following code snippet is a basic script that copies the first detected barcode from the first tool to the output.

```
/** Returns the OutputResult type object
* @param {InputResult} inputResult
* @returns {OutputResult}
*/
export function onResults(inputResult) {
return {
 common: inputResult.tool_results[0].decodes[0].value

};
}
```

Connectivity Gateway Solutions

The Zebra Connectivity Gateway provides asynchronous passthrough and synchronized leader-follower solutions for use cases requiring high-speed scan tunnels, sorting facilities, and multi-point or multi-side barcode scanning for parcels and boxes.

When assembling the Connectivity Gateway with the Gateway License, consider the following device characteristics before determining which device acts as a leader or follower.

- Fixed Scanning and Gateway devices enable USB HID mode as output. However, GS20 devices do not natively support a USB port.
- Fixed Scanning and Gateway devices support up to 9 total GPIO. GS20 devices have 4 total GPIO.
- It is recommended that no more than four followers be implemented in a given Connectivity Gateway solution. This implies that the Fixed Scanning and Gateway devices may act as follower and leader devices simultaneously. If this is implemented in your system, expect a slight degradation in read rate performance in the device acting in both modes.

Asynchronous Passthrough

In asynchronous leader-follower use cases, multiple devices send result data to a single leader device. The leader passes the data through as a single point of contact to the host.

Examples:

- High-speed scan tunnels.
- Humans present barcodes for scanning in a sorting facility.

Fixed Scanner Input:

- Generic
- TCPIP (Followers)

Fixed Scanner Output to Host:

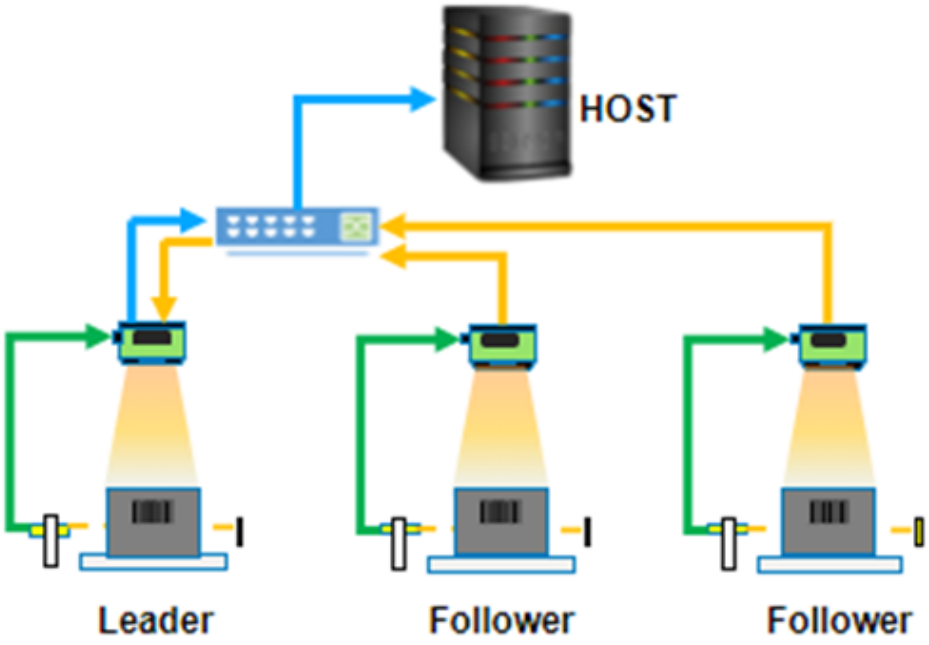
- TCPIP
- Industrial Communication
- Serial
- USB HID

Functionalities:

- Enable simple passthrough to send result data without changes.

- Enable advanced passthrough to change or add input text before the output.

Figure 1 Asynchronous Passthrough



Synchronized Leader Follower

In synchronized leader-follower use cases, the leader receives a trigger, activates the follower devices, receives the data from the followers, and sends the results to the host.

Examples:

- Multi-sided reading of parcels.
- Multi-point reading of barcodes on a tire rim (overhead view only).

Leader Input:

- Generic:
  - GPIO Trigger
- From Follower:
  - Results:

Leader Output:

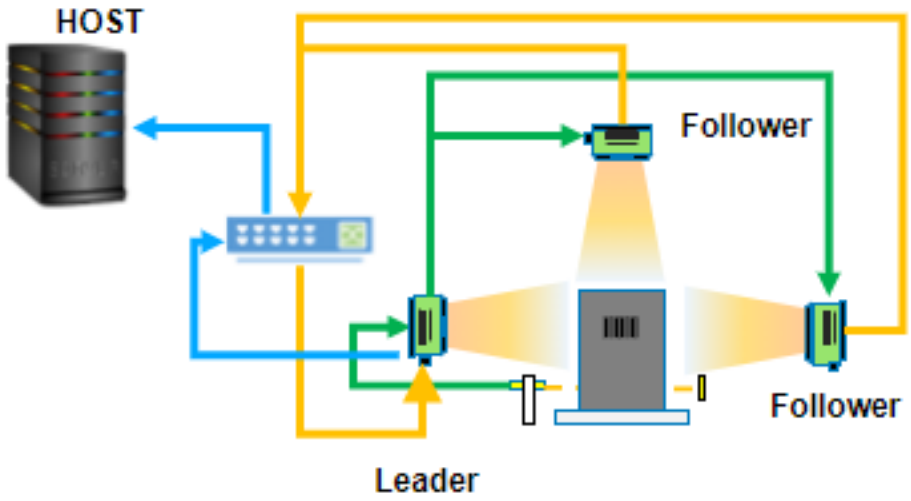
- To Host:
  - TCPIP
  - Industrial Communication
  - Serial
  - USB HID

- To Follower:
  - Trigger
- To Machine/PLC:
  - Total Result String
  - Total Pass/Fail GPIO

Functionalities:

- Enable synchronous triggers from leader to follower devices.
- Recieve results from follower devices and apply pass/fail criteria.
- Send result data to the host, GPIO, or PLC.

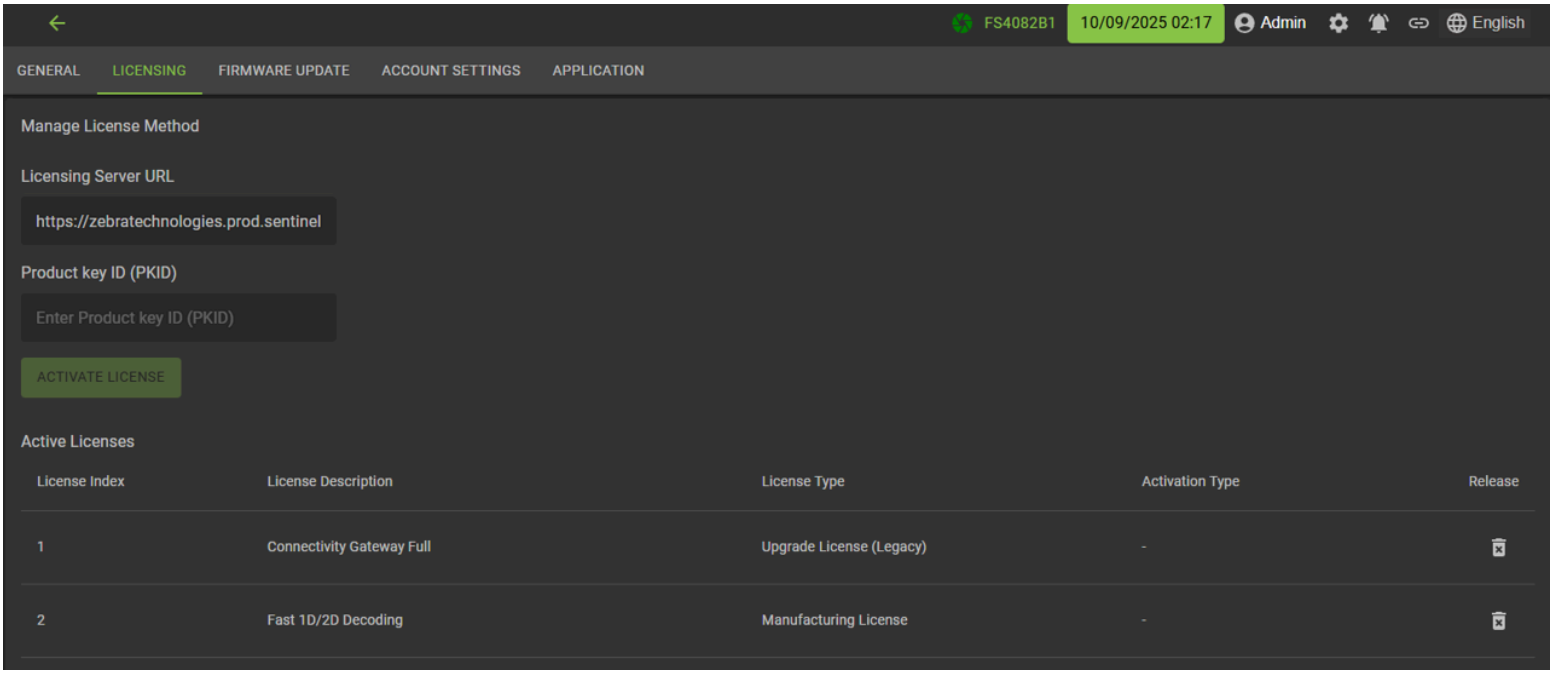
Figure 2 Synchronized Leader-Follower



Gateway Licensing

Manage licenses for the Connectivity Gateway solution using the Licensing tab in the Web HMI.

Click the **Licensing** tab to activate a new license and view active licenses.



NOTE: GS20 devices include a license by default.

Compatible PLC Devices

The Connectivity Gateway supports communication to and from the following PLC models.

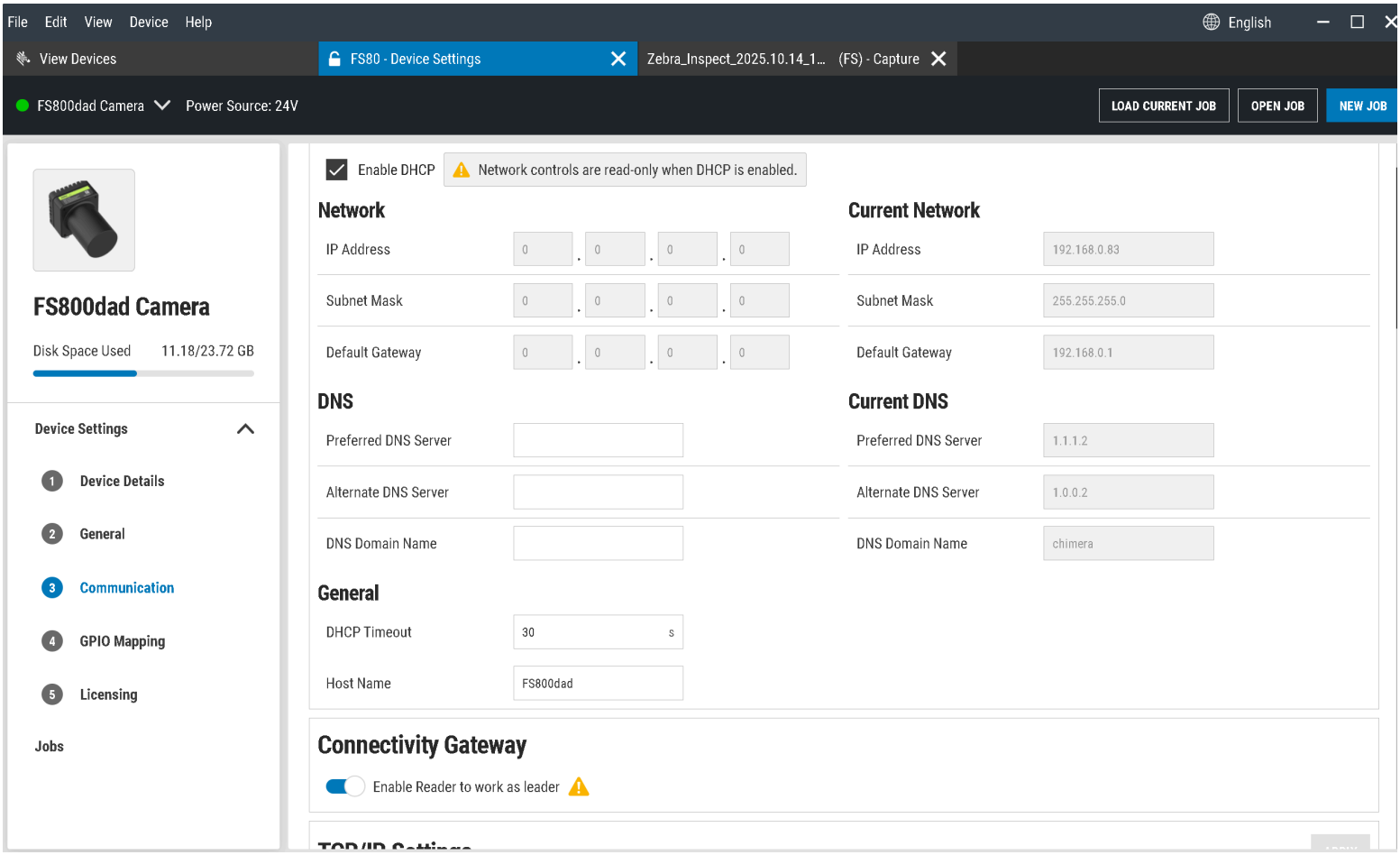
PLC Manufacturer	Protocols
Siemens	S7 TCP/IP
	Logo!
	S7-200
	S7-300
	S7-400
	S7-1200
	S7-300 (ERPC)
	ET 200 Pro
	S7-1500
	SINUMERIK 840D
	PCS 7
Rockwell	ControlLogix CPU
	CompactLogix CPU
	Micro800 CPU
Mitsubishi	QCPU (Built-in Ethernet)
	LCPU (Built-in Ethernet)

PLC Manufacturer	Protocols
	RCPU (Built-in Ethernet)
	FX3CPU
	FX5CPU
	GOT
	ACPU
Omron	CS1 CPU
	CJ1 CPU
	CV1 CPU
	NJ CPU
	CP1 CPU
	NX CPU
	CJ2 CPU
	NE1S CPU
Modbus	TCP/IP

Enabling a Device as the Leader Module

Configure a device as the leader module in a connectivity gateway.

1. Go to **Device Details**
2. Select **Communication** from the menu.



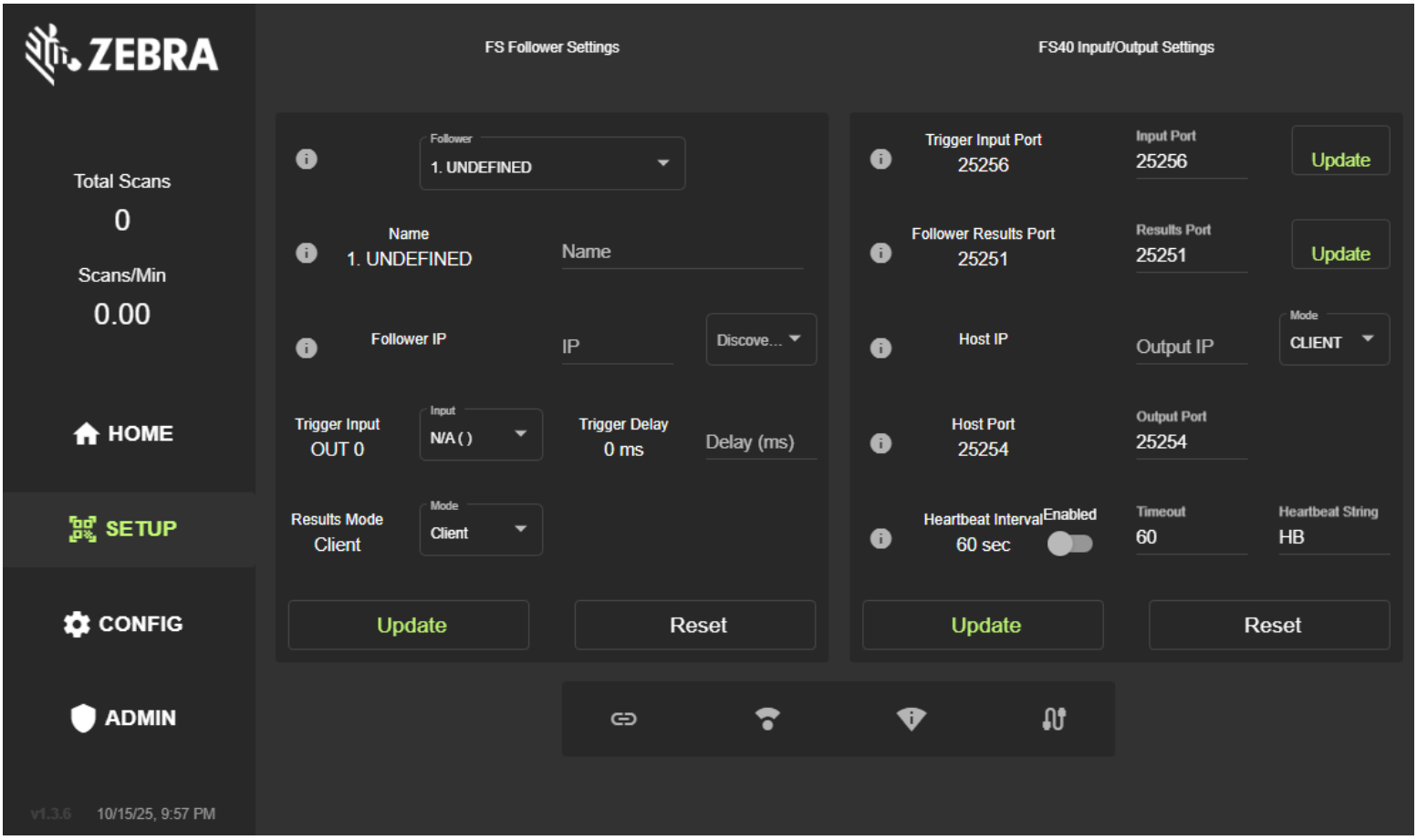
3. Click **Enable Edit Mode**
4. Slide the toggle to enable the reader to work as a leader.

NOTE: This change requires a reboot.

Setting Up Follower Devices

Set up follower devices for the Connectivity Gateway using the Web HMI.

1. Navigate to the **Setup** screen using the left menu on the **Home** screen.



- If you are in triggered mode and intend to update the port that the device is monitoring for an incoming trigger string, enter the required port into the Trigger Input field (0 to 65535) and click **Update**.
- To configure the IP and Port address of the host, enter the required IP address and port into the Output IP and Output Result fields. Click **Update** for the changes to update the field values.
- To set up the TCP port to act as a Server, select this option from the **Mode** menu and click the icon above the menu to edit the server settings.

2. Click **Update** to enable the changes and update the field values. Select **Server** to use the TCP port as a server.

NOTE: When the Leader to Host port acts as a Server, a timeout occurs after five seconds of inactivity. In some cases, you may need to set up a heartbeat to keep the connection alive by sending a string (for example, HB) from the host machine to the device IP address and the specified server port with a frequency higher than every five seconds.

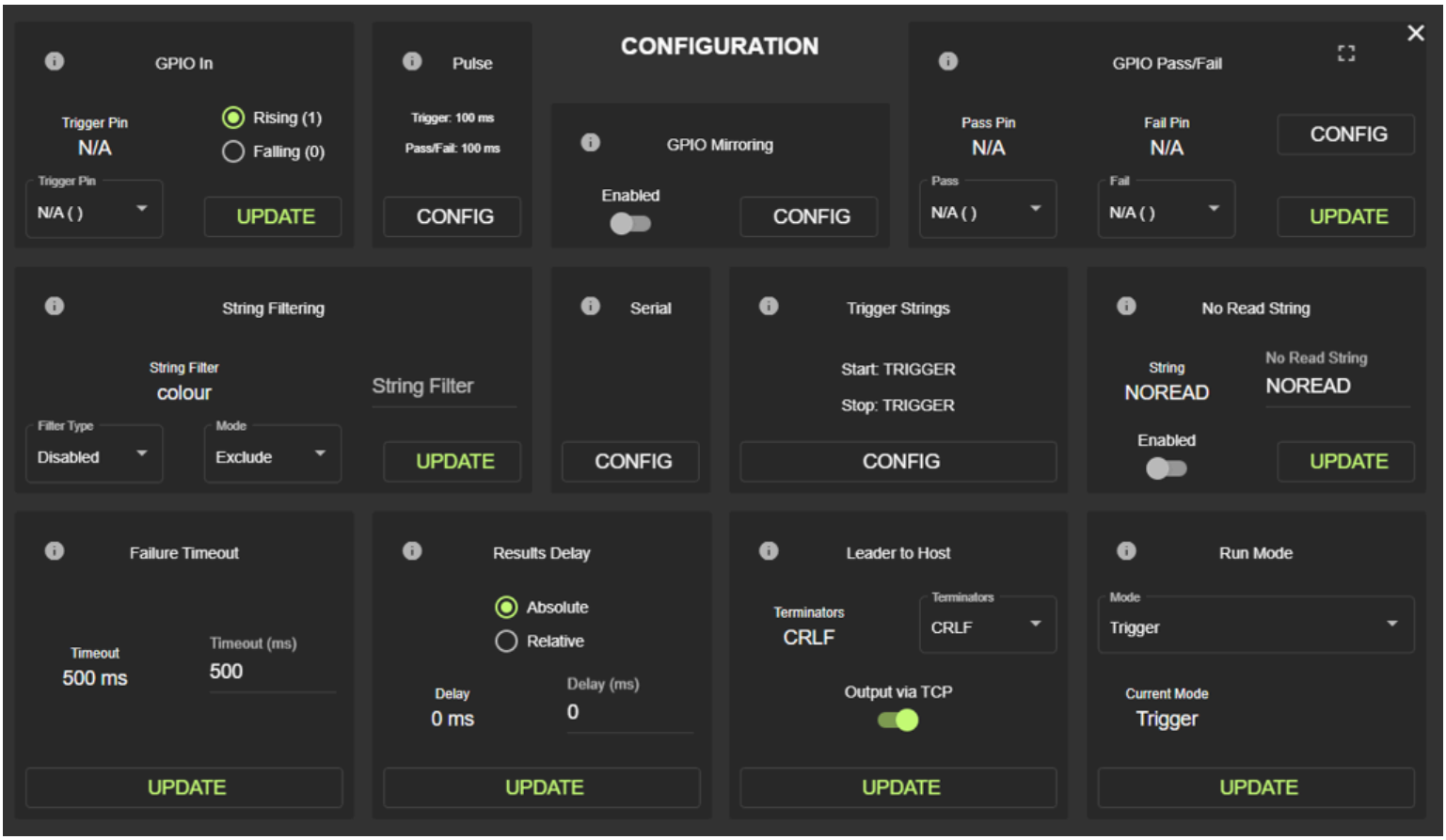
3. Once the form fields are populated, click **Commit Changes**.
4. Observe the message and wait for at least two minutes before sending the next trigger to the Connectivity Gateway.

NOTE: Inactivity Timeout is the time the device waits for the next TCP message before the connection is lost. This setting is only applicable when the leader device operates in Server mode. The default setting of one day is helpful for stable network connections. However, lower

timeout options should be used along with a heartbeat from the host to recognize dropped connections quickly on unstable networks. If your network is unstable, set up a heartbeat to maintain the connection by sending a string (for example, HB) from the host machine to the leader's device IP address and specifying the server port with a frequency higher than the inactivity timeout.

Configuring Follower Devices

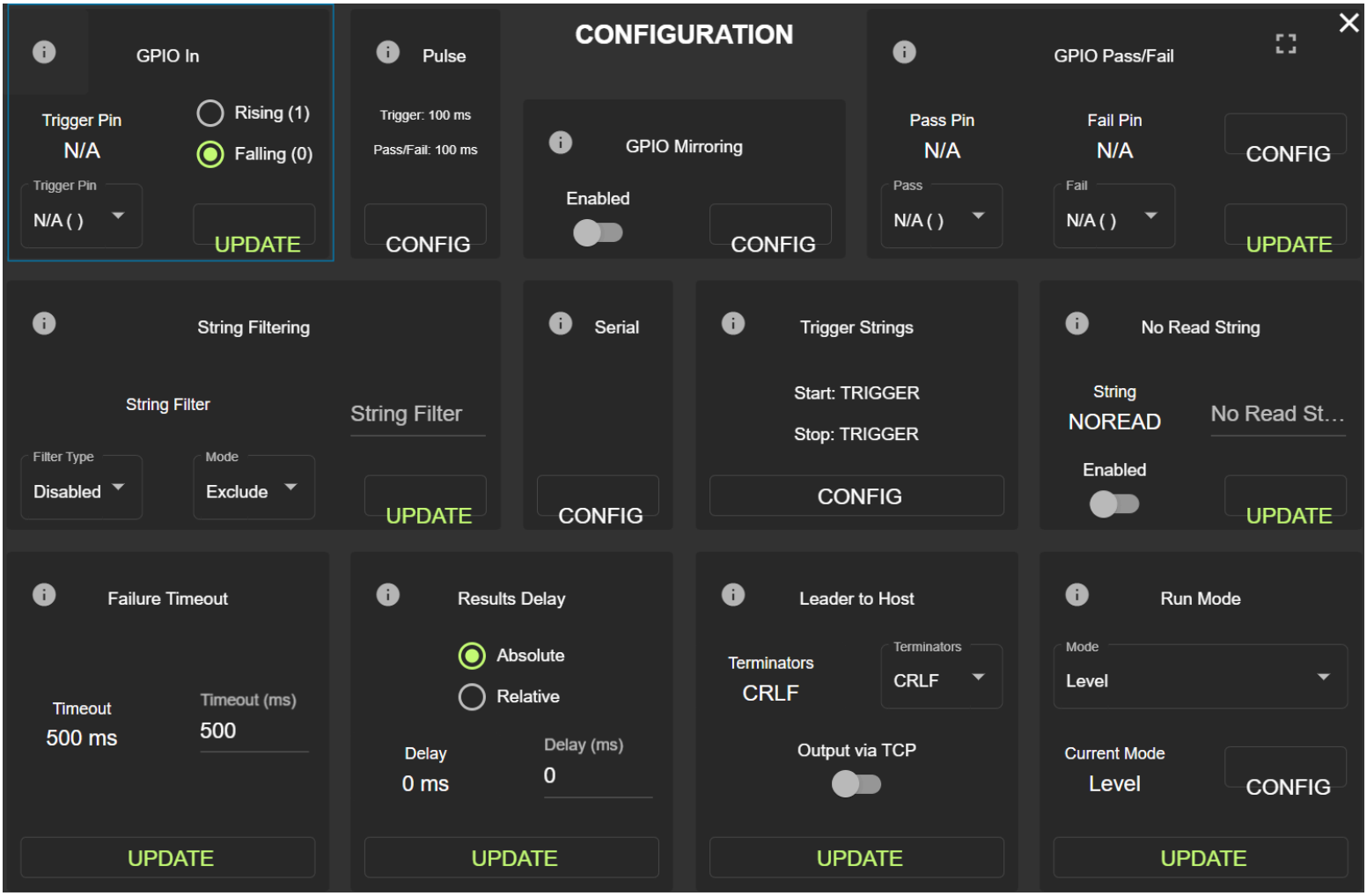
Click **Config** on the **Home** screen to configure GPIO, Filtering, Triggering, Timeout, and Run Mode settings.



GPIO Triggering

Define a pulse width for the signal to trigger a device over the input of your choice.

1. Navigate to the **Configuration** screen using the left menu on the **Home** screen.
2. In the top left GPIO section, select the desired trigger pin (either In 0 or In 1) and click **Update**.



3. Click **Config** to edit the Trigger Pulse and Pass/Fail Pulse fields.

4. Enter the desired values and click **Update**.
   - Trigger Pulse: specifies the pulse width for the signal that triggers the follower devices.
   - Pass/Fail Pulse: specifies the pulse width for the pass/fail result signal.

NOTE: If you configure the same output for GPIO Pass/Fail and as an input for the follower device (for example, trigger input), the trigger input for the follower device takes precedence.

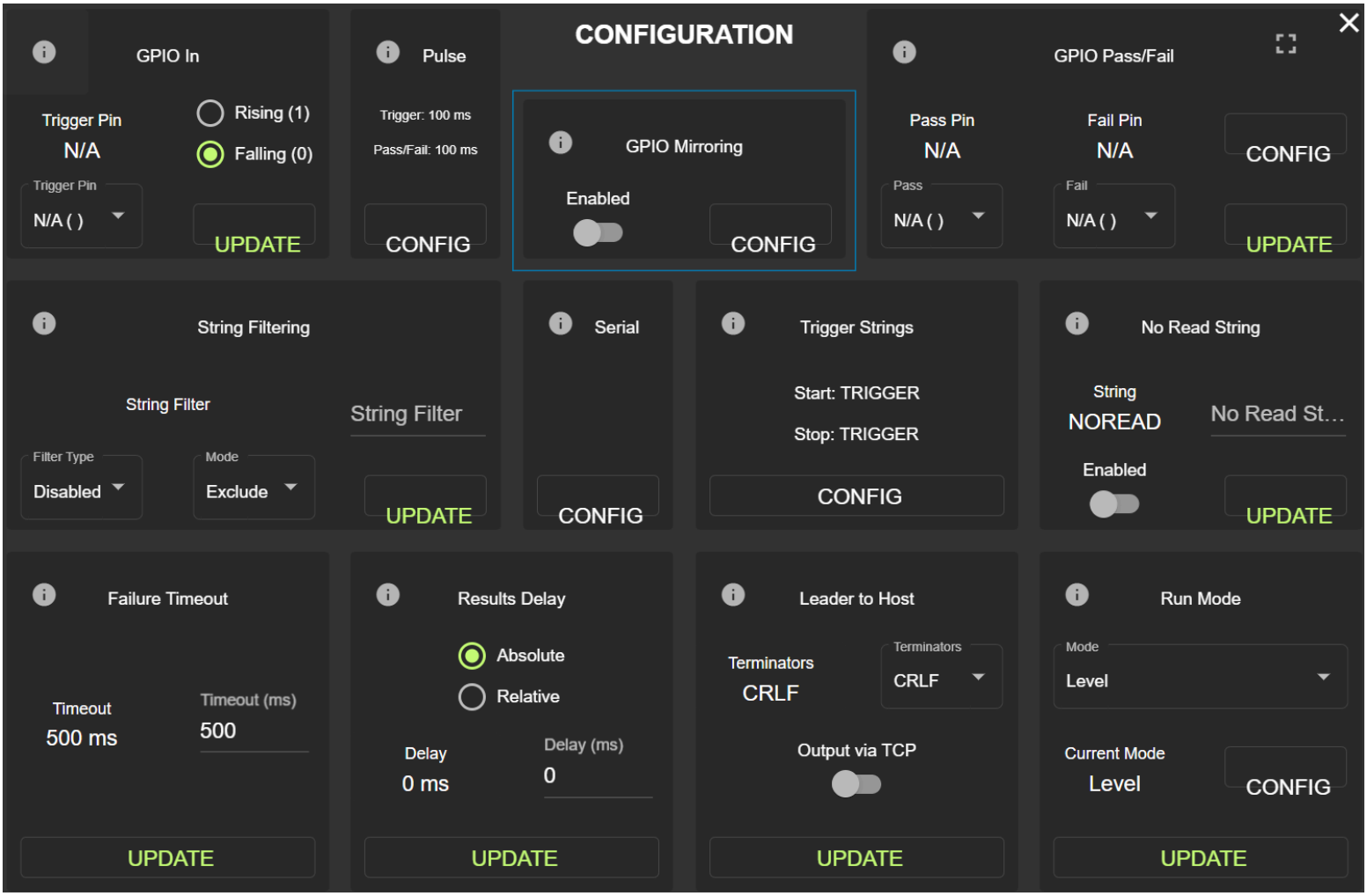
GPIO Mirroring

Use the HMI to turn GPIO Mirroring on or off and inverse the mirroring to allow Input 0 and Input 1 to mirror to Out1 and Out0.

- Enabled GPIO Mirroring: the output signal goes high whenever the input pin that it is mirroring goes high.
- Disabled GPIO Mirroring: the output signal goes high for its predefined pulse width and then goes low immediately after.

- Enabling Opposite pins: input 0 mirrors Out0, and Input1 is mirrored to Out1 by default. Enabling opposite inverses the mirroring to allow Input0 and Input1 to mirror Out1 and Out0, respectively.

1. Navigate to the **Configuration** screen using the left menu on the **Home** screen.
2. Locate the **GPIO Mirroring** section and toggle **GPIO Mirroring** to enable.

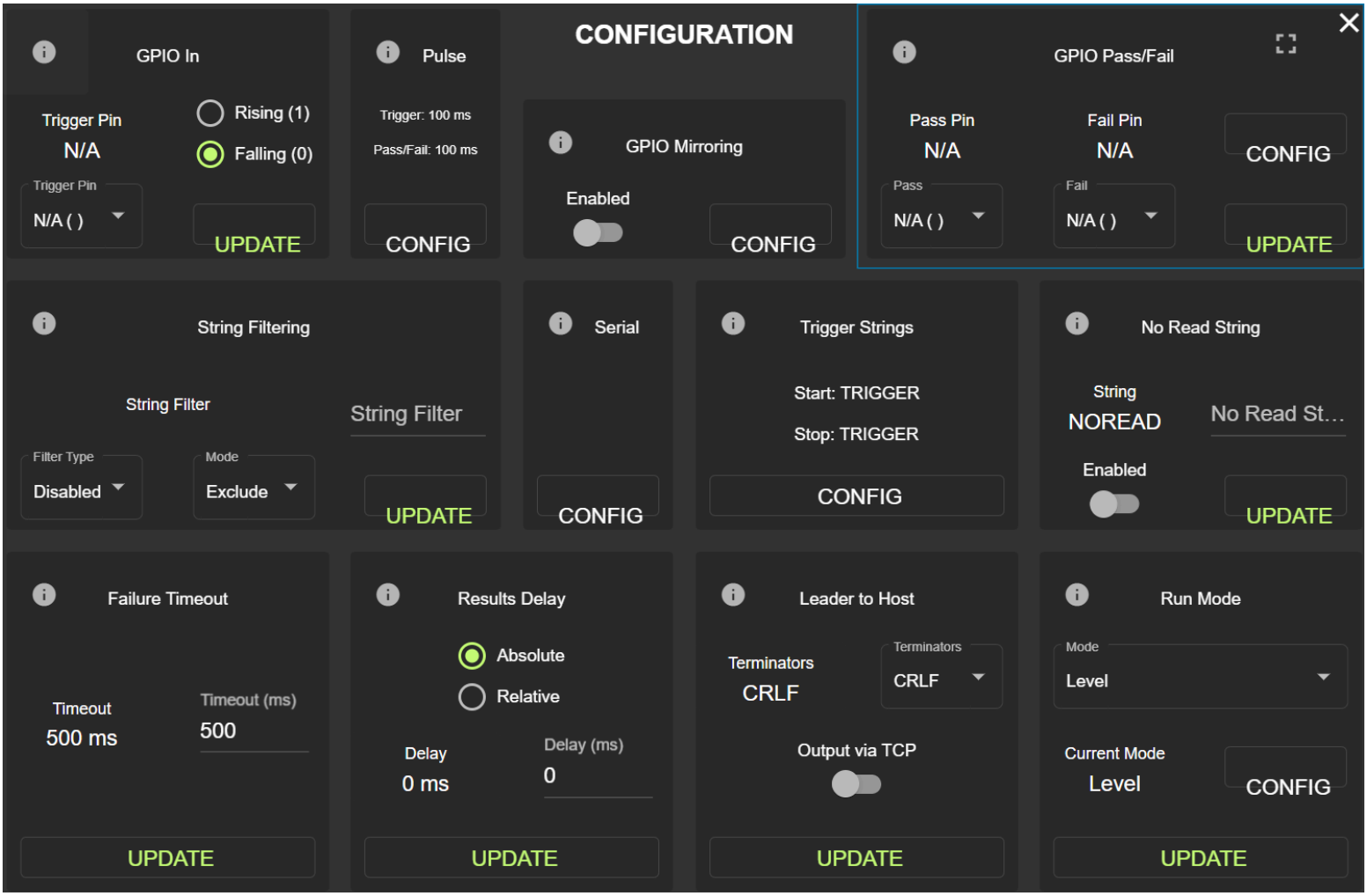


3. Click **Config** and select an **Input Pin** and an **Output Pin** from drop-down.
4. Click **Update** to apply the changes.

GPIO Pass/Fail

Use the HMI to configure GPIO Pass/Fail on an output pin of your choice.

1. Determine which output to use for Pass/Fail. The output triggers followers and cannot be used to output scanning results.
2. Navigate to the **Configuration** screen using the left menu on the **Home** screen.
3. Use the top-right **GPIO Pass/Fail** section to select Pass and Fail pins from the drop-down menu.



4. Set the toggle to **Enabled** and click **Update**.

String Filtering

The device supports string filtering modes such as exclude and include and string filters such as exact match, start with, ends with, contains, and RegEx.

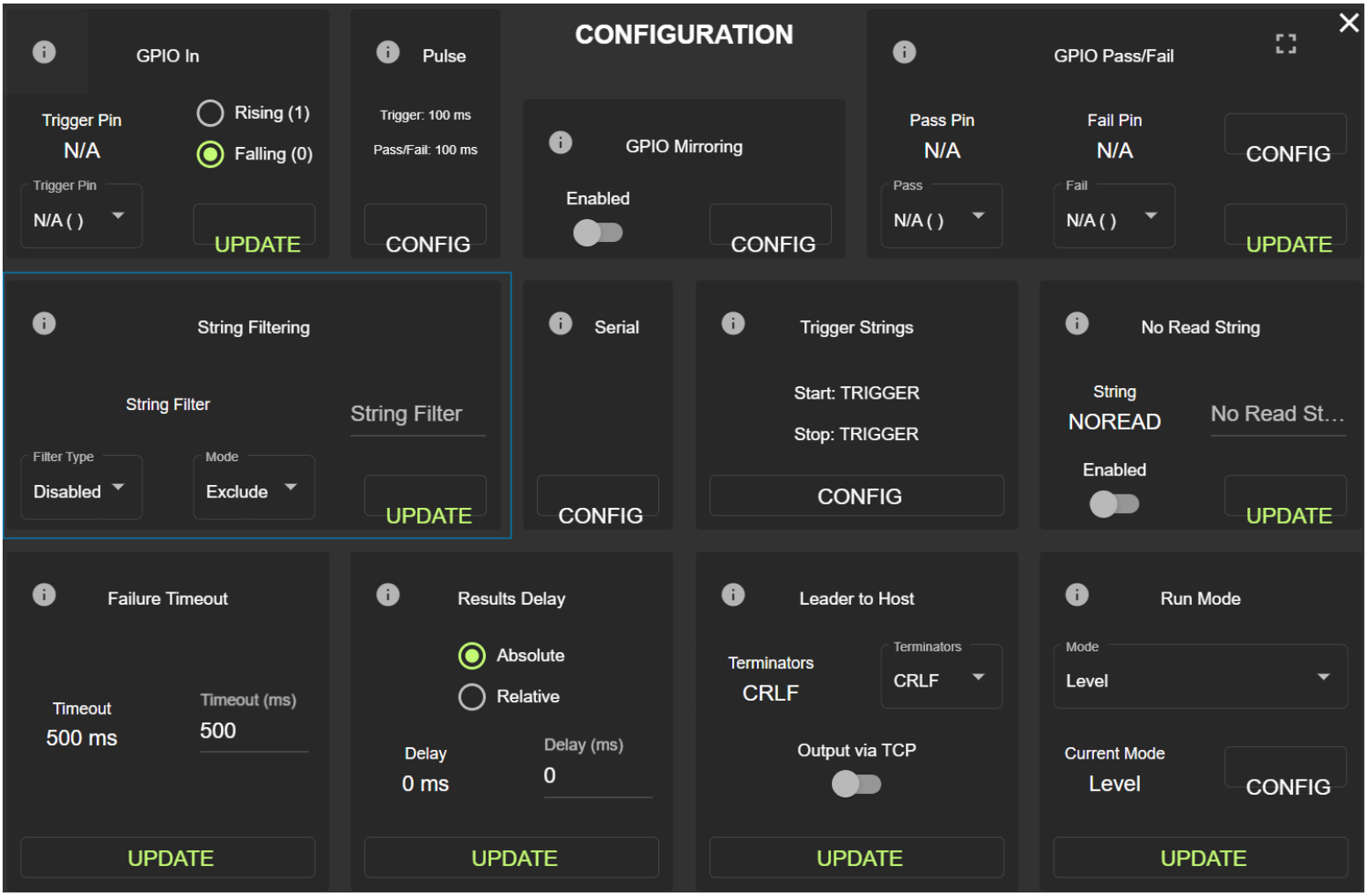
- **Exclude:** the device excludes barcode results that match the specified string filter.
- **Include:** the device only includes barcode results that match the specified string filter.

Supported string filters include:

- **Disabled:** disables string filtering.
- **Exact Match:** filters strings that exactly match the input string.
- **Start With:** filters string matches at the beginning of the string.
- **Ends With:** filters string matches at the end of the string.
- **Contains:** filters string matches anywhere in the string.
- **RegEx:** uses a regular expression to define string filtering.

To set up string filtering:

1. Navigate to the **Configuration** screen using the left menu on the **Home** screen.
2. Use the **String Filtering** section to select the desired **Mode**, **Input**, **String Filter**, and **Filter Type**.



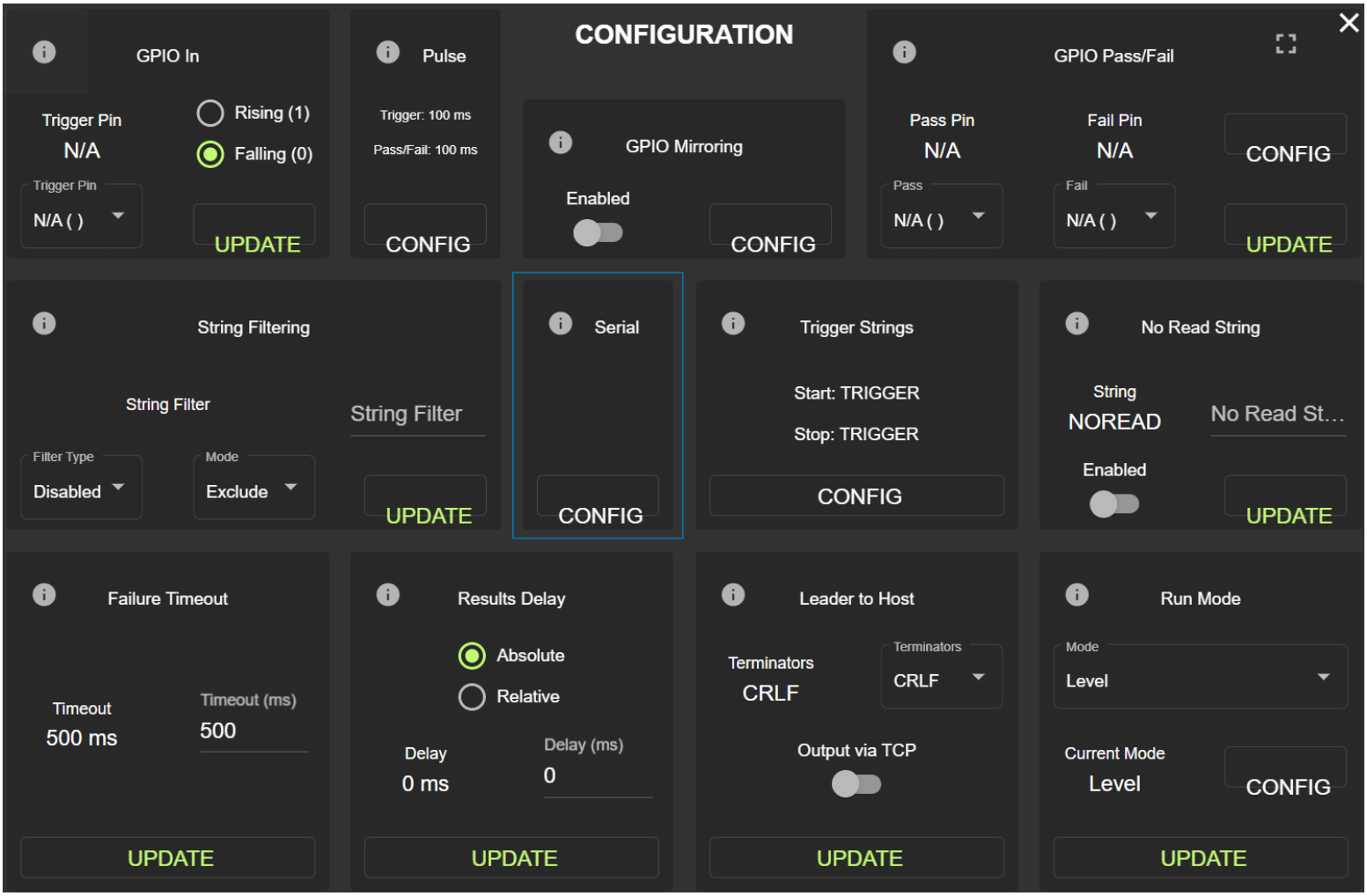
3. Click **Update** to apply the changes.

Serial Triggering

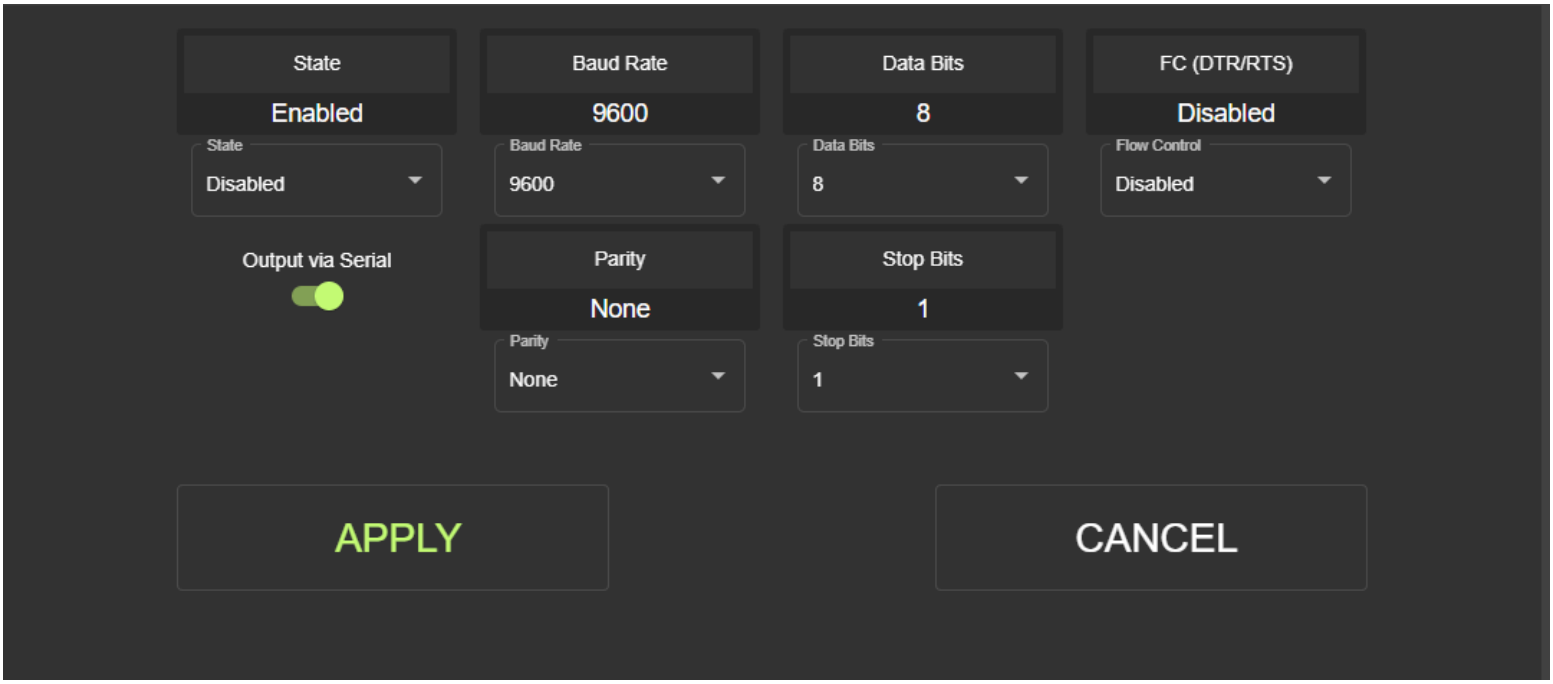
Set up a Serial Trigger from the host to the Connectivity Gateway.

NOTE: There is no serial connection between the leader and the follower devices; triggering between the devices occurs over TCP/IP.

1. Navigate to the **Configuration** screen using the left menu on the Home screen.



2. Locate the **Serial** section and click **Config** to set the serial settings for the device.

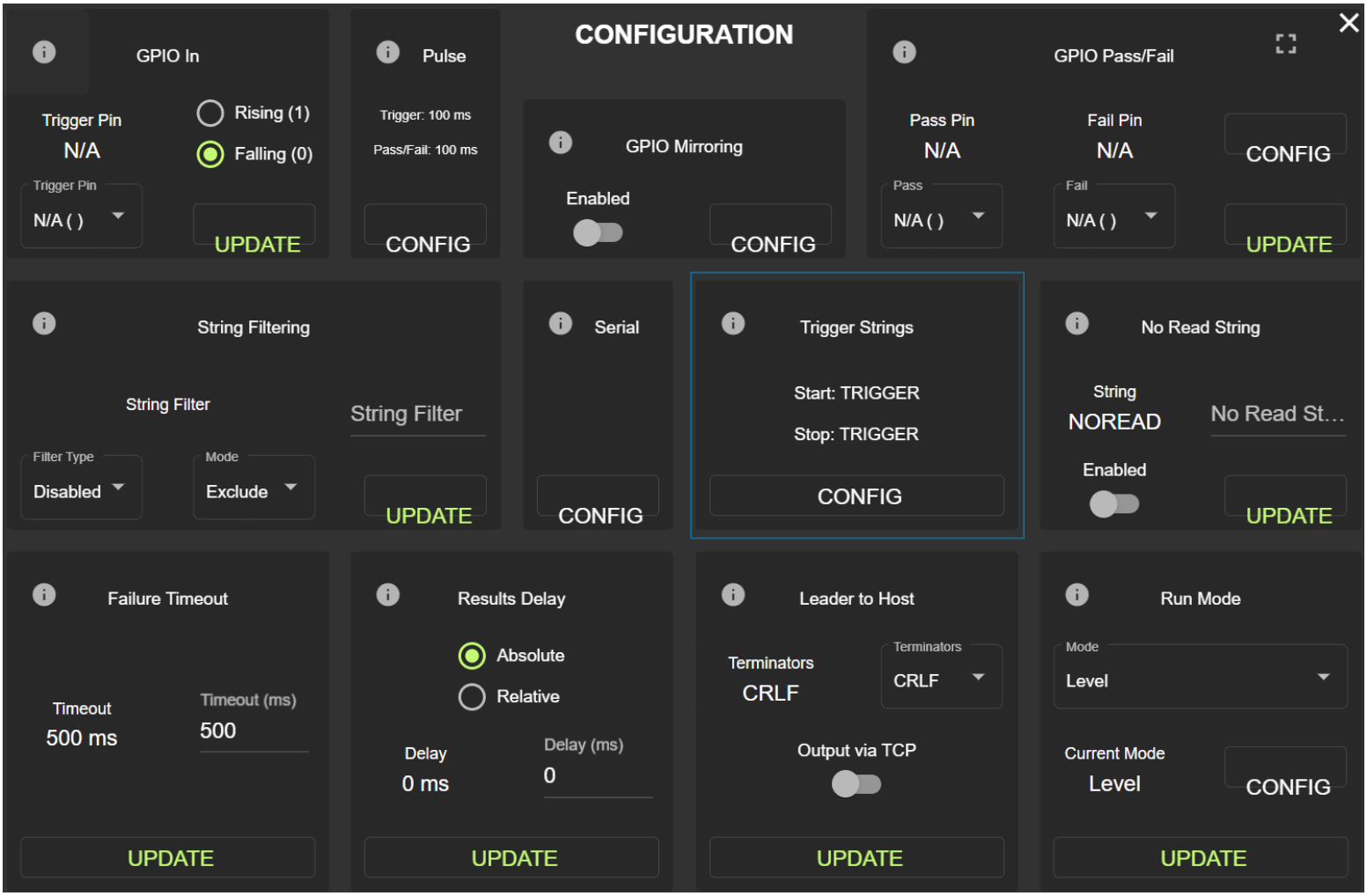


3. Configure the form fields and click **Confirm** to enable the settings on the device.

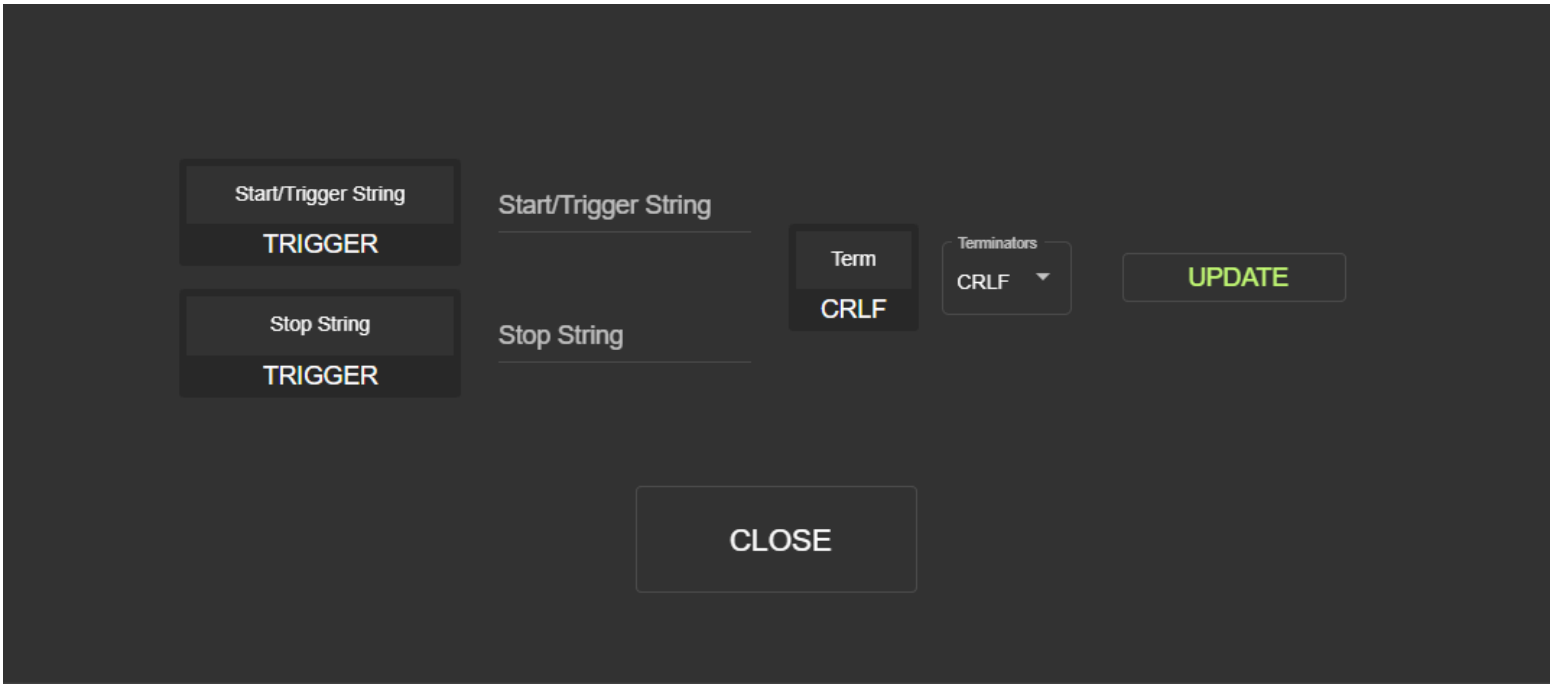
TCP/IP Triggering

Before setting up TCP/IP triggering on the device, configure the Host to Leader port and follower devices.

1. Navigate to the **Configuration** screen using the left menu on the **Home** screen.
2. Locate the **Trigger Strings** section and click **Config**.



3. Select the terminator for each trigger from the drop-down menu and click **Update** to apply the changes.

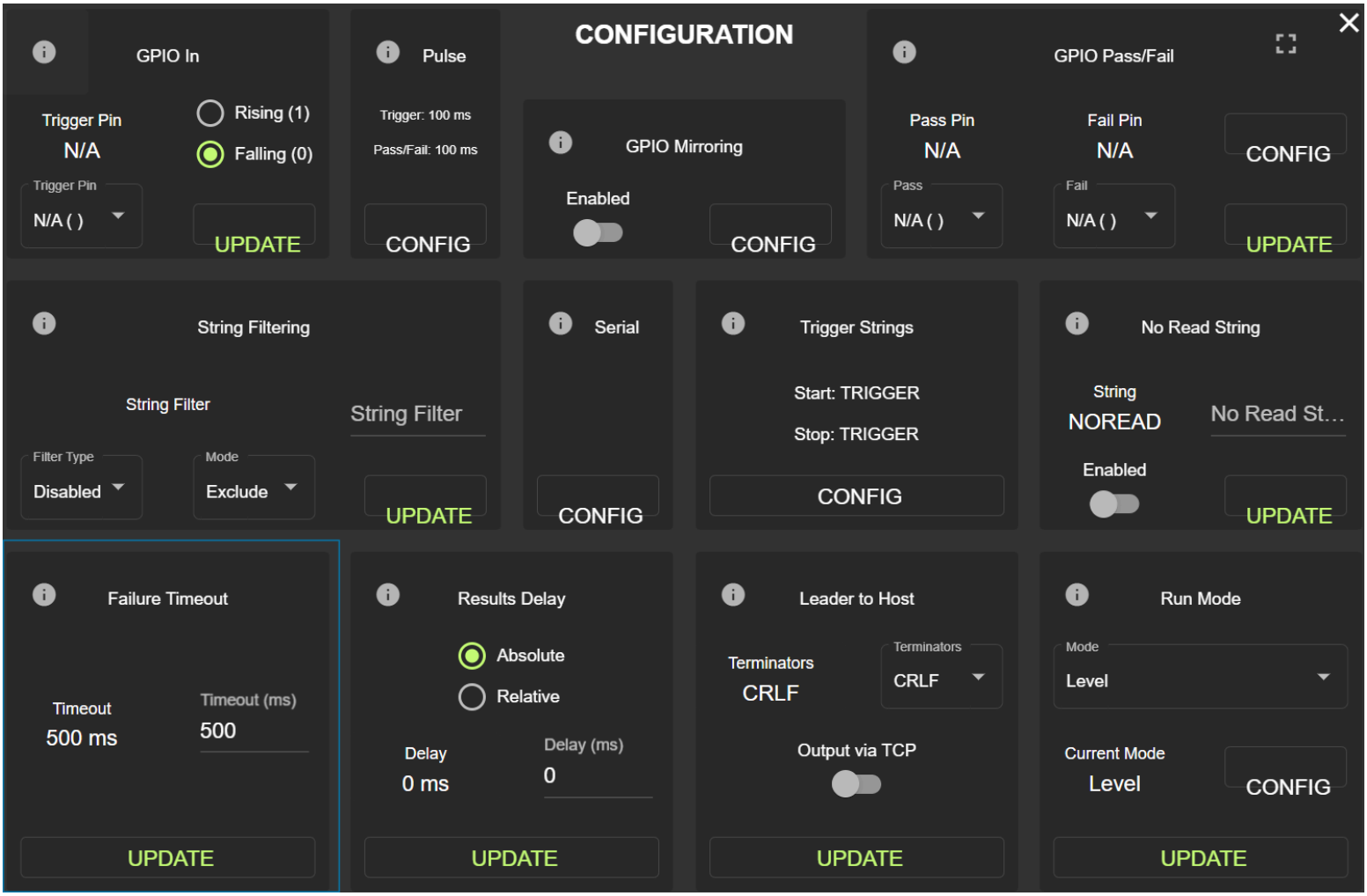


NOTE: Serial Interface does not support a null terminator.

Failure Timeout

Set a failure timeout to account for trigger input delays.

1. Navigate to the **Configuration** screen using the left menu on the **Home** screen.
2. Use the bottom left **Failure Timeout** section to specify a timeout.



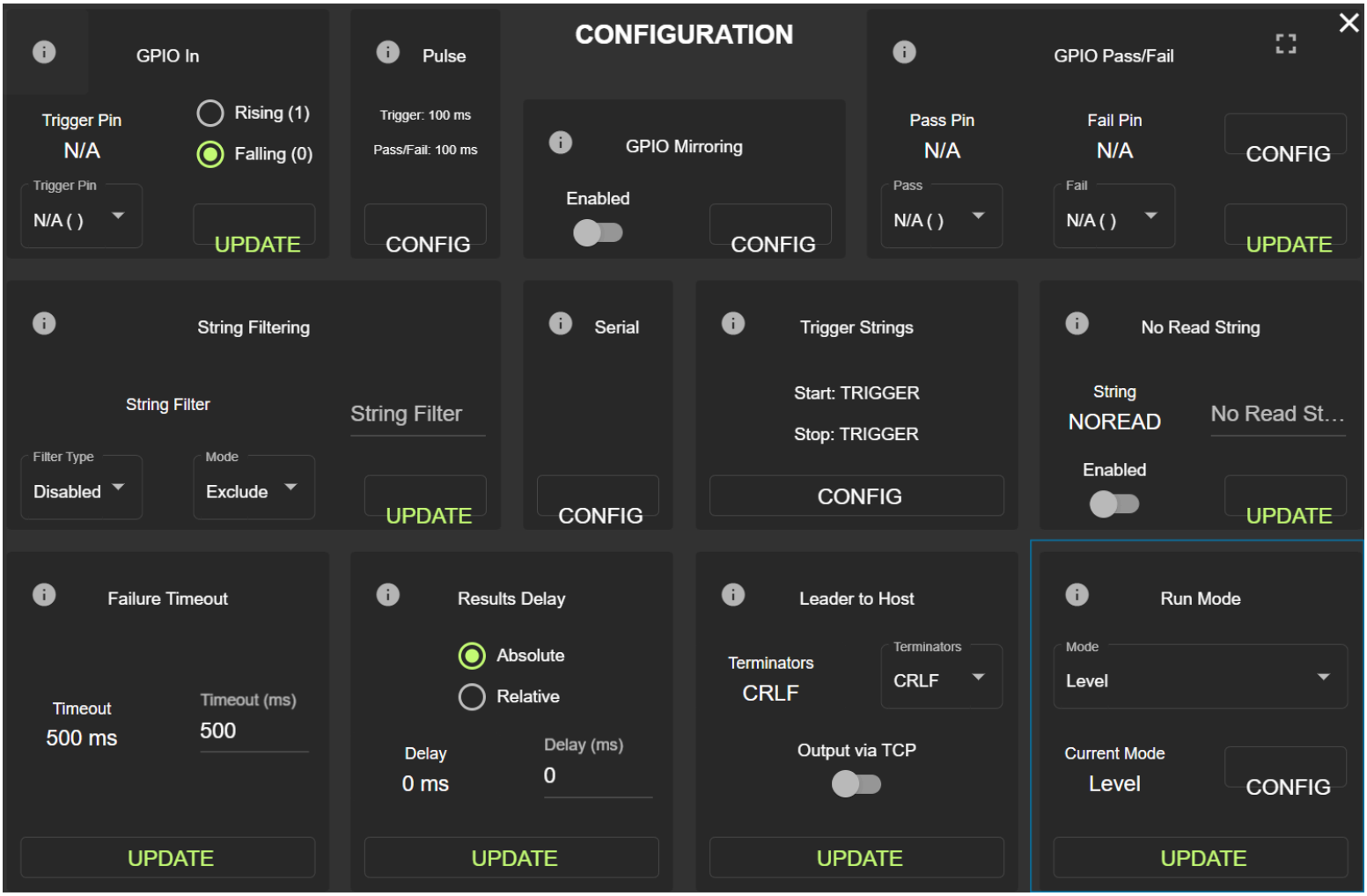
3. Click **Update** for the changes to take effect.

NOTE: Ensure your failure timeout accounts for any trigger input delays specified on the follower side.

Run Mode

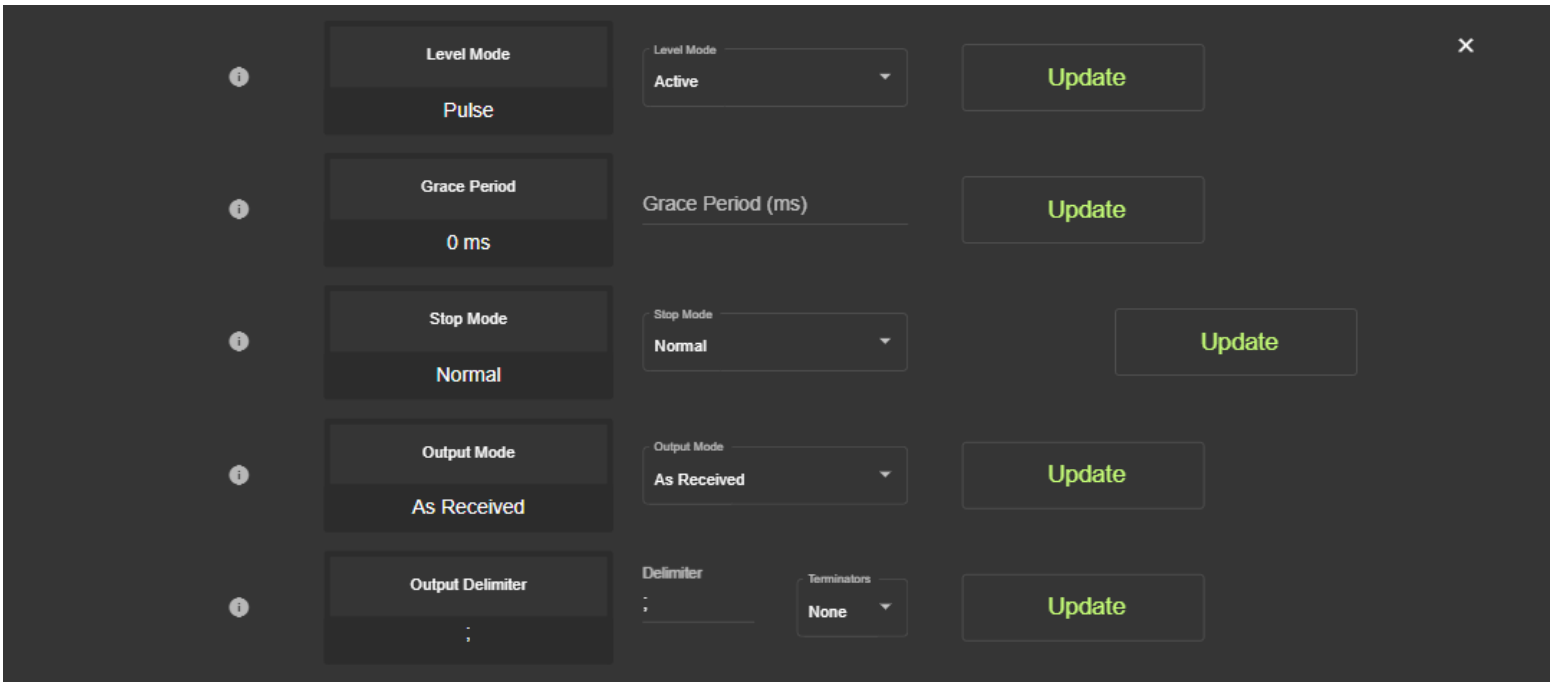
The Connectivity Gateway operates in Single Shot, Continuous, or Level mode.

1. Navigate to the **Configuration** screen from the **Home** screen using the left navigation menu.
2. Use the bottom right **Run Mode** section on the **Configuration** screen to change between **Single Shot**, **Continuous** and **Level** modes.



3. Click **Config** to edit **Run Mode** settings.
4. Configure the required trigger settings:
   - Level Mode - determines how a level starts and stops. The default setting is **Active**. When set to **Active**, the level starts/stops on active high or low. Set this value to **Pulse** to start or stop a level when the trigger tag goes high or low twice. This can increase workflow efficiency by terminating data capture processes once a successful decode is achieved, preventing unnecessary resource usage.
   - Grace Period - the amount of time to wait after a level stop signal is received to accept incoming barcodes before rejecting them.
   - Stop Mode - determines how the level session stops. The default setting is **Normal**. When set to **Normal**, the application waits for the stop trigger signal from the host or GPIO and forwards the data to the followers. When set to **On Decode**, the application immediately sends the stop trigger to the followers upon the first decode.
   - Output Mode - determines how barcodes are output over all configurable host interfaces while running in level mode. The default setting is **As Received**. When set to **As Received**, barcodes are sent as received. When set to **Level End**, barcodes are combined into a single payload separated by the output delimiter.

- Output Delimiter - determine the delimiting characters separating the barcode output after a level session when running in **Level End** output mode.



5. Click **Update** for the changes to take effect.

6. Observe the warning screen indicating that scan data is erased after the changes are implemented. Acknowledge the warning to proceed with your changes.

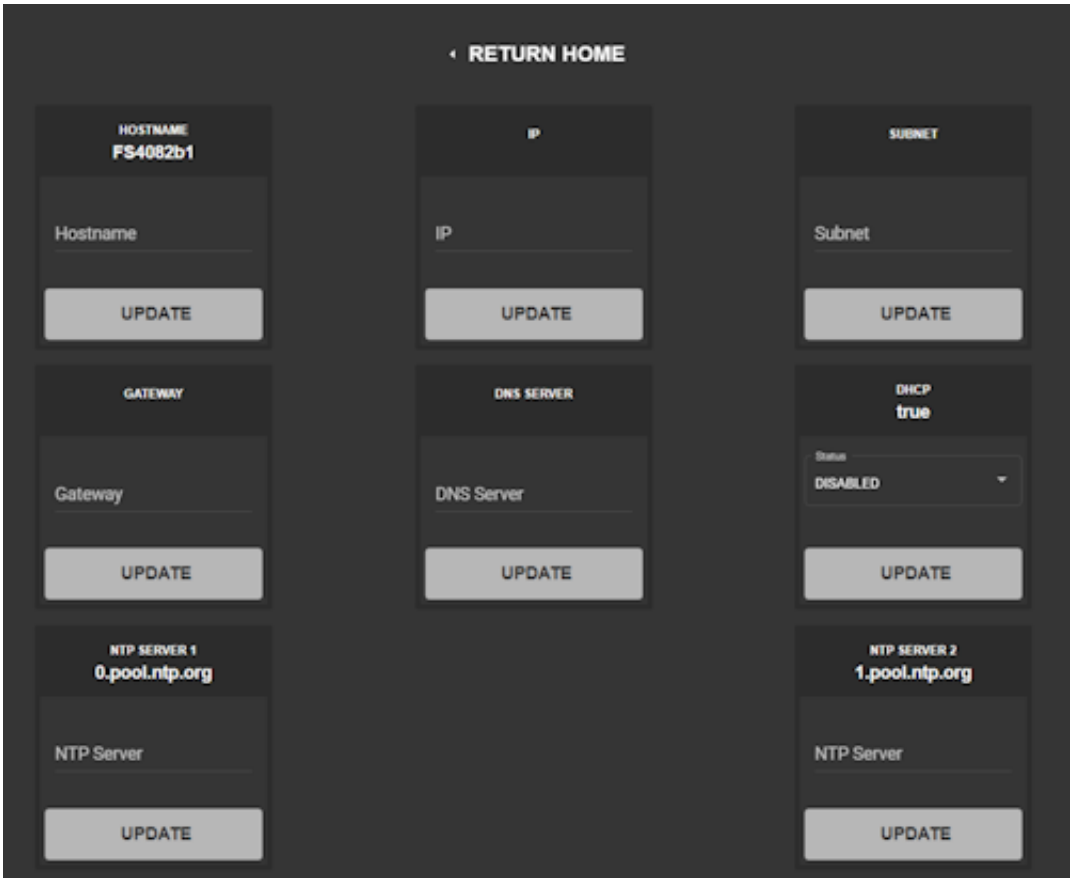
Administrator Settings

Accessible administrator settings include network, GPIO, PLC, ZETI, database, and alerts.

Setting Up a Network

Configure network settings, including the hostname, IP address, subnet, gateway, and DNS server.

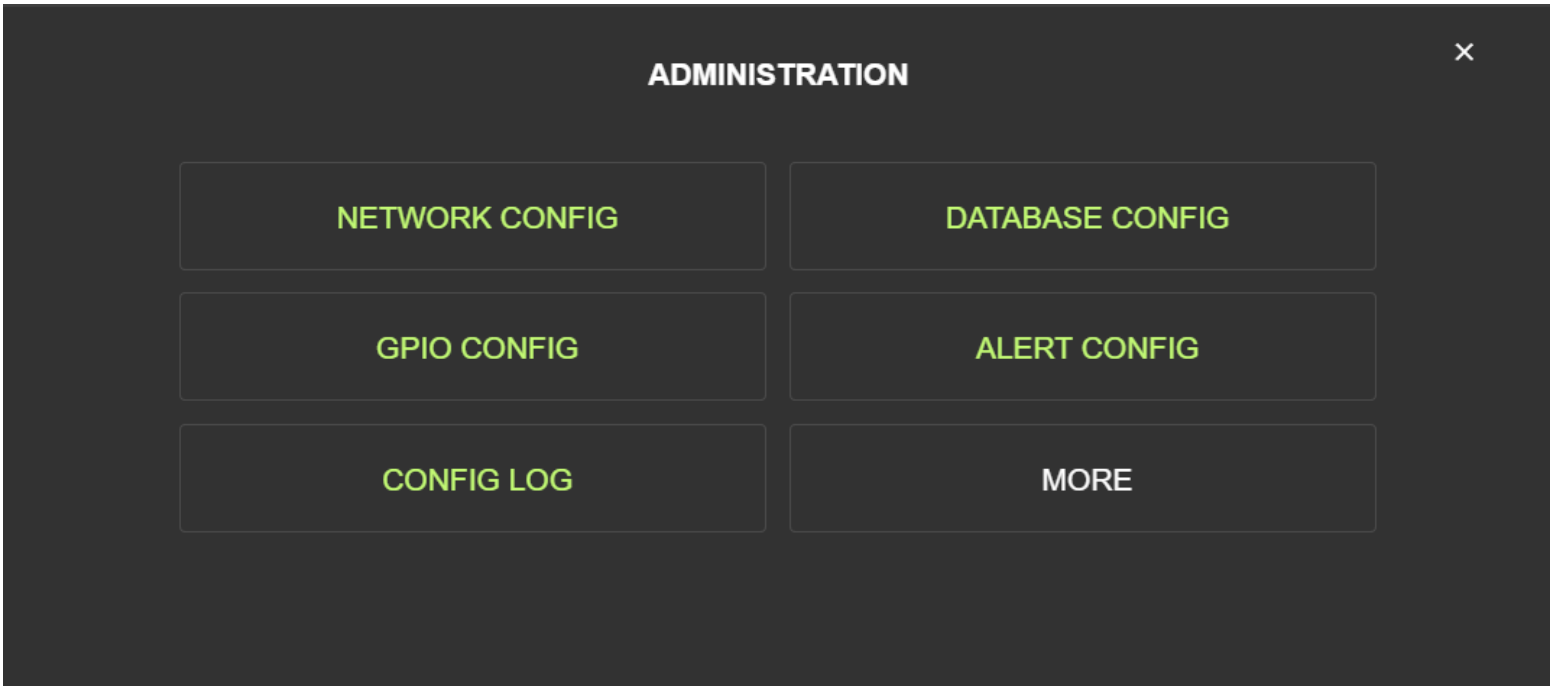
Access **Network Config** from the **Admin Settings** menu to edit the hostname and server settings and click **Update** to save each setting.



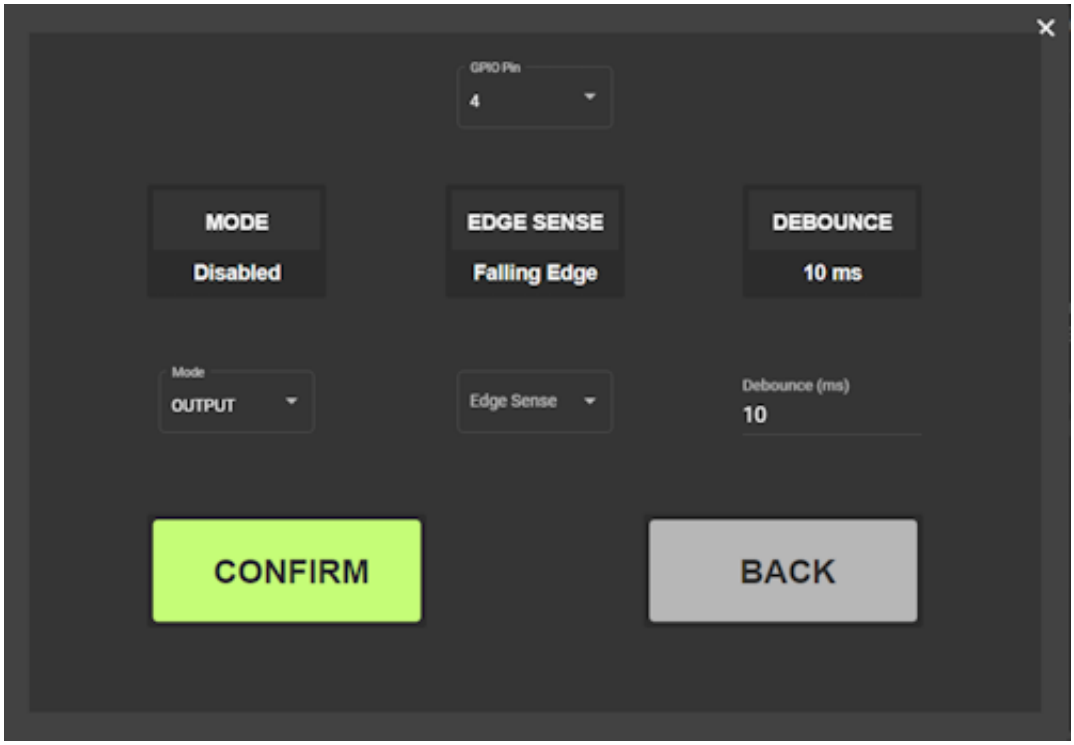
Configuring GPIO

Configure GPIO settings such as Mode, Edge Sense, and Debounce.

1. Access **GPIO Config** from the **Admin Settings** menu.



2. Select the GPIO pin from dropdown and configure the **Mode**, **Edge Sense**, and **Debounce** settings.

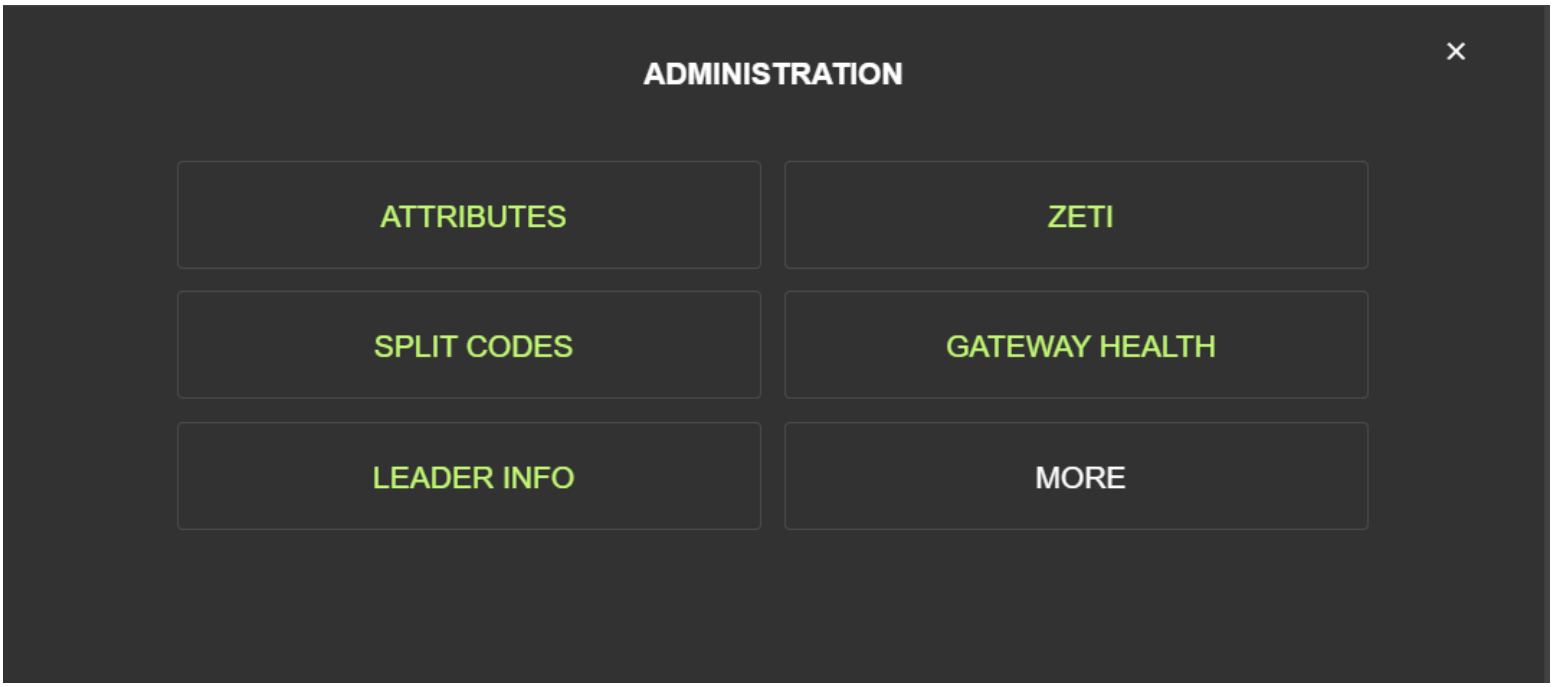


3. Click **Confirm** to apply the changes.

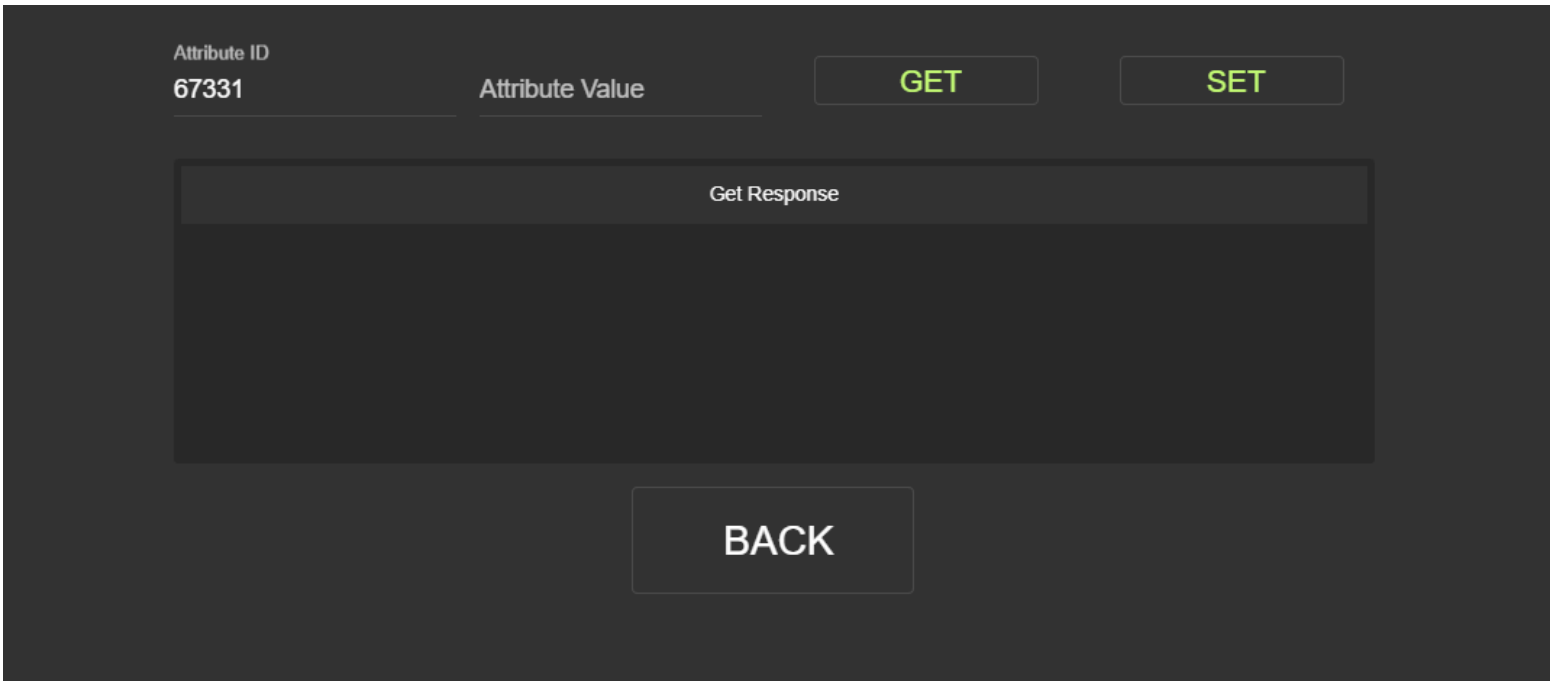
Attributes

Provide the Attribute ID or value to set or retrieve the response.

1. Access **Attributes** from the **Admin Settings** screen.



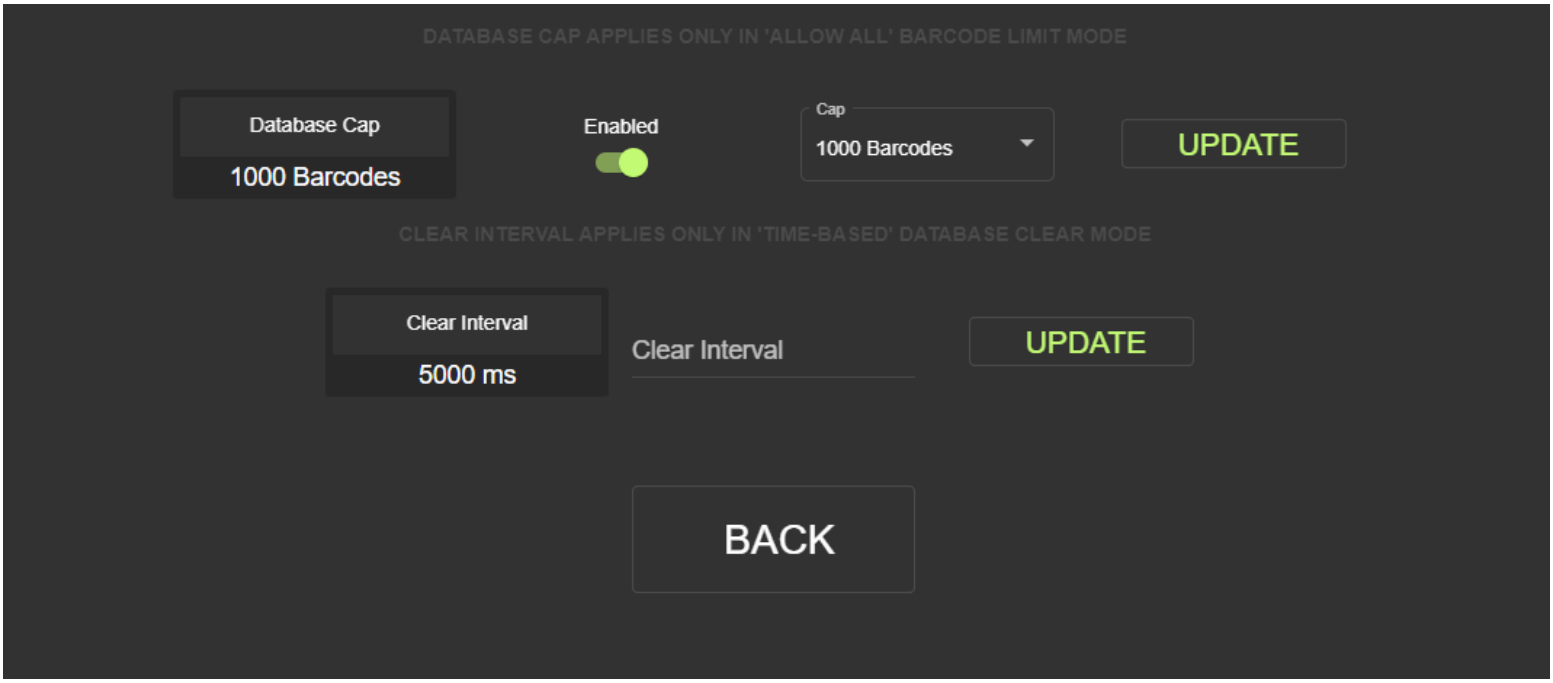
2. Enter the **Attribute ID** and **Attribute Value** and click **Get** or **Set** to retrieve the response.



Configuring the Database

Configure the database to set a data capacity or clear interval.

1. Click **Database Config** on the **Admin Settings** menu.
2. Enable a **Database Cap** by sliding the toggle to the right and selecting an entry capacity from the drop-down menu.
3. Enter a **Clear Interval** in ms.

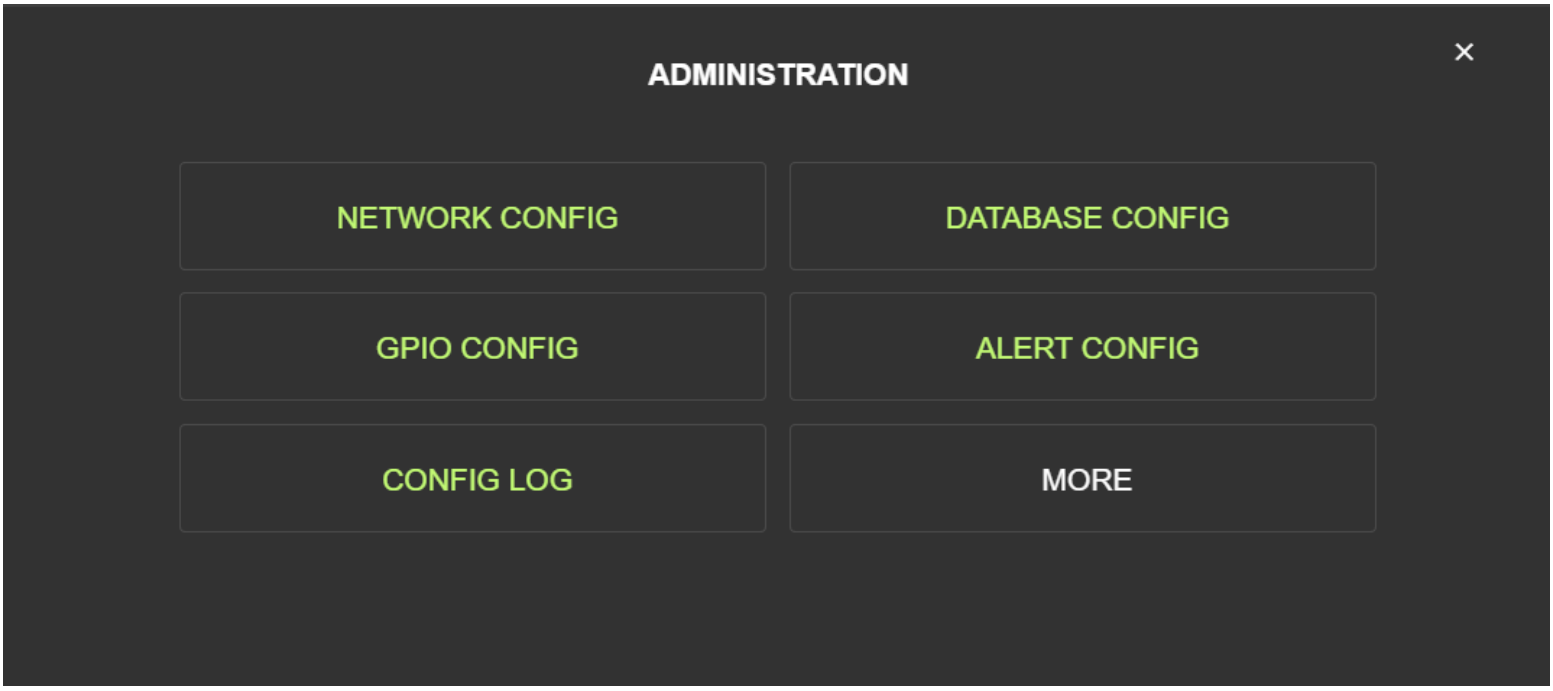


4. Click **Update** to apply the changes.

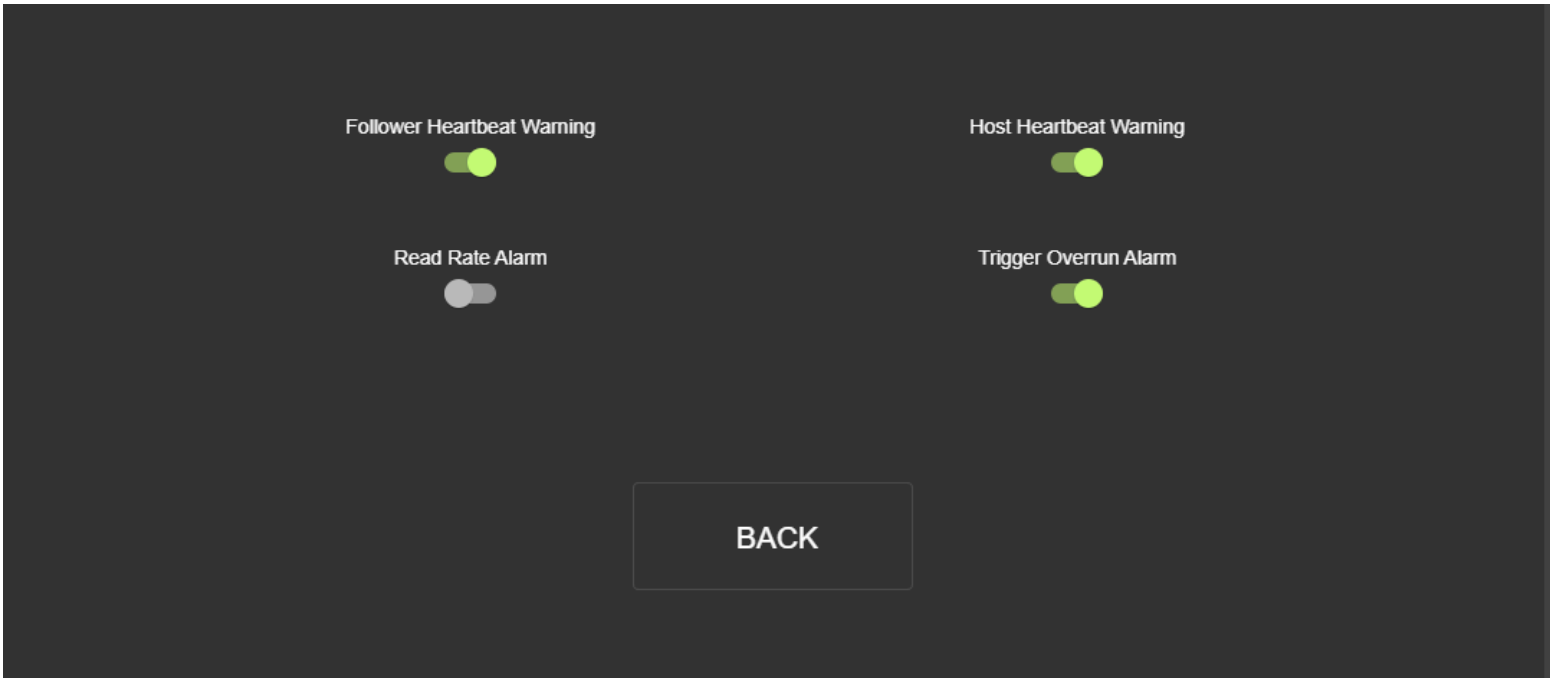
Enabling Alerts

Enable alerts to understand the timeout intervals of the follower and leader devices or if the read rate drops.

1. Click **Alert Config** on the **Admin Settings** menu to enable alerts.



2. Enable an alarm for **Follower Heartbeat**, **Host Heartbeat**, **Read Rate**, or **Trigger Overrun** by sliding the corresponding toggle to the right.



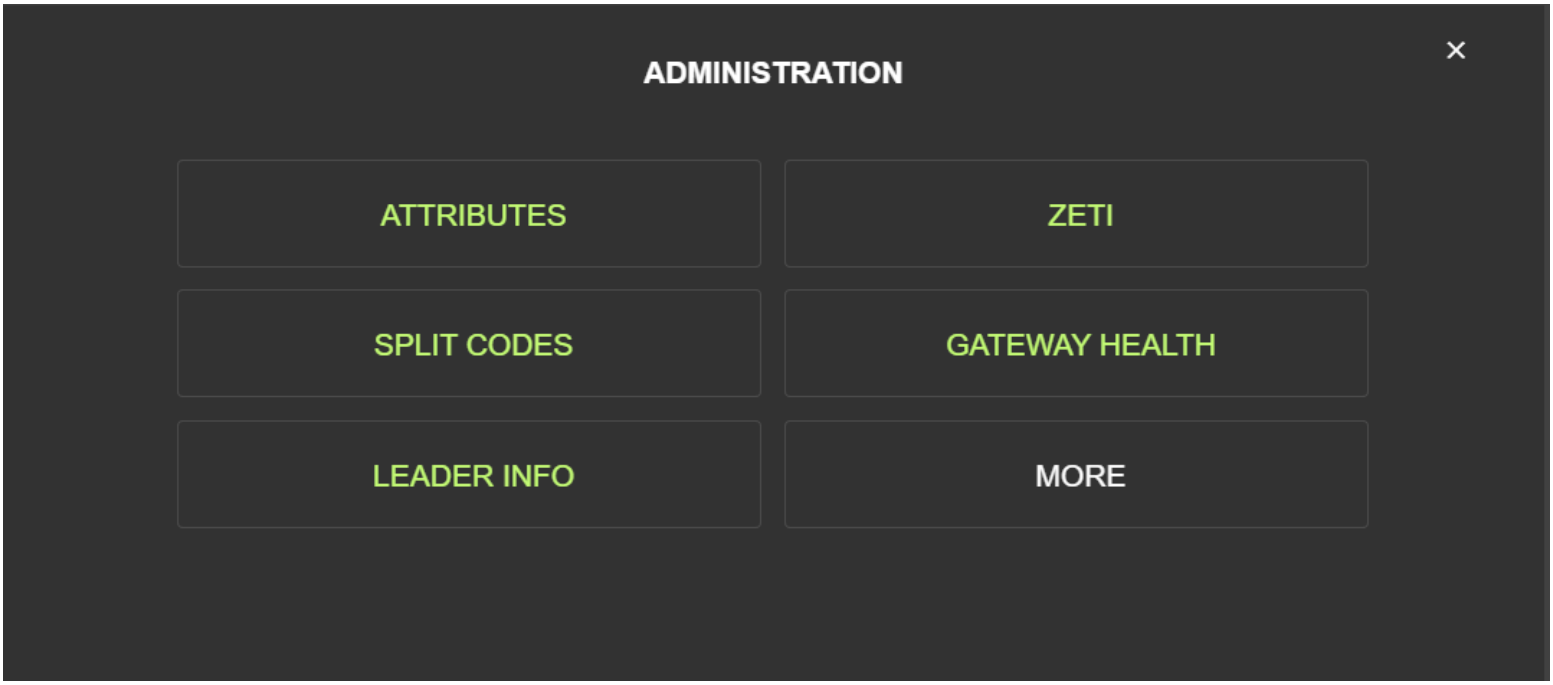
- Follower Heartbeat: enables a heartbeat (timeout interval) for follower devices.
- Host Heartbeat: enables a heartbeat (timeout interval) for the host device.
- Read Rate: enables a notification on the Home screen if the read rate drops below 95% over 1000 scans.

- Trigger Overrun: enables a notification on the Home screen when triggering occurs faster than the result is received from the follower or when a failure timeout occurs.

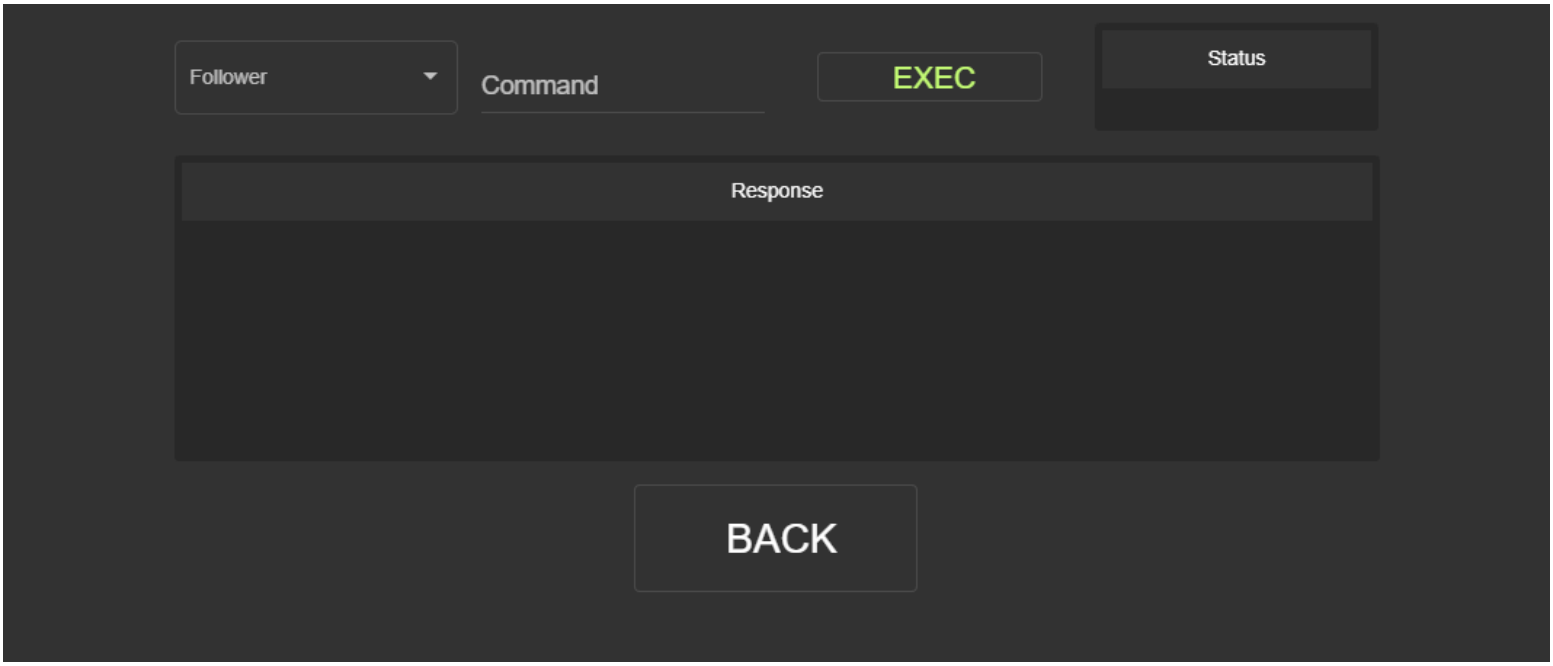
Using ZETI with Follower Devices

Use Zebra Easy Text Interface (ZETI) to execute specific commands and retrieve a response.

1. Access **ZETI** from the **Admin Settings** menu.



2. Select a follower from the drop-down menu and enter a command.
3. Click **Exec** to execute the command and observe the **Status** and **Response**.



Split Codes

Configure follower devices for Split Code functionality using Zebra Aurora Focus.

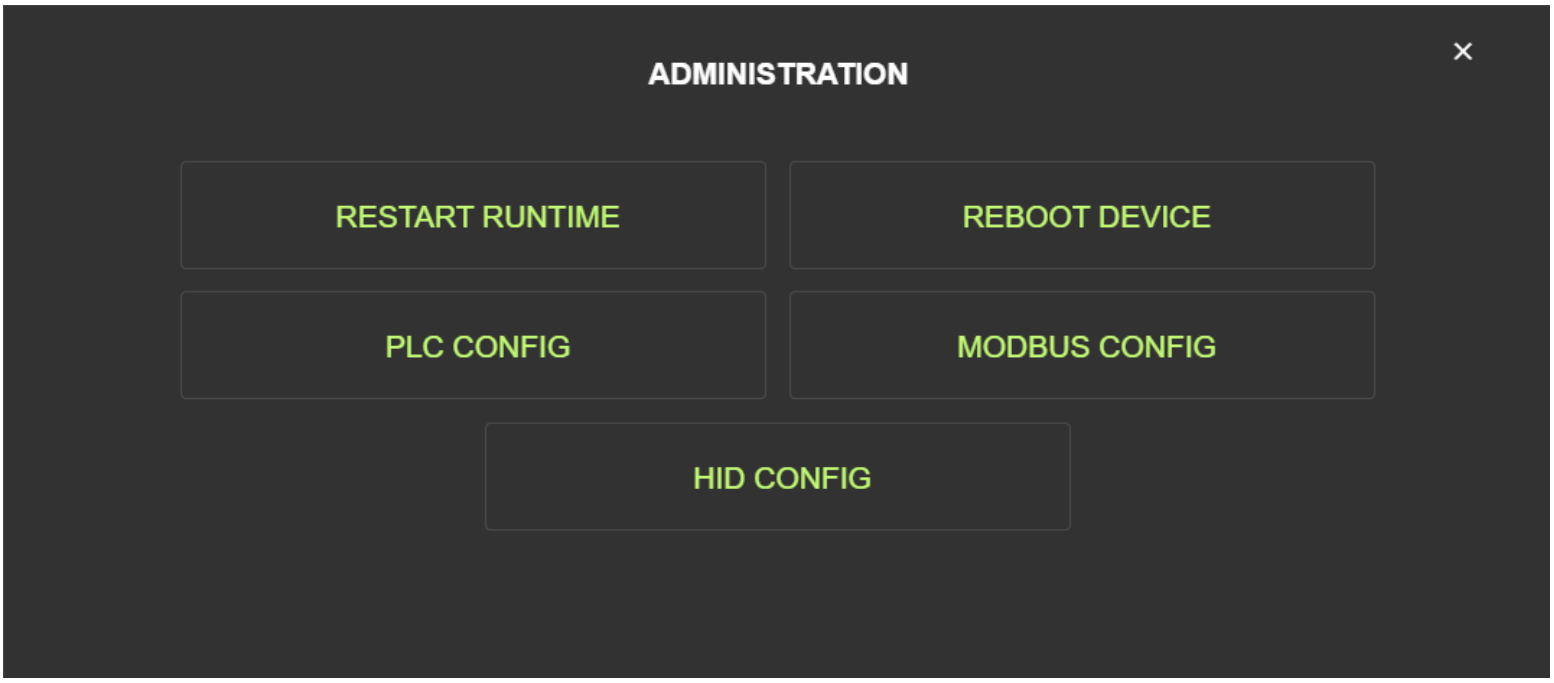
1. Create a new job for the follower device and navigate to the Build tab in Zebra Aurora.
2. Navigate to the **Symbologies** section. Ensure that only the symbologies necessary for the application are enabled.
3. Navigate to the **ManyCode** tab. Configure the total number of Barcodes to Decode (3 max). Sort by symbology and enable partial results.
4. Navigate to the **Data Formatting** tab and select **Advanced Formatting**.
5. Select **ManyCode** and input the following:
   - Prefix: None
   - Data
     - Many Code String
     - Custom String "~~$$~~"
   - Suffix: None
   - Delimiter
     - Data Type: None
     - End: None
     - ManyCode String: None
6. Select All Symbologies
   - Prefix: None

- Data
  - Symbology
  - Custom String "",
  - Full String: Base 64
  - Custom String: "~~##~~"
- Suffix: None
- Delimiter
  - Data Type: None
  - End: None

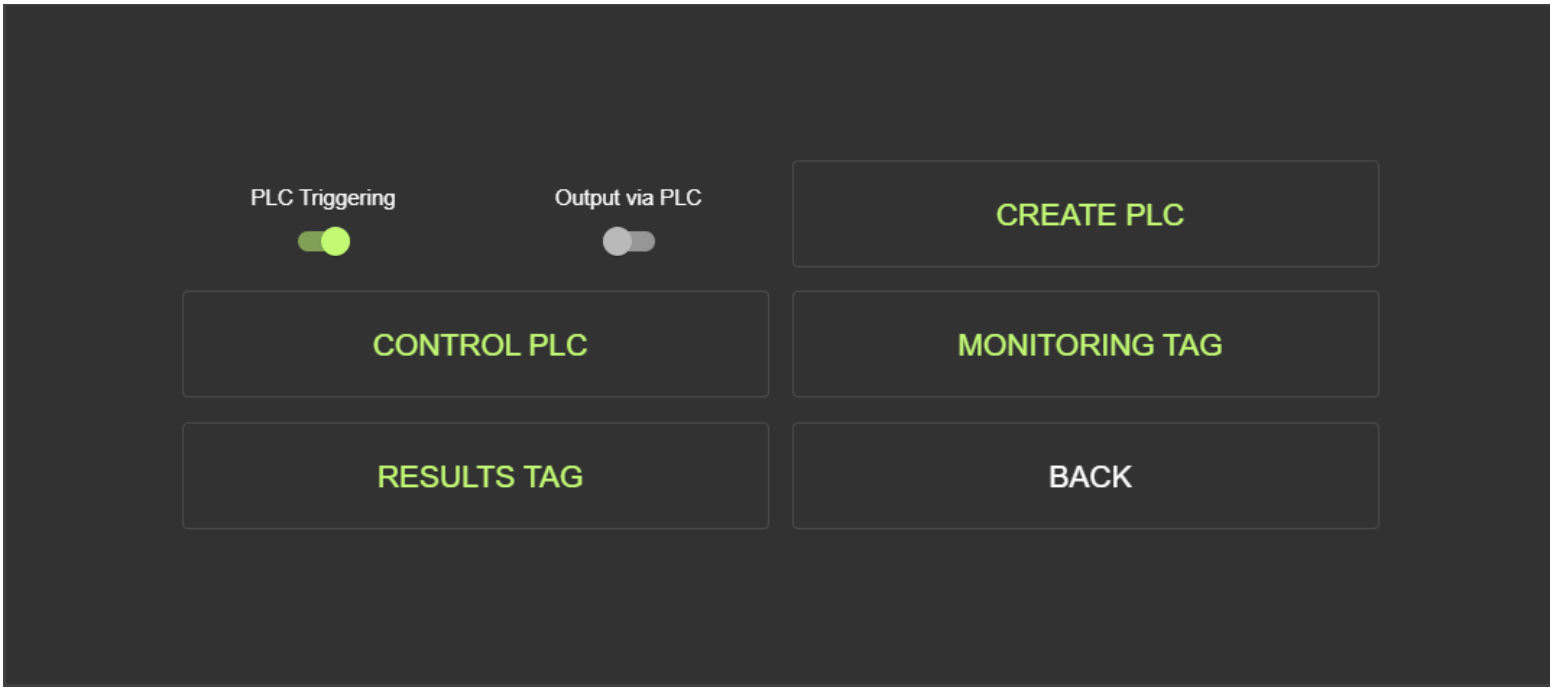
PLC Configuration

Create a PLC device to connect to the Connectivity Gateway.

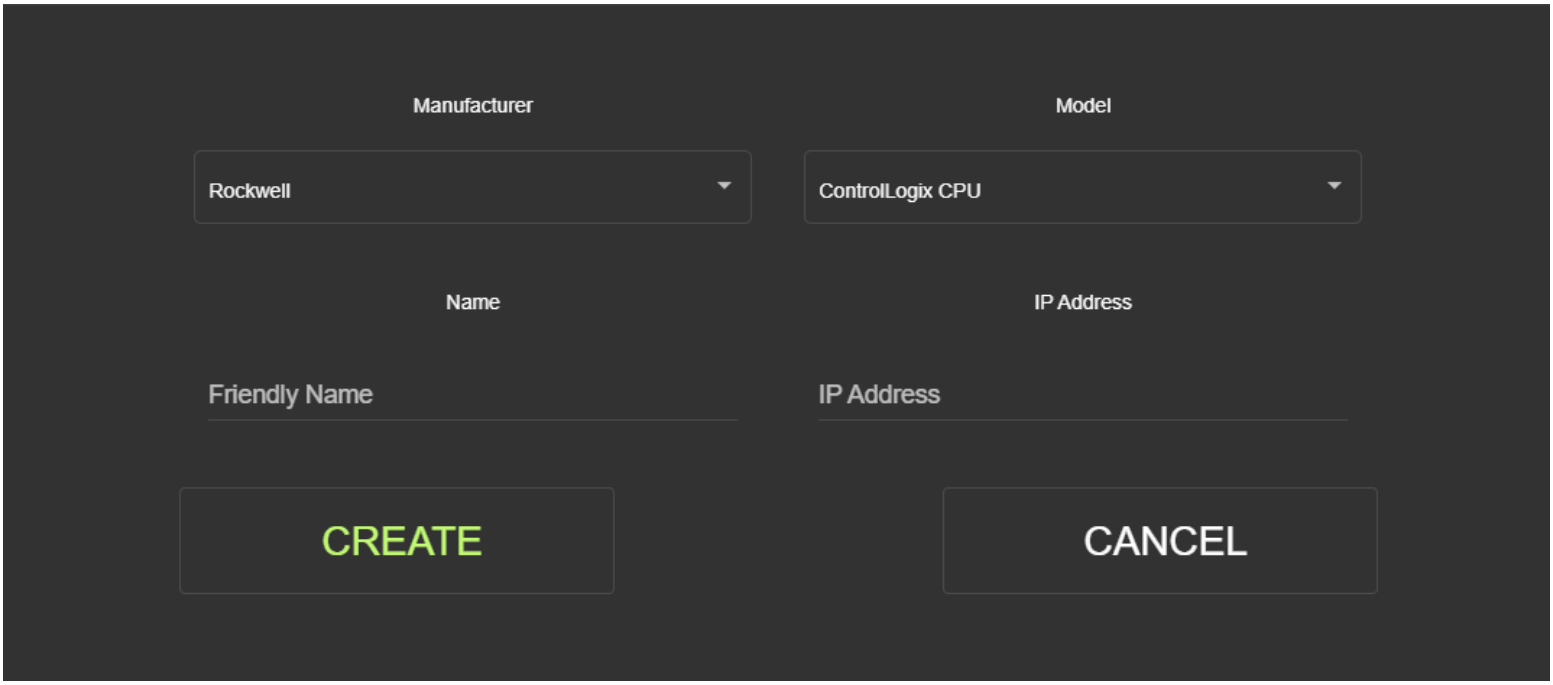
1. Select **PLC Config** from the **Admin Settings** menu.



2. Select **Create PLC** from the **PLC Configuration** menu.



3. On the Create PLC screen, select the **Manufacturer** and **Model** of the PLC, specify a **Friendly Name** for the PLC, and enter the **IP Address**.

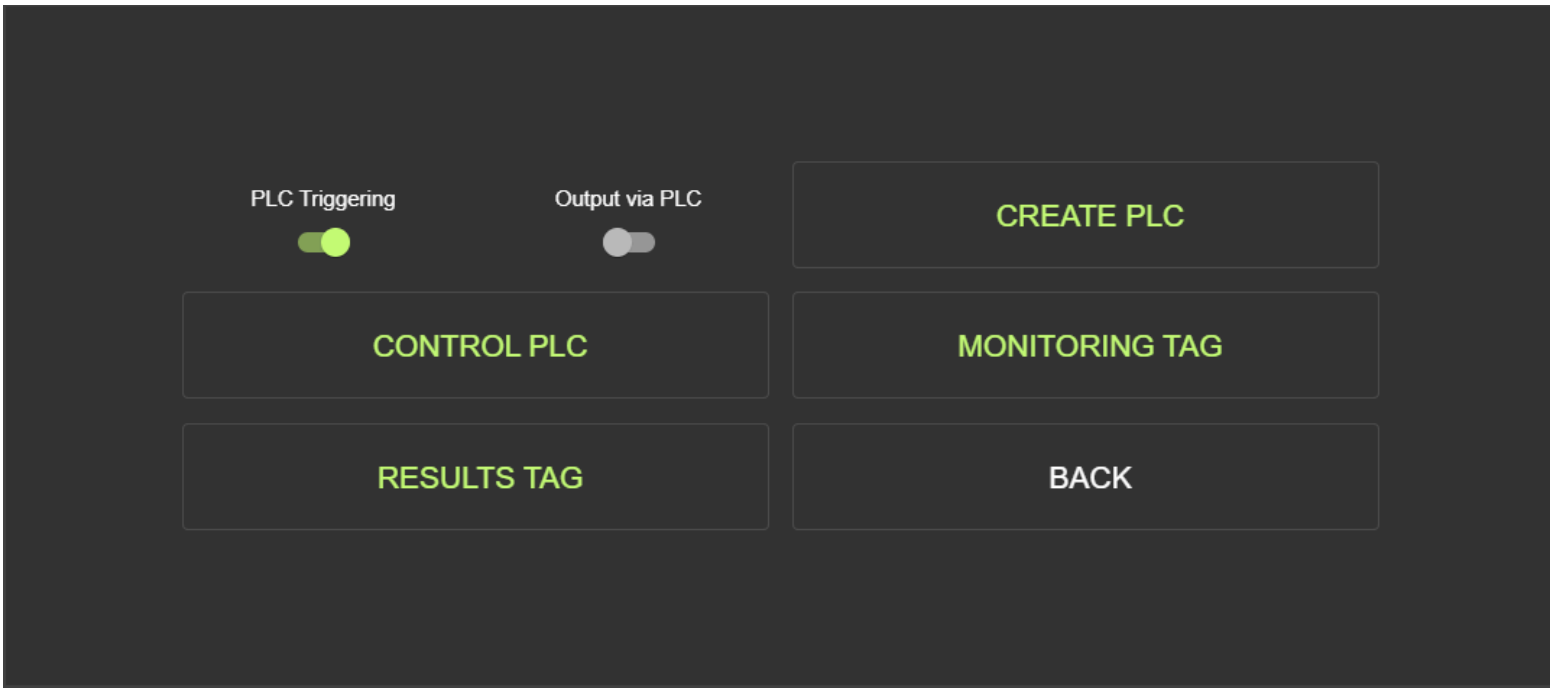


4. Click **Confirm** and observe the confirmation message.

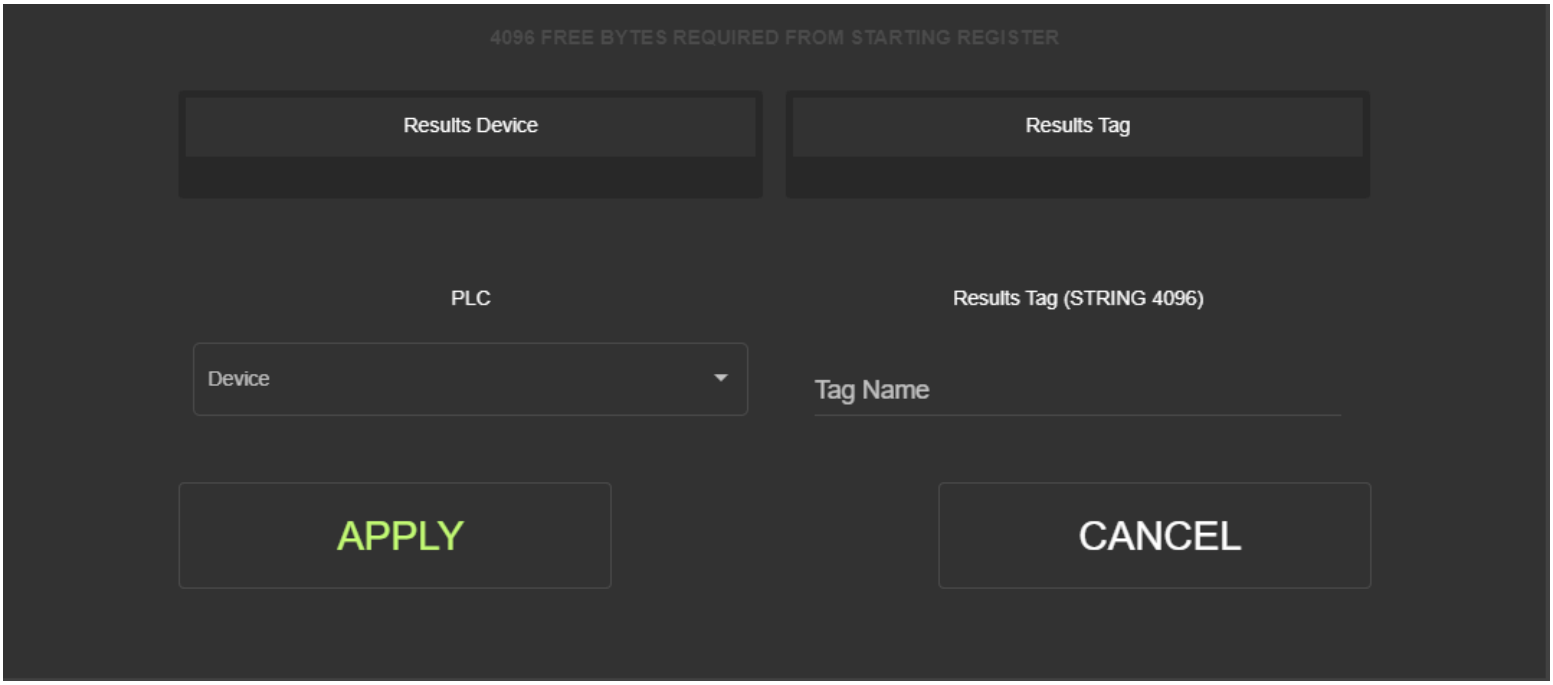
Sending Results to a PLC Device

Use the Connectivity Gateway to send results to the connected PLC device.

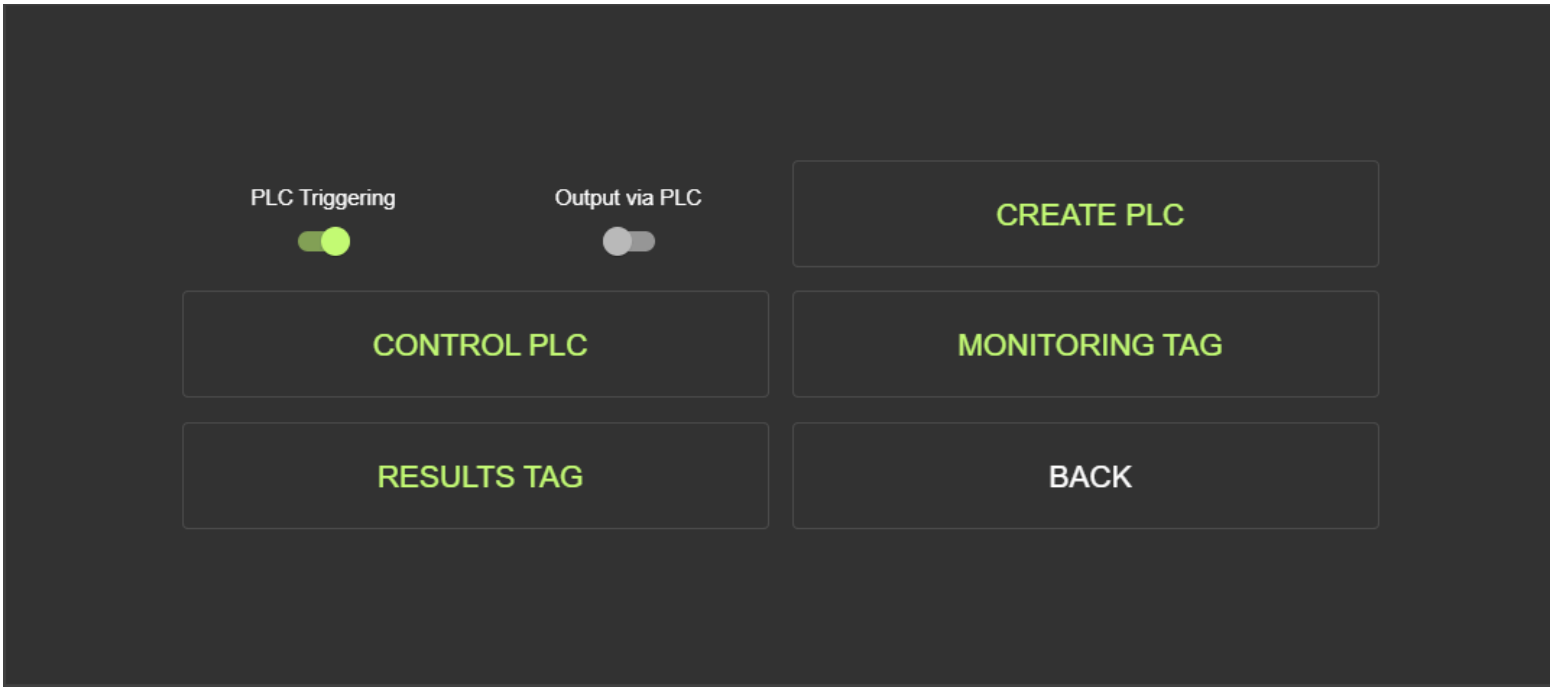
1. Navigate to the **Admin Settings** screen from the left menu on the **Home** screen and click **PLC Config**.
2. Click **Results Tag** on the **PLC Configuration** screen.



3. Select the PLC device from the menu and enter a value for the tag.



4. On the **PLC Configuration** screen, toggle **Output via PLC** to **On** (green).



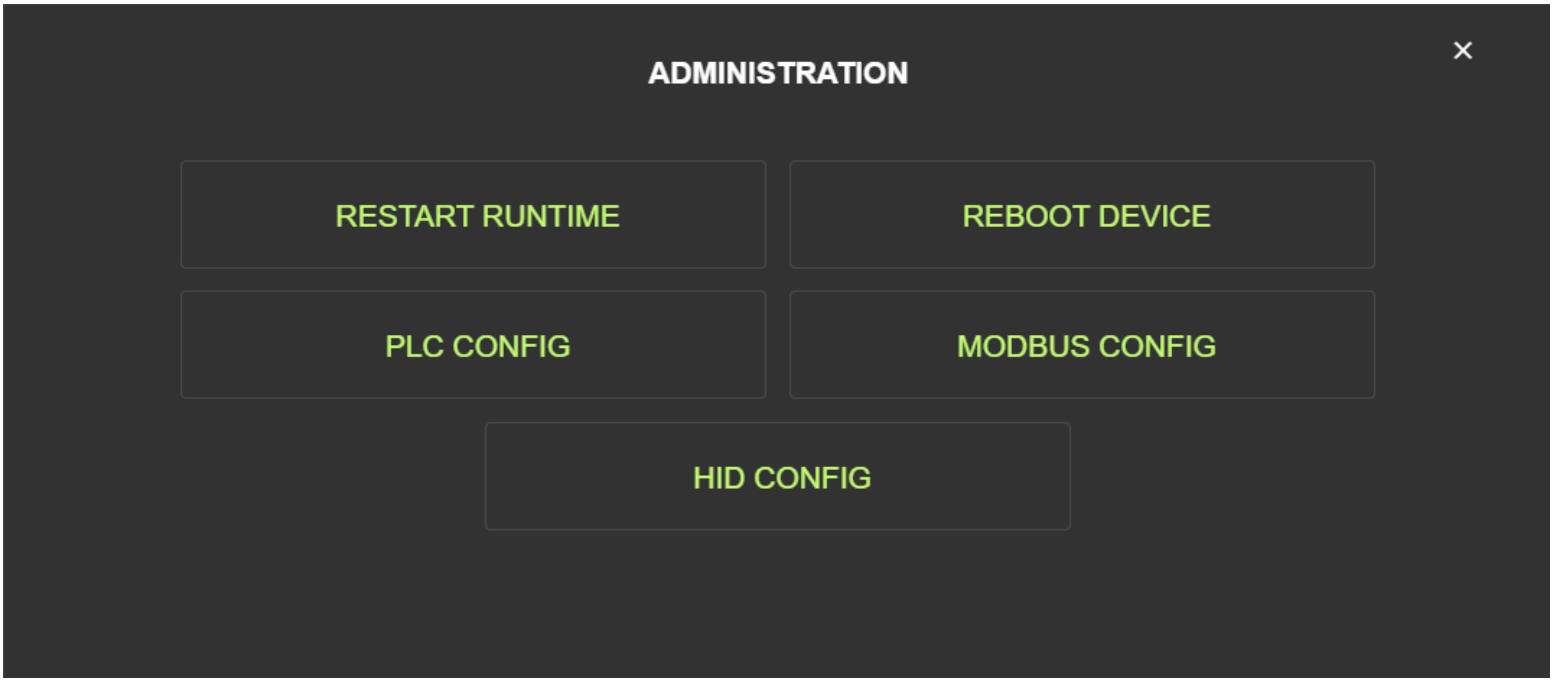
NOTE: The results tag supports a string of up to 4096 characters, including the CR+LF terminator.

5. After the configuration is complete, navigate to the **Control PLC** screen, select the PLC device from the menu, and click **Start** to begin sending results.

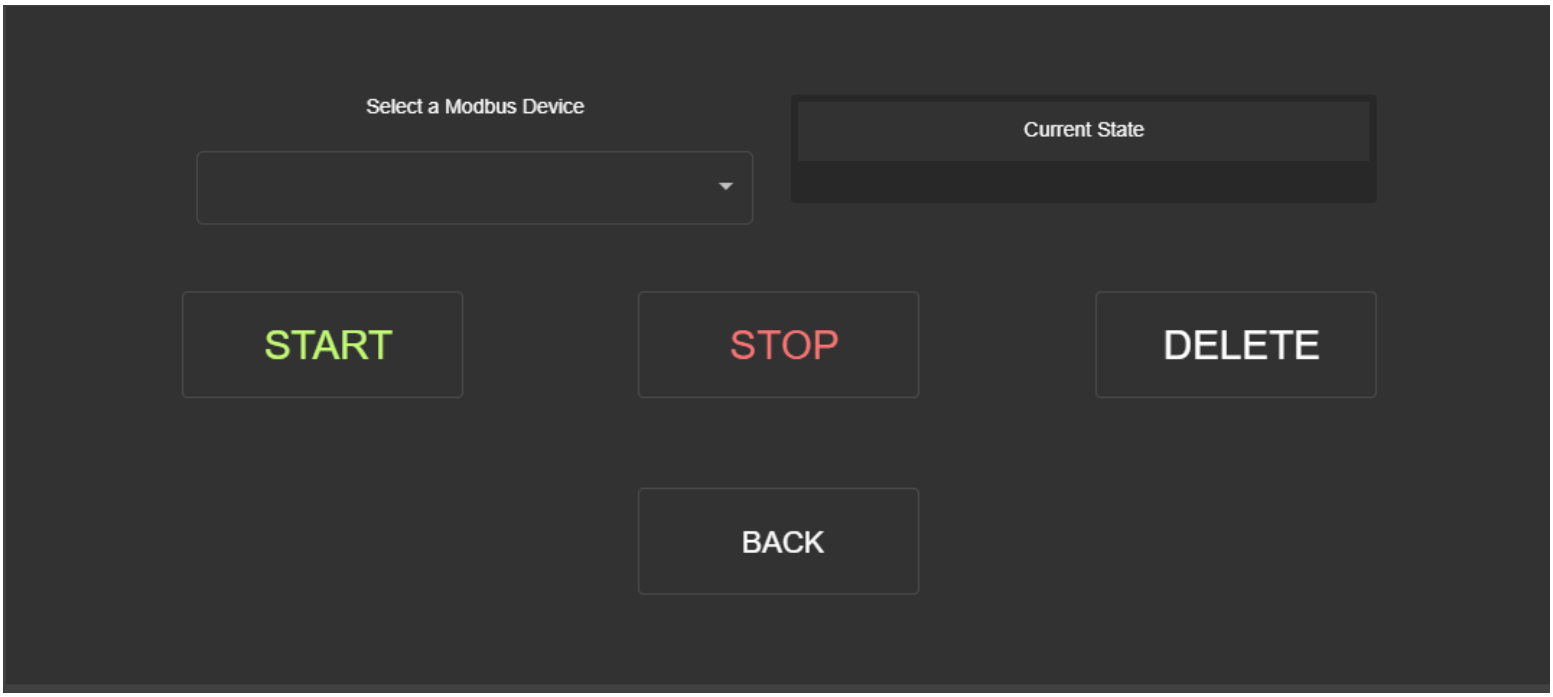
Modbus Configuration

Configure a device to use the Modbus communication standard.

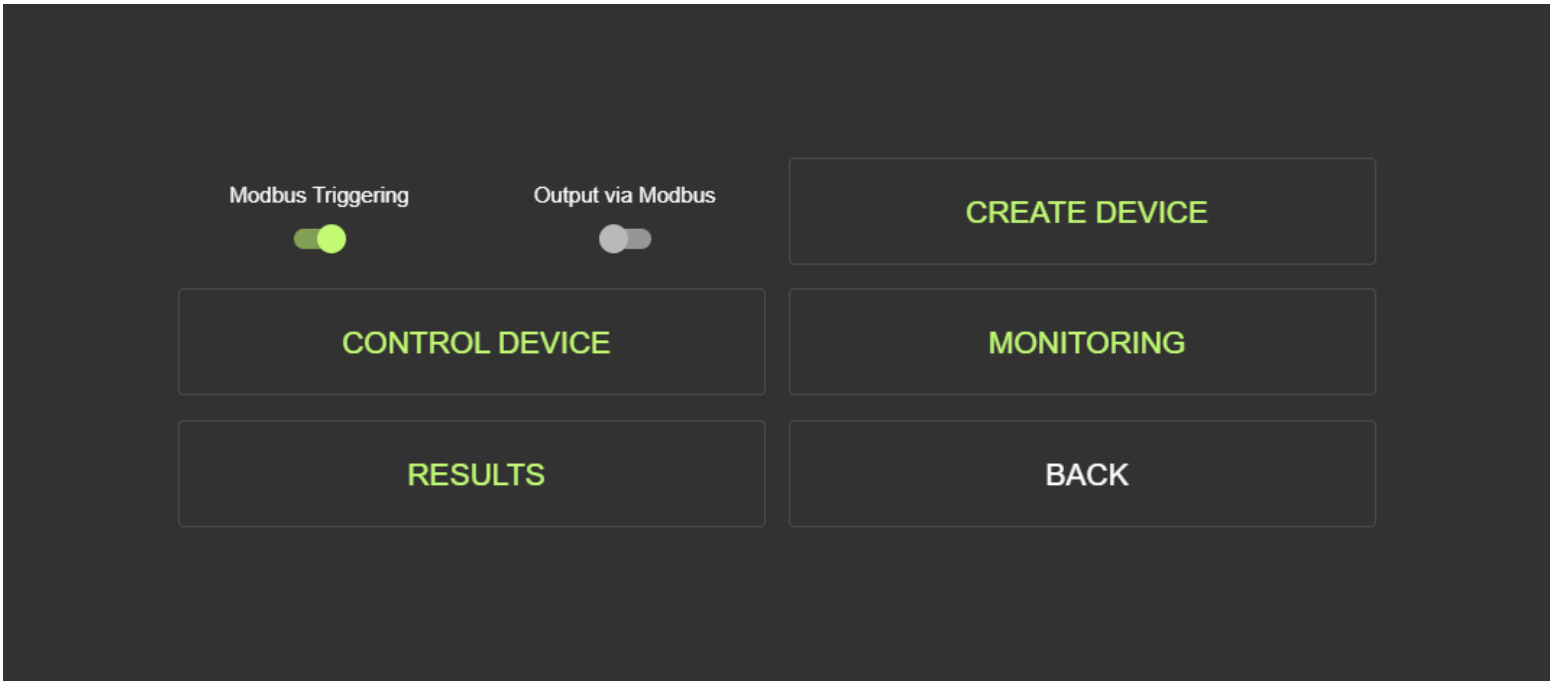
1. Navigate to the **Admin Settings** menu from the **Home** screen and click **Modbus Config**.



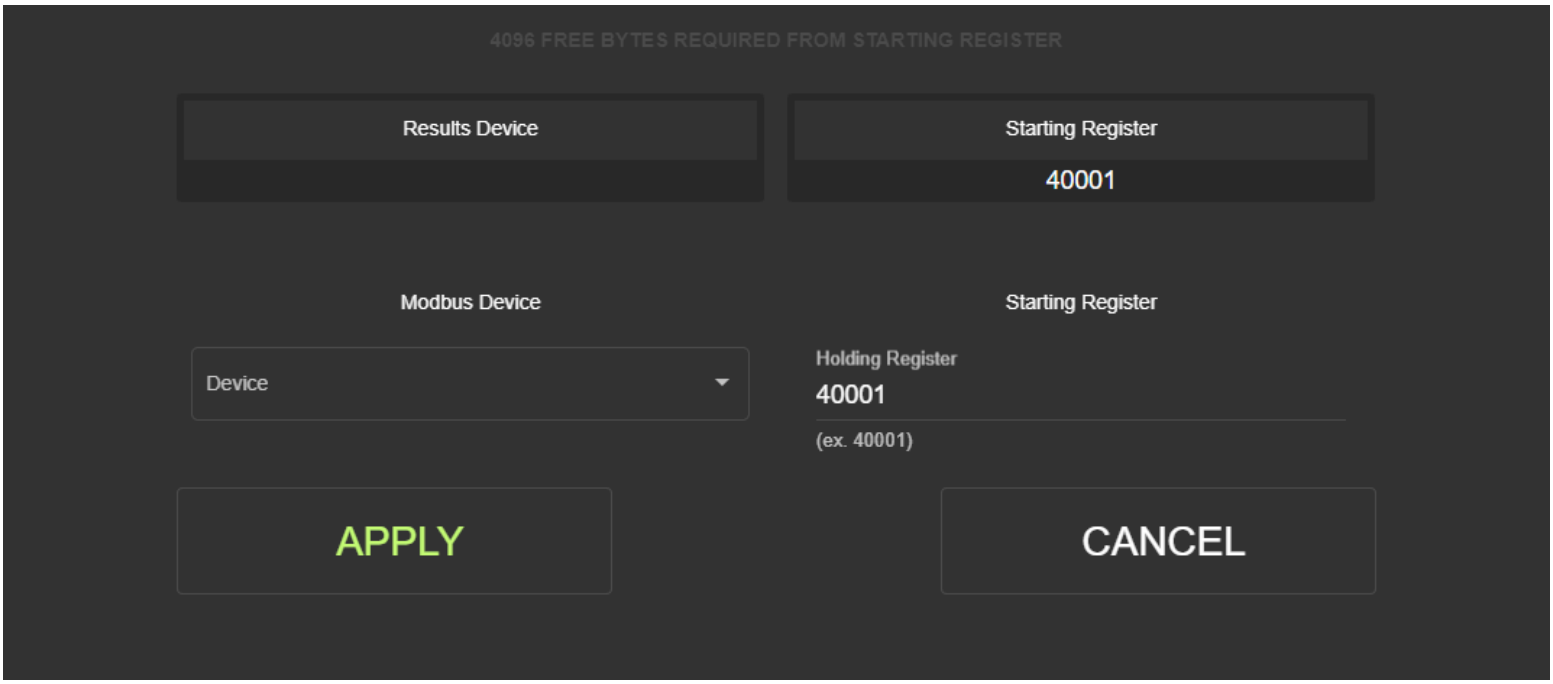
2. Slide the toggle to the right to enable **Modbus Triggering** or **Output to Modbus**.
3. Click **Control Device** to access controls to the Modbus device.
4. Select the Modbus device from the menu and click **Start** to begin controlling the device. Click **Stop** to end the session or **Delete** to remove the device. When you are finished controlling the device, click **Back** to return to the **Admin Settings** menu.



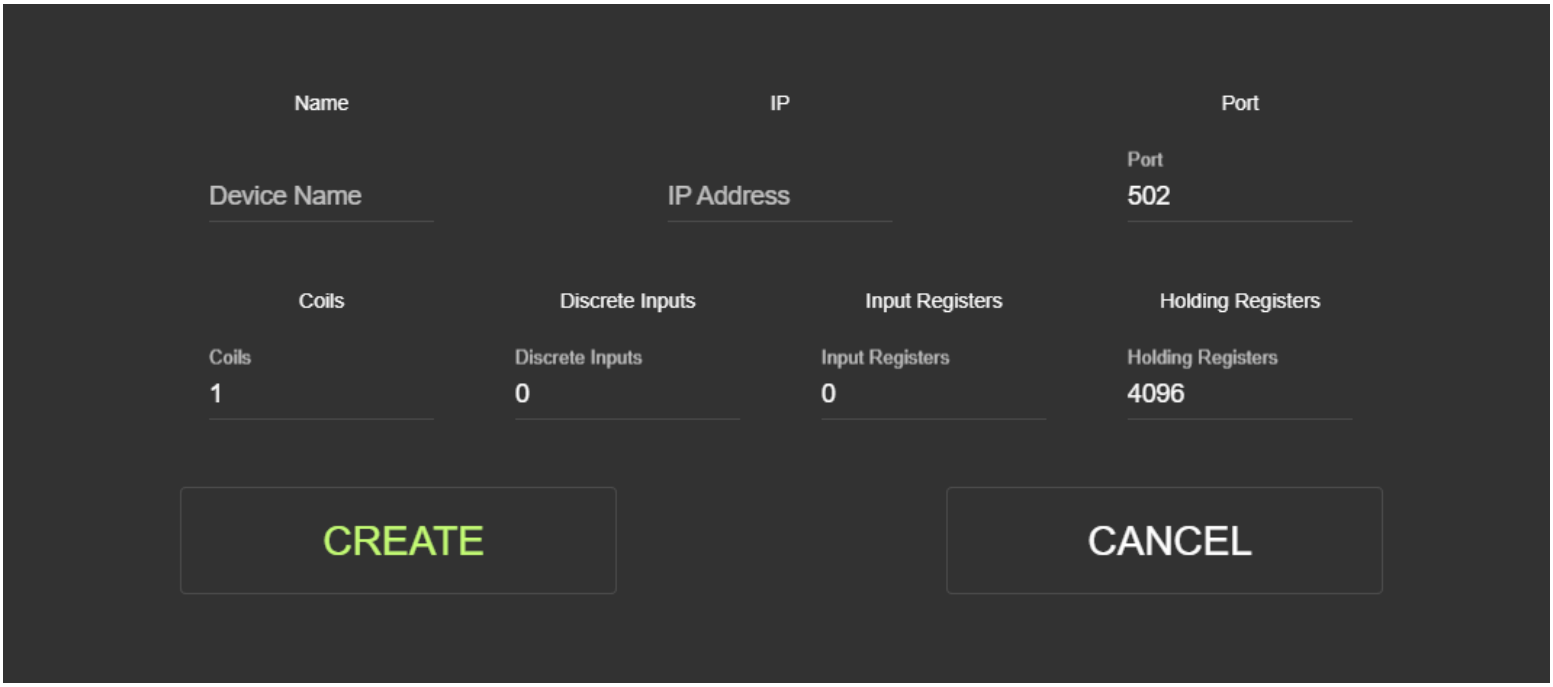
5. Click **Results** to access Modbus results and configure where results are sent to.



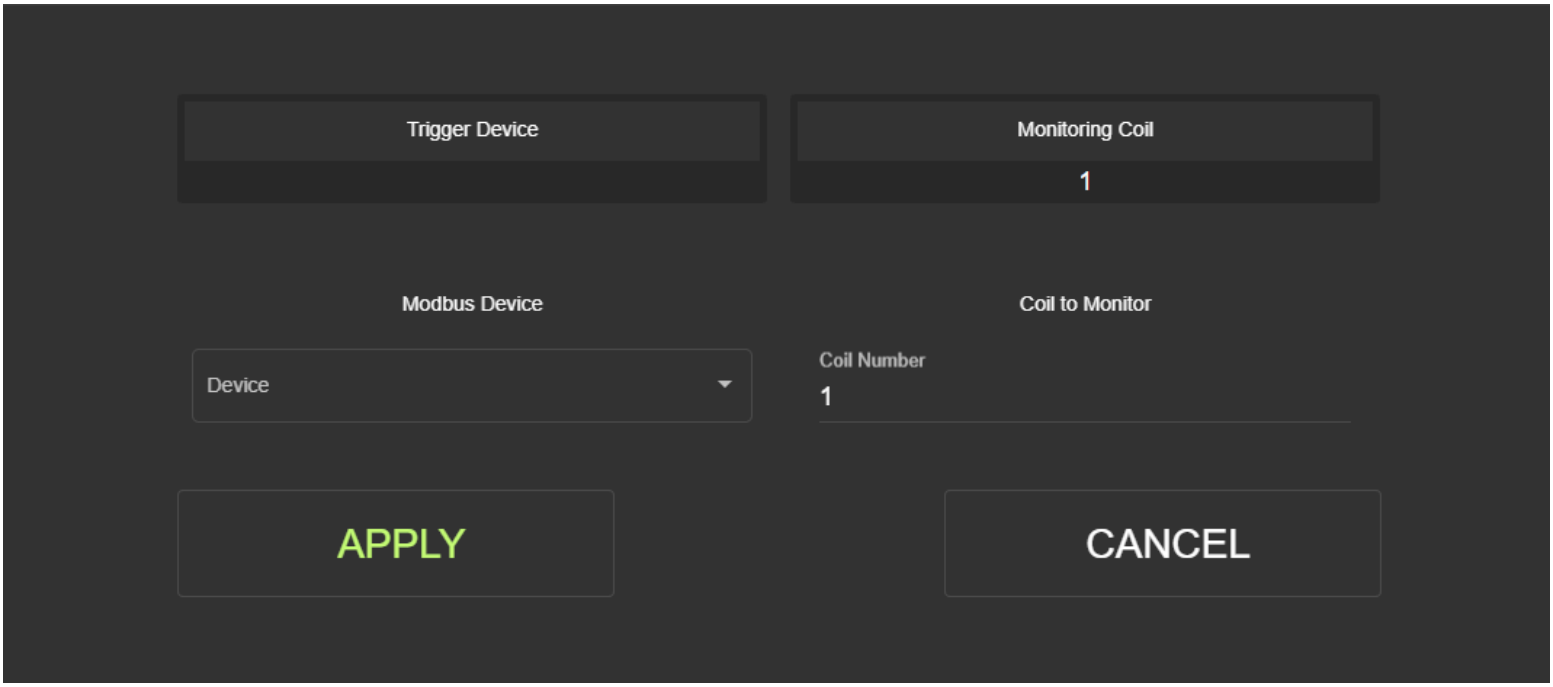
6. Select the Modbus device from the menu and define a **Holding Register**.



7. Click **Confirm** to save changes and return to the **Admin Settings** menu.
8. Click **Create Device** to configure the device information.
9. Provide the required device information and click **Create** to return to the **Admin Settings** menu.



10. Click **Monitoring** to monitor a specific coil on the Modbus device
11. Select a Modbus device from the menu and determine a coil number to monitor.



Accessing the Web HMI

Access the Web HMI by entering the device IP address into a web browser. To obtain the device IP address, select **View Devices** from the menu on the left of the Zebra Aurora Focus application.

When logged in to the Zebra Web HMI, the application presents a dashboard with key hardware metrics such as average inspection per minute, total uptime, temperature, CPU load, communication status, average pass/fail, and resource utilization.

NOTE: Use Google Chrome for optimal performance while using the Web HMI.

Web HMI Top Menu

The Web HMI provides access to the following options using the menu in the top right of the application:

- Device Status Icon - describes the state of the device:
  - Blue - indicates the device is managed and available in Zebra Aurora Focus.
  - Red - indicates a job is actively being edited or deployed in Zebra Aurora Focus.
  - Green - indicates the device is online and running.
- Hostname (FS4072E7) – displays the hostname of the camera.
- Timestamp - displays the current date and time.
- Profile Icon (Operator) - displays the current user role. Click the icon and provide the appropriate credentials to access other roles, such as Administrator.
- Gear Icon (Settings Menu) – provides access to updating the device firmware, selecting a language, restoring factory defaults, setting date and time settings, and backing up the device.
- Notification Icon - displays the current number of unread notifications.
- Localization - displays the current language of the UI.

Web HMI Dashboard

The Web HMI dashboard provides insight into the device's details, such as resource utilization, uptime, temperature, and GPIO communication. Scanning metrics such as total pass/fail count and average inspection per minute are available on the dashboard.

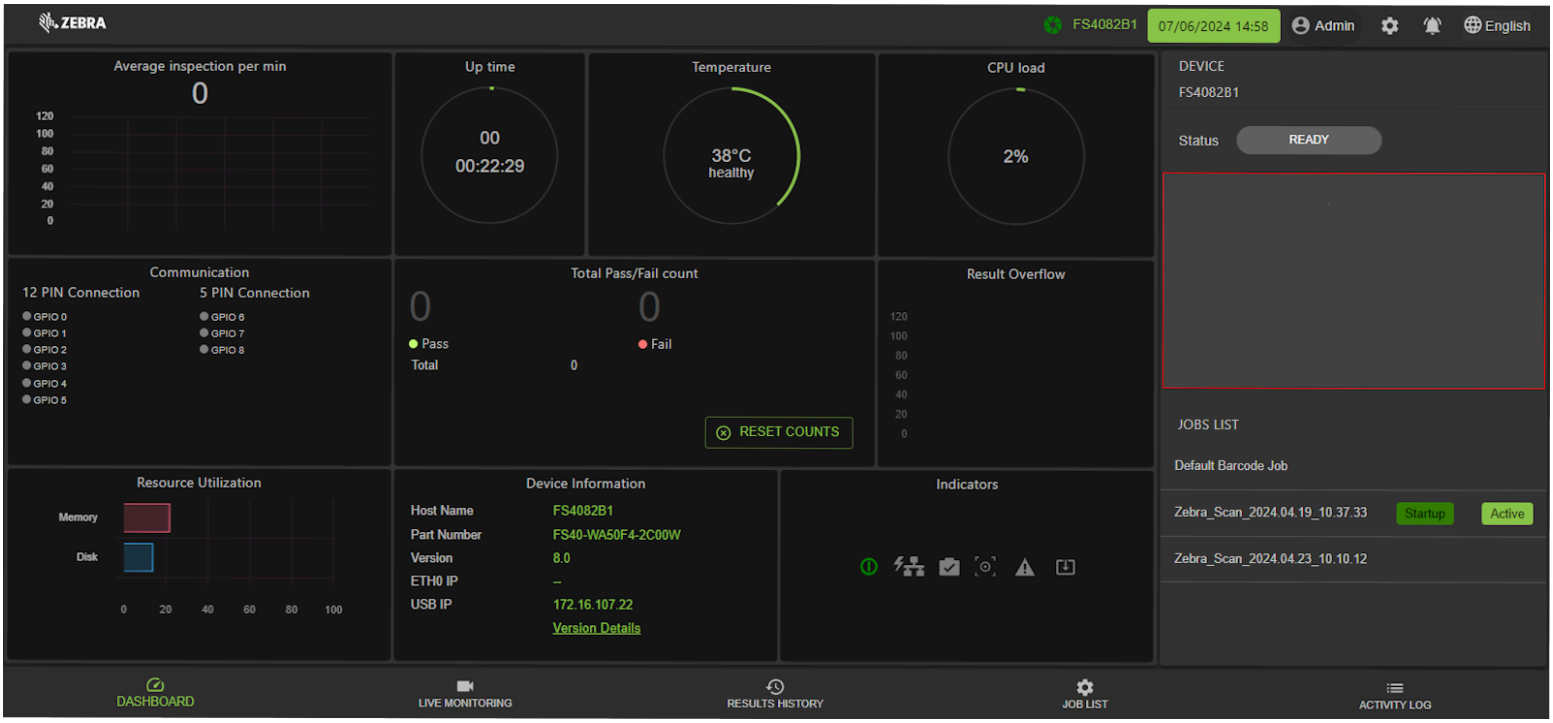
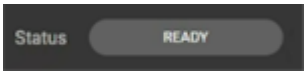


Table 24 Web HMI Dashboard Overview

Section	Description
Average Inspection Per Minute	Displays the number of scans per minute.
Up Time	Displays the total time the device has been running in a given session.
Temperature	Displays the operating temperature of the device.
CPU Load	Displays the current number of processes executed by the CPU.
Communication	Displays the pinout configuration of the GPIO.
Total Pass/Fail	Displays the total number of successful and unsuccessful reads.

Table 24 Web HMI Dashboard Overview (Continued)

Section	Description
Result Overflow	Provides insight into the performance of the setup by displaying the number of results that were not sent out. This could be due to the amount of results data (size/frequency of data) greater than the output interface's throughput. If the system setup is correct, the graph shows 0. If the graph shows an interface greater than 0, an adjustment must be made to preserve the result data. For example, utilize an interface with greater throughput, decrease the amount of output data, or decrease inspection frequency).
Resource Utilization	Provides resource utilization by memory and disk.
Device Information	Provides device information such as Host Name, Part Number, Version, Ethernet IP, and Build number.
Indicators	Power, PoE Network, Device Status, Focus, Warning, Auto Flash, and Firmware Update.
Status	 Ready - the camera is waiting for a trigger, or there are no active jobs.  Busy - the camera is processing one or more triggers.  Editing - Zebra Aurora Focus is editing a job.

Live Monitoring with the Web HMI

The Live Monitoring feature allows users to view decode results as they occur in real time by clicking the Live Monitoring tab at the bottom of the interface.

The Web HMI also provides the capability to update the device firmware by selecting the settings icon in the top right corner of the application.

Table 25 Live Monitoring with the Web HMI

Section	Description
Live Monitoring View	Provides real-time monitoring of the camera's view while processing jobs.
Recent Inspections	Lists all recent jobs and their status, the toolset used for the job, and total job time.

Viewing Result History

The Results History tab provides job result information, such as status, execution time, and date. Trigger information, including total triggers, total pass/fail, and missed triggers, are available on the Job panel on the right.

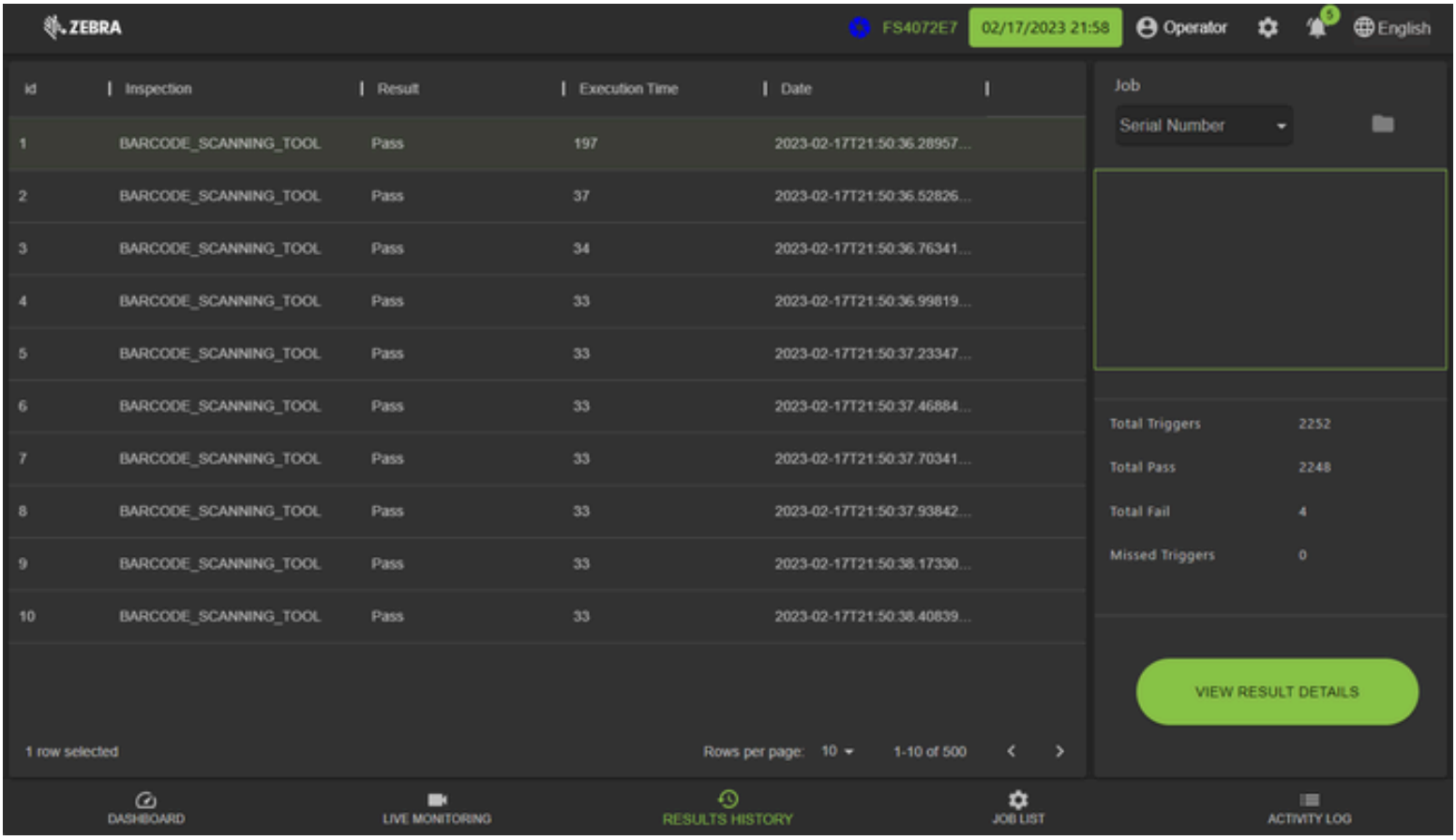


Table 26 Viewing Result History

Section	Description
Result History Table	Displays decode result information such as ID, status, execution time, and date.
Job Statistics	Select a job from the menu to view total triggers, pass/fail, and missed trigger details.

Viewing the Job List

The job list provides information on the currently active jobs that can be run on the device. Additional details on the right of the list include filtering by active and inactive jobs, the trigger mode of the specified job, its slot number, and its description.

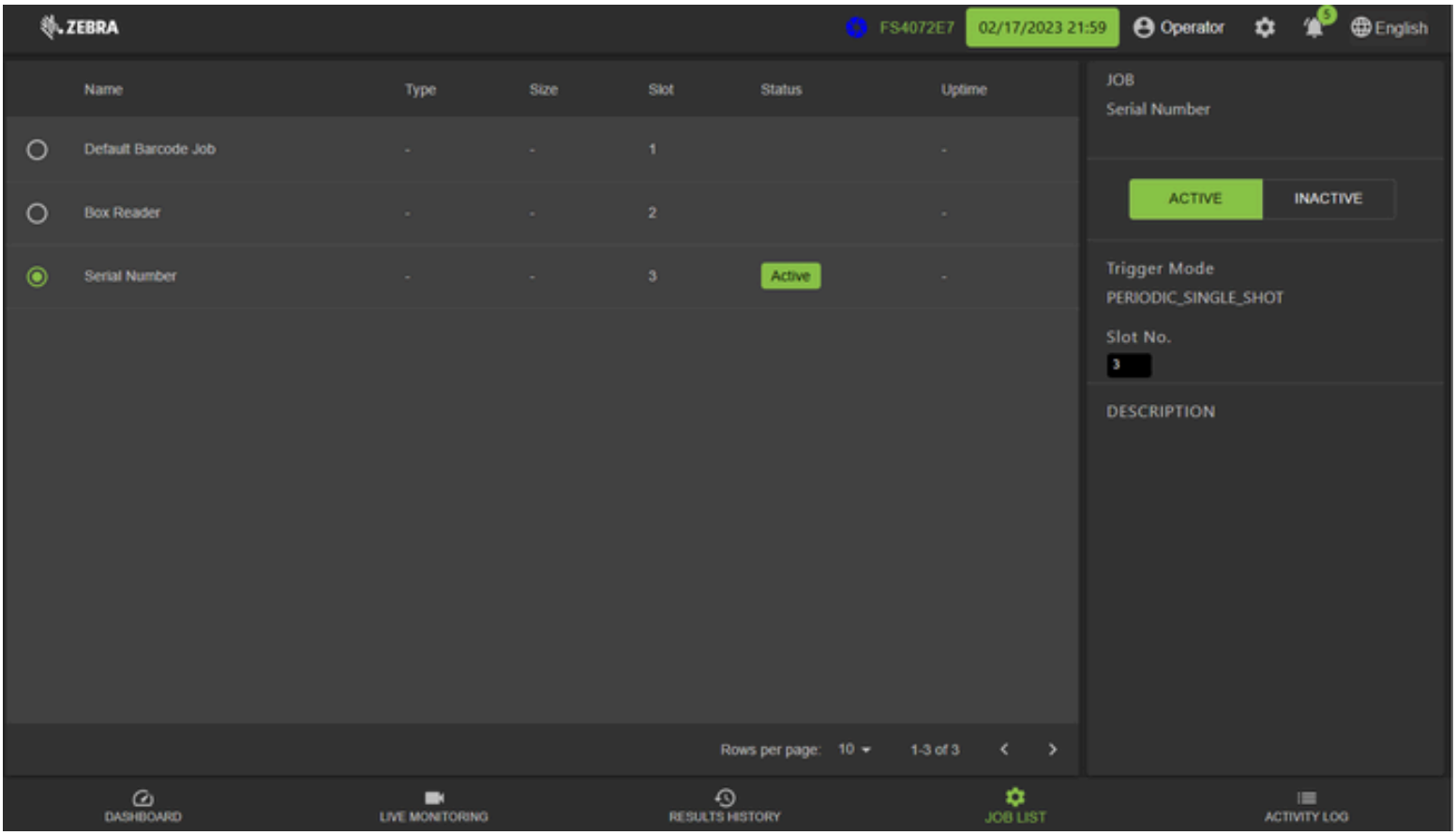


Table 27 Job List

Section	Description
Job List	Displays each job's statistics, such as type, size, slot status, and uptime.
Current Job	Provides additional details on a specific job, including its Trigger Mode and description.

Viewing the Activity Log

The Activity Log provides information on specific actions taken by the device, a live view of the device state, and a list of all currently active jobs and jobs deployed upon startup.

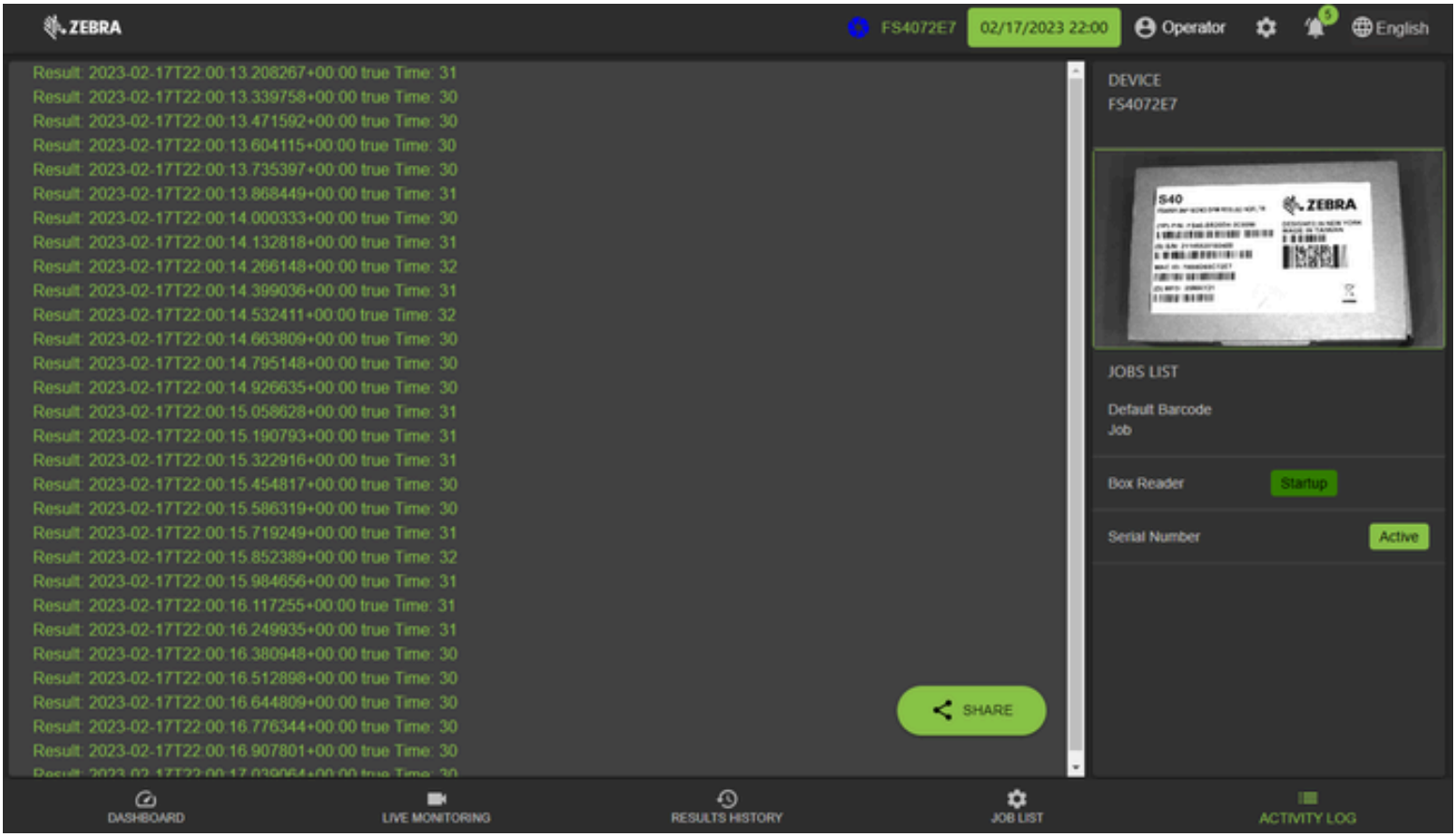


Table 28 Viewing the Activity Log

Setting	Description
Activity Log	Lists all of the device's recent activities.
Live View	Real-time view from the camera of the device.
Job List	Displays all jobs sorted by jobs deployed upon startup and currently active jobs.

Updating Device Firmware

The device must be connected to a PC using a USB or Ethernet cable or from Ethernet to a router (common network with a host PC) to update the firmware. The USB connection provides power and communications to the device.

NOTE: Conduct a Factory Reset after applying a firmware update to a Connectivity Gateway license-enabled device.

Requirements:

- Web browser (Google Chrome, Mozilla Firefox, or equivalent)
- Firmware file on the local PC

For additional details on setting up the device, refer to the FS/VS Smart Camera Product Reference Guide.

USB-A to USB-C Hardware Setup

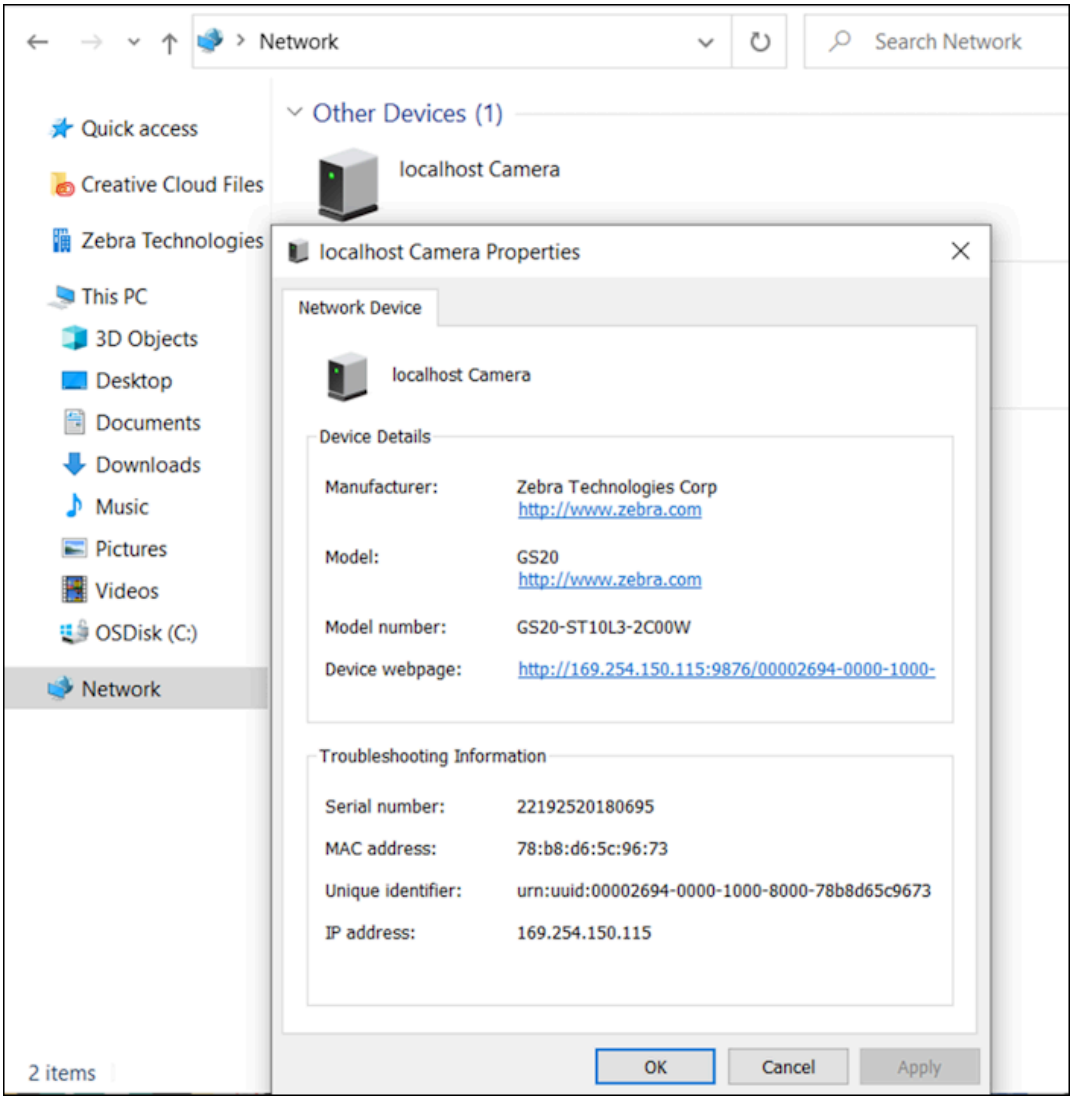
Required hardware:

- Windows 10 or higher PC with USB-A or USB-C port
- USB-A to USB-C cable, Zebra P/N: CBL-USB0200-USA00 or
- USB-C to USB-C cable, Zebra P/N: CBL-USB0200-USC00

Firmware Update

This section outlines how to perform a firmware update on the device.

1. Open **File Explorer** on your host Windows desktop PC.
2. On the left window pane, scroll down and select **Network** to view devices on your network and identify a device.
3. Right-click on the device and select **Properties** to obtain the device's IP address information.

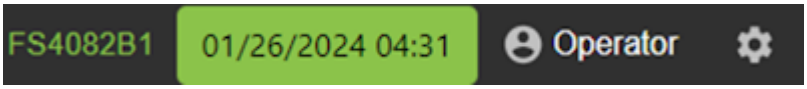


4. Enter the IP address (or hostname) into your browser.

NOTE: This example refers to the hostname as localhost in the top left corner of the Properties window.

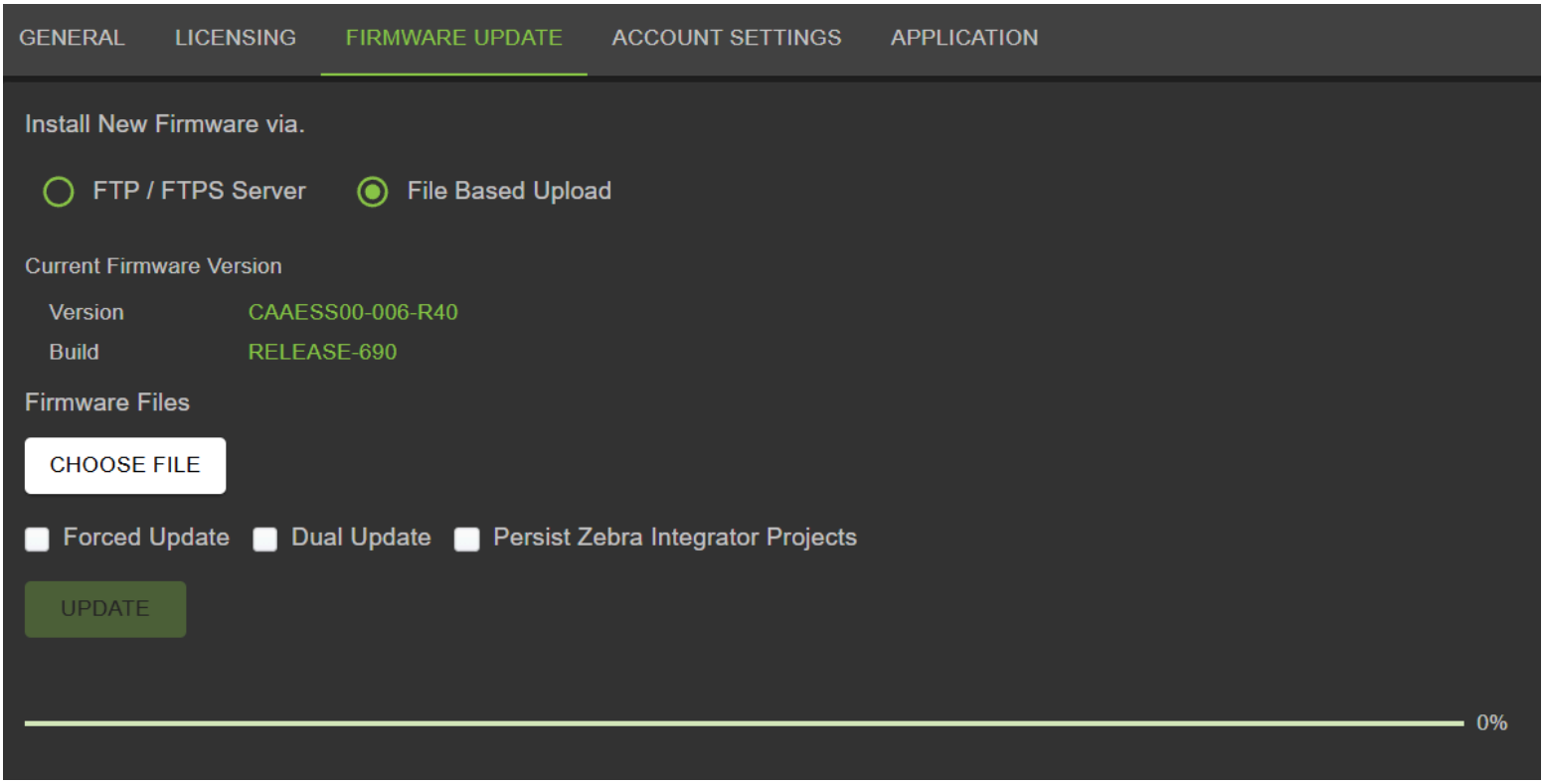
5. View the Zebra Web HMI and observe the build number RELEASE-xx (where xx is the build number) in the **Device Information** section of the dashboard.

6. Click **Operator** to launch the login window.

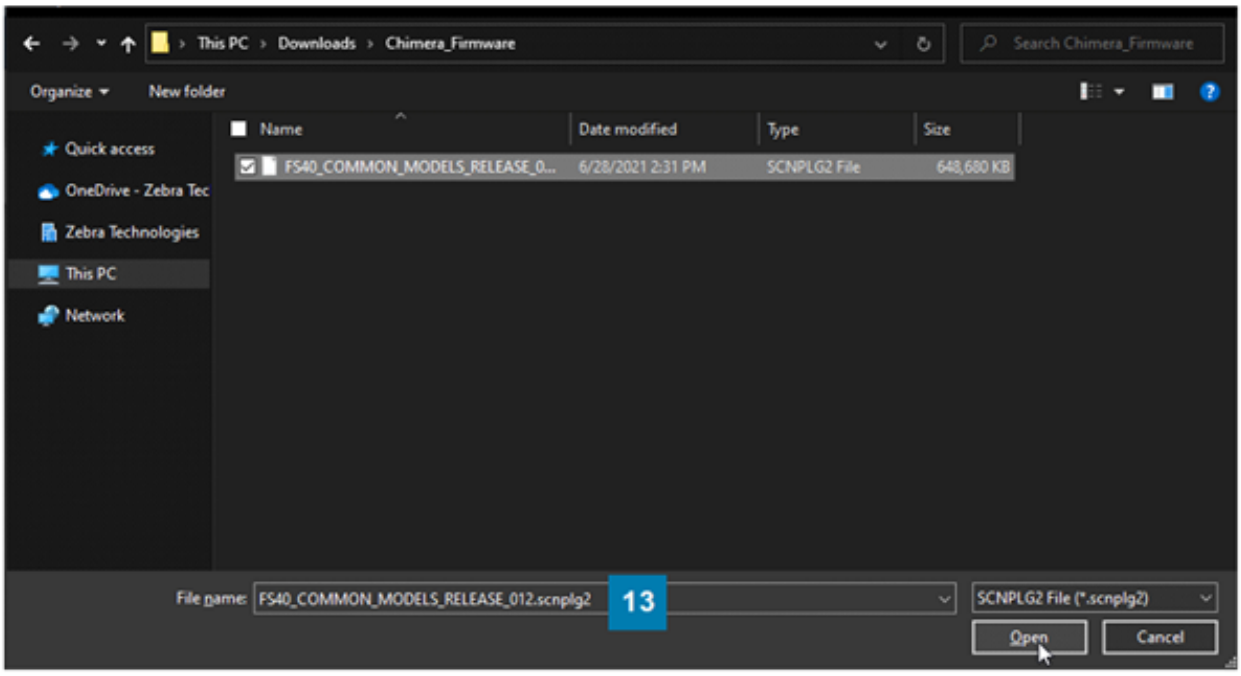


7. Enter the following login credentials:
   a) User ID: admin
   b) Password: admin
8. Click **Login**.
9. Observe that **Operator** now displays as **Admin**.

10. Click .
11. Click the **Firmware Update** tab.
12. Click **Choose File**, navigate to the location of the file stored on the Local PC, and select the latest firmware file.



13. Click **Open**.



14. Select either **Forced Update** or **Dual Update**.

NOTE: Forced Update loads the device in cases where it is identical to the current firmware on the device. Dual Update loads the device firmware and updates the backup partition. This option typically takes longer (twice as long) than a forced update.

15. Click **Update**.

    The device LED blinks red and the upload progress displays on the screen. The device reboots after the upload is complete.

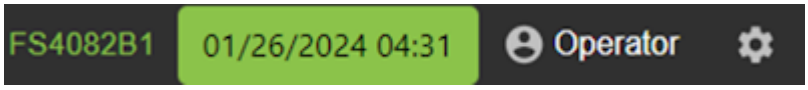
16. Refresh the browser window and view the build number in the device information field to confirm that it has been updated from the previous version.

Performing a Factory Reset

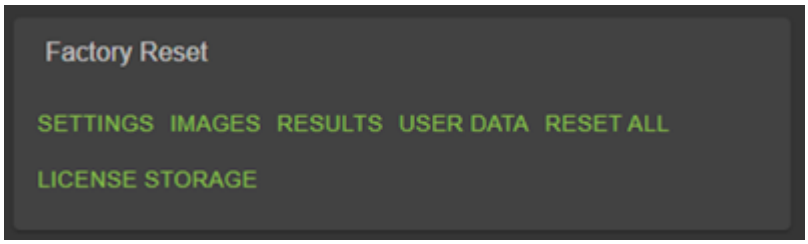
A factory reset deletes all created jobs on the camera.

NOTE: A factory reset deletes all created jobs on the camera. It is critical to save all jobs and user-specific information before performing the factory reset.

1. Click **Operator** to launch the login window.



2. Enter the following credentials:
   a) User ID: admin
   b) Password: admin
3. Click **Login**.
4. Click ⚙.
5. On the **General** tab, click **Reset All** in the **Factory Reset** field, and wait for the device to reboot (approximately 45-60 seconds).



Using Fixed Industrial Scanning Tools

Use Fixed Industral Scanning symbologies and tools and set image banks and format data.

Editing and Deploying FS Jobs

To set up a Fixed Scanning (FS) job, set decoder parameters, symbologies, OCR settings, code quality metrics, and data formatting rules.

Begin by configuring the decode parameters and selecting the appropriate set of symbologies. A symbology is chosen by clicking Symbologies Tab. Next, select specific symbologies to be deployed by clicking the corresponding checkboxes.

Once the job is in progress, monitor the **Image Viewer** and the **Filmstrip** controls to view job progress. The **Image Viewer** contains a status bar that displays the result and runtime. In the FS editor, the status bar displays the decode time, decoded value, PPM, and the type of symbology decoded.

View Results provides additional data on the decode and displays the results for each job instance.

Using the Job Toolstrip

The Jobs Toolstrip provides access to available devices or emulators to apply the job to, the barcode type, and the detected power source. Save the configuration by clicking the disk icon.

Table 29 Jobs Toolstrip

Setting	Description
Device/Emulator Selector	Select the device or emulator for the job to be deployed to.
Barcode Type	Select the applicable barcode type for the job.
Power Source	Displays the power source type that the device is connected to.
Save Options	Saves the job configuration.

Fixed Industrial Scanner Settings

Configure Fixed Industrial Scanning settings such as timeout, adaptive ROI search, barcode string match, or a no-read string, depending on your use case.

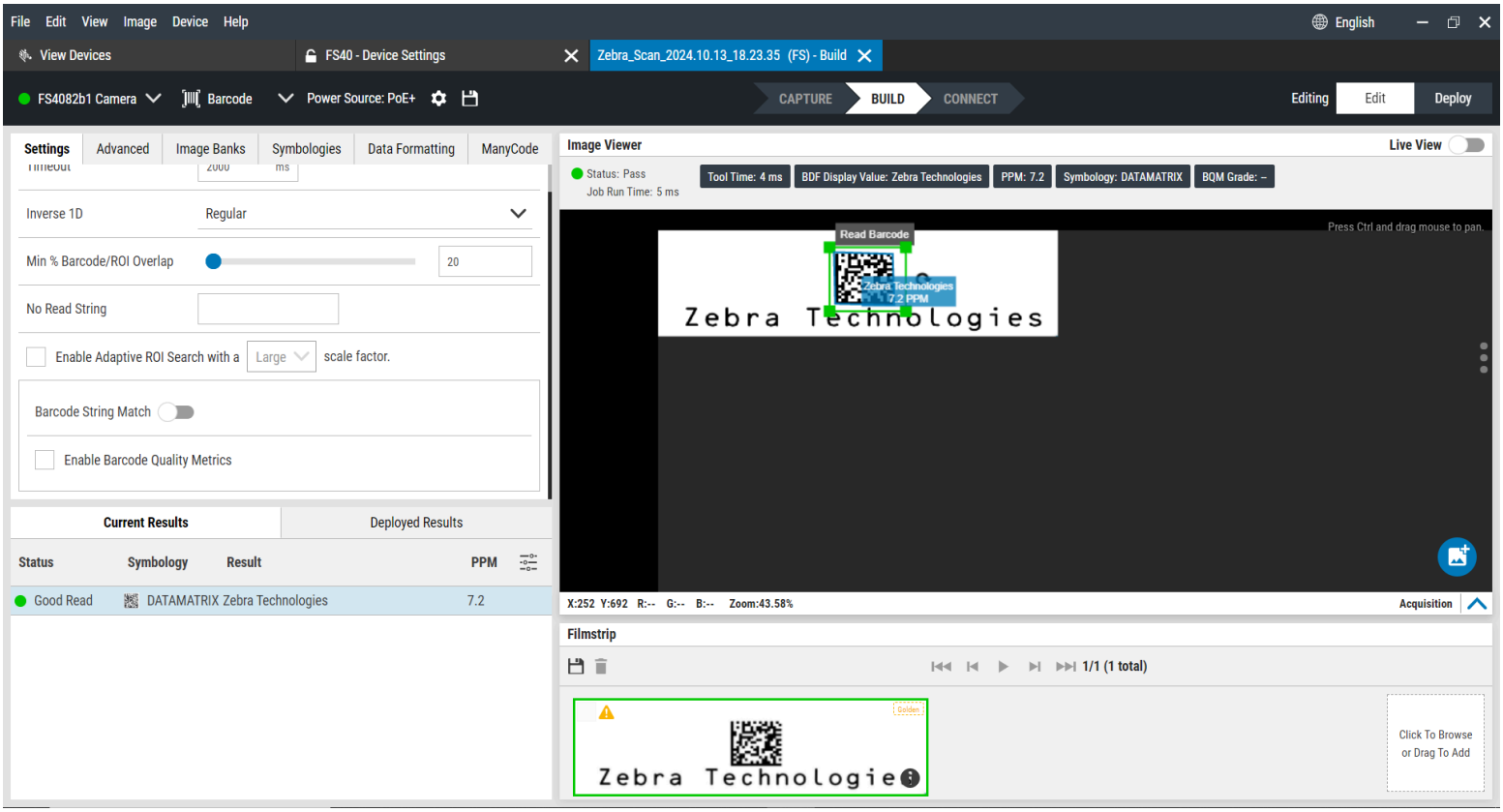


Table 30 Fixed Industrial Scanning Tool Settings

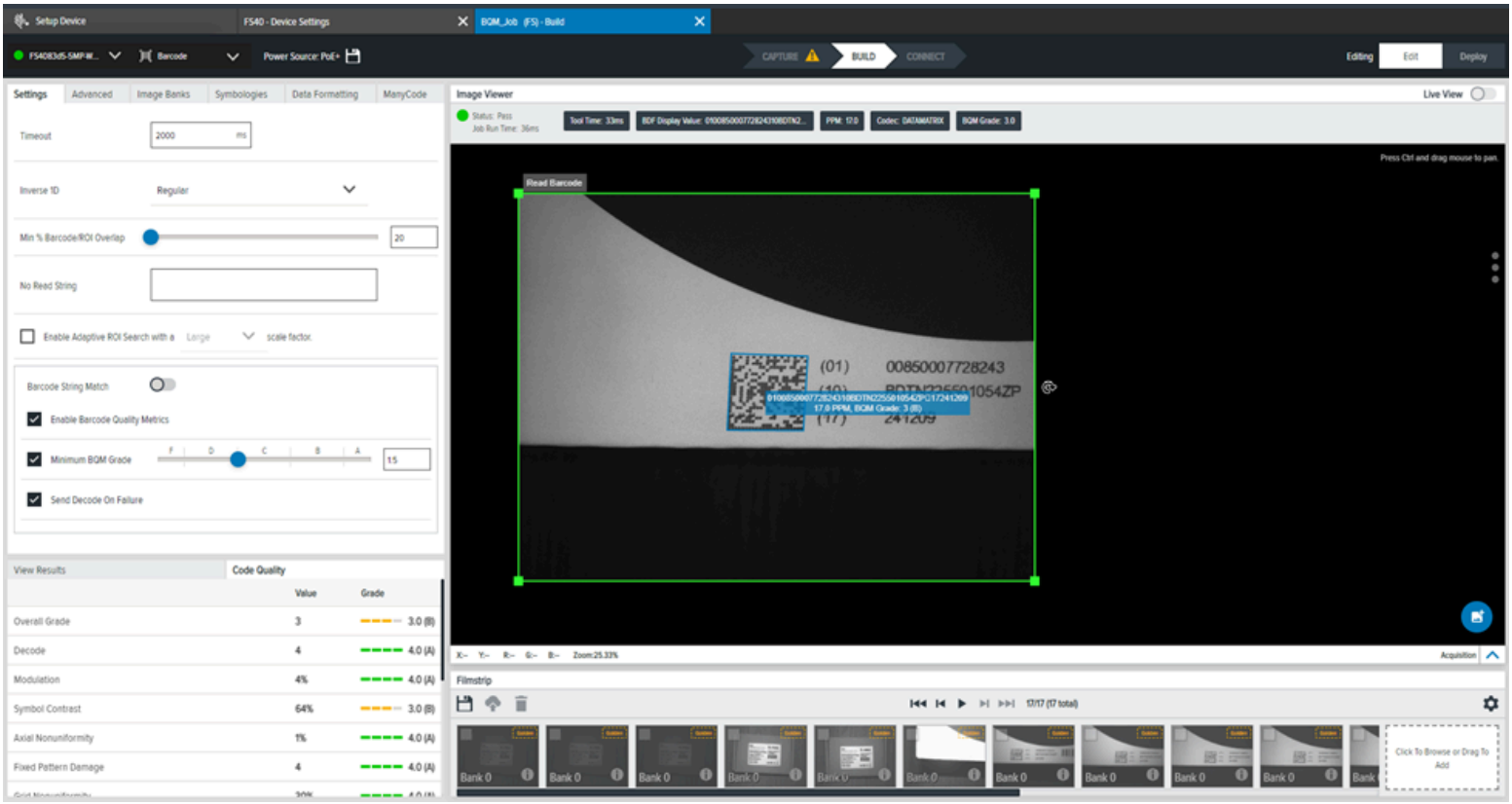
Setting	Description
Timeout	The amount of time (ms) the tool should stop its process of searching for barcodes. **NOTE:** For high-speed barcode-reading applications (for example, a fast conveyor belt or high-speed turntable), set this value as low as possible, just above the average time to read a barcode. Deploy the application and obtain real-world data on the decode speed in your specific use case (for example, a range of 5-15ms). Set this to a value just above that speed (for example, 20ms). This allows the application to capture frames from high-speed subject images, increasing the odds of a good read.
Inverse 1D	• Regular - reads dark foreground/light background barcodes. • Inverse Only - reads light foreground/dark background codes only. • Inverse Autodetect - reads both barcode types defined above.

Table 30 Fixed Industrial Scanning Tool Settings (Continued)

Setting	Description
Minimum Percentage Barcode/ROI Overlap	Define the minimum percentage a barcode needs to be contained within the ROI to be read. • For 1D barcodes, this applies only to the horizontal (X) dimension. • For 2D barcodes, this applies to horizontal and vertical (X & Y) dimensions.
No Read String	Define the data that should be output if no barcode is read (in place of barcode data in a successful read, as opposed to no output at all).
Enable Adaptive ROI Search	Enabling this parameter allows the underlying barcode scanning algorithm to alter (shrink and reposition) the area of the ROI for subsequent reads based on the presence of past-read barcodes. This provides quicker and better-performing barcode reads. The small/ medium/large drop-down settings alter how the adapted ROI component is generated.
Barcode String Match	Enable this setting to define the pass/fail criteria based on the contents of the decoded barcode. If the Match String matches and the substring within the barcode is, the barcode tool passes.
Minimum BQM Grade	Defines a minimum BQM threshold (produces a barcode-based pass/fail if it does not meet this minimum BQM score).
Send Decode on Failure	If a barcode is read but does not meet the **String Match** criteria, enabling this setting allows the barcode data to be output while the tool fails.

Using BQM for Fixed Scanning Jobs

Barcode Quality Metrics are enabled on the **Settings** tab for FIS jobs.

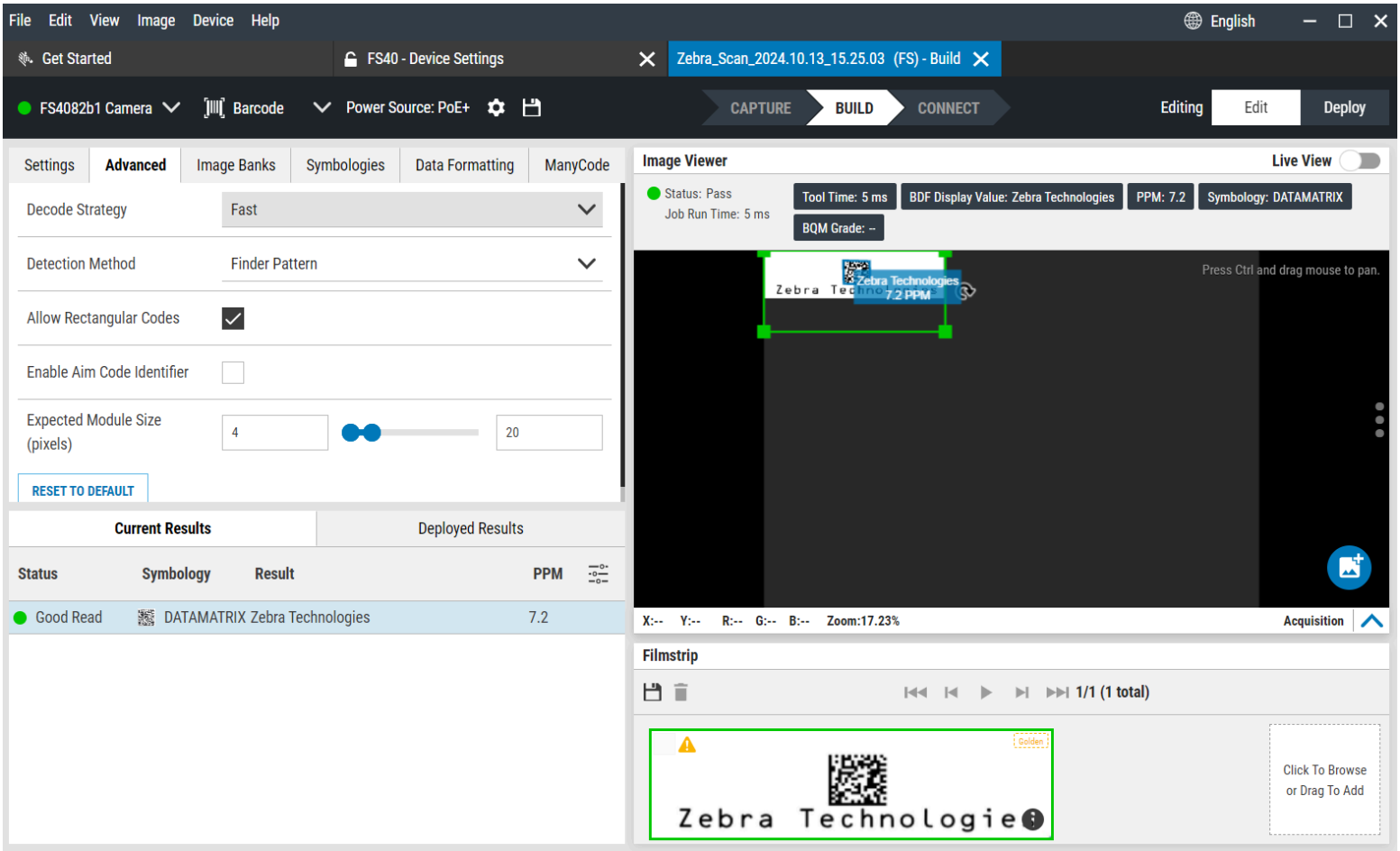


After the **Minimum BQM Grade** is set, observe the additional metrics displayed on the **Code Quality** tab in the **Results** section.

NOTE: BQM is not supported for QR codes.

Advanced FIS Tools

The Advanced tab provides access to additional Fixed Industrial Scanning tool settings.



NOTE: All Advanced detection methods apply to both 1D and 2D barcodes.

Table 31 Advanced Fixed Industrial Scanning Tool Settings

Setting	Description
Decode Strategy	Changes the variables to alter the balance of speed and performance. • Fast – This strategy is designed to be fast but may not find a barcode if it is hard to find. • Moderate - This strategy is helpful for most use cases with a moderate increase in analysis times. • Exhaustive – This strategy does an exhaustive search to identify hard to find barcodes. This method can sometimes lengthen analysis time significantly.

Table 31 Advanced Fixed Industrial Scanning Tool Settings (Continued)

Setting	Description
Detection Method	Determine the preferred method for detecting the data: • Uniform—uses a splatter pattern to analyze the whole image uniformly. This is helpful for 1D and 2D barcodes with a lot of noise. If you are looking for a random pattern, favor the center of the image slightly. • Quiet Zone - specialized to find 2D barcodes fast and helpful when over 8 pixels of buffer zone around the barcode is expected. This setting may not be helpful for use cases when the buffer area is less than 4 pixels or 1D barcodes. • Finder Pattern - uses a contrast map that finds patterns with little or no noise. This method is applicable to 1D and 2D barcodes.
Allow Rectangular Codes	Allows the reading of 2D rectangular barcodes in addition to square 2D barcodes.
Expected Module Size (Pixels)	Define the range (pixels) you expect a module to be in given barcodes to help increase read performance. A module is the smallest divisible unit of a barcode; for 1D, typically the width of a single thin line; for 2D, the pixel size. This setting serves as a general guidance to increase performance, but is not a strict threshold.

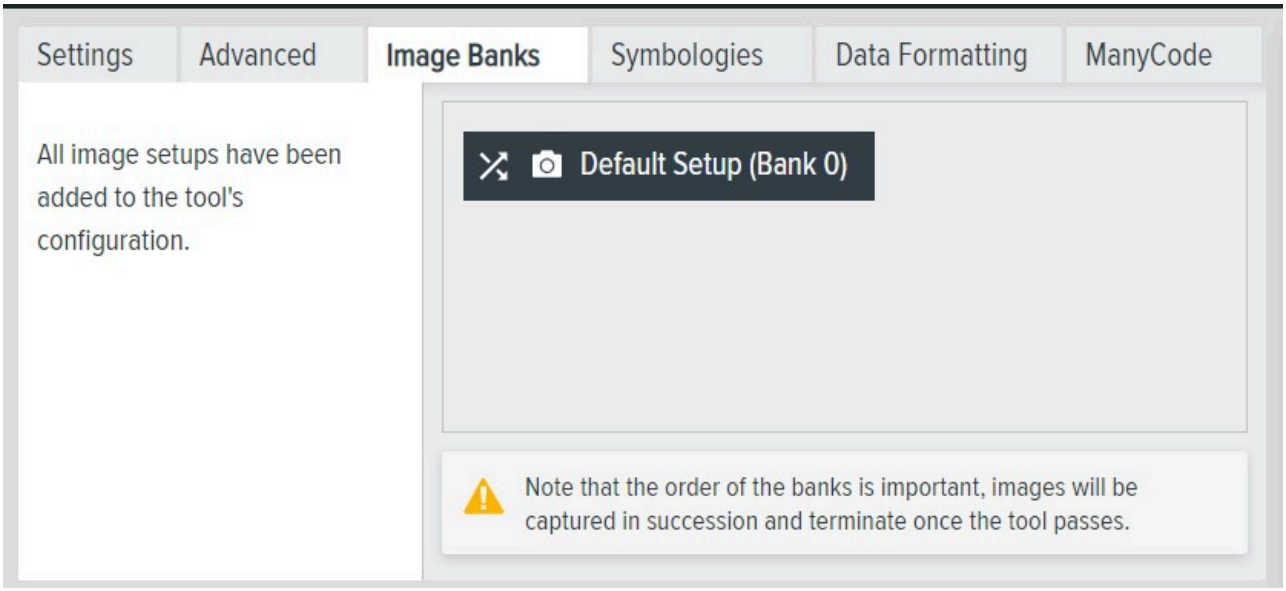
Image Banks

All available image banks display on the left side of the table. The right side of the table provides a top-down sequence of image capture acquisition banks that are used for an **ImagePerfect** job.

NOTE: Image banks are taken in sequential order, starting from the top. After a barcode is successfully decoded, the sequence stops. For example, if there are four banks and the first two decode attempts fail, but the third one decodes successfully, the fourth acquisition attempt does not occur.

ImagePerfect provides multiple banks for a single job run and is commonly used for applications that require different focus levels in the same Field of View (FoV).

For example, two barcodes in the same FoV, one 12” away from the camera and the other 36” away from the camera. These barcodes require two different focus levels to be clear enough to read. ImagePerfect multiple-acquisition-bank functionality makes this possible.


Settings
Advanced
Image Banks
Symbologies
Data Formatting
ManyCode
All image setups have been added to the tool's configuration.
Default Setup (Bank 0)
Note that the order of the banks is important, images will be captured in succession and terminate once the tool passes.

Symbologies

Configure the settings of each symbology based on your use case.

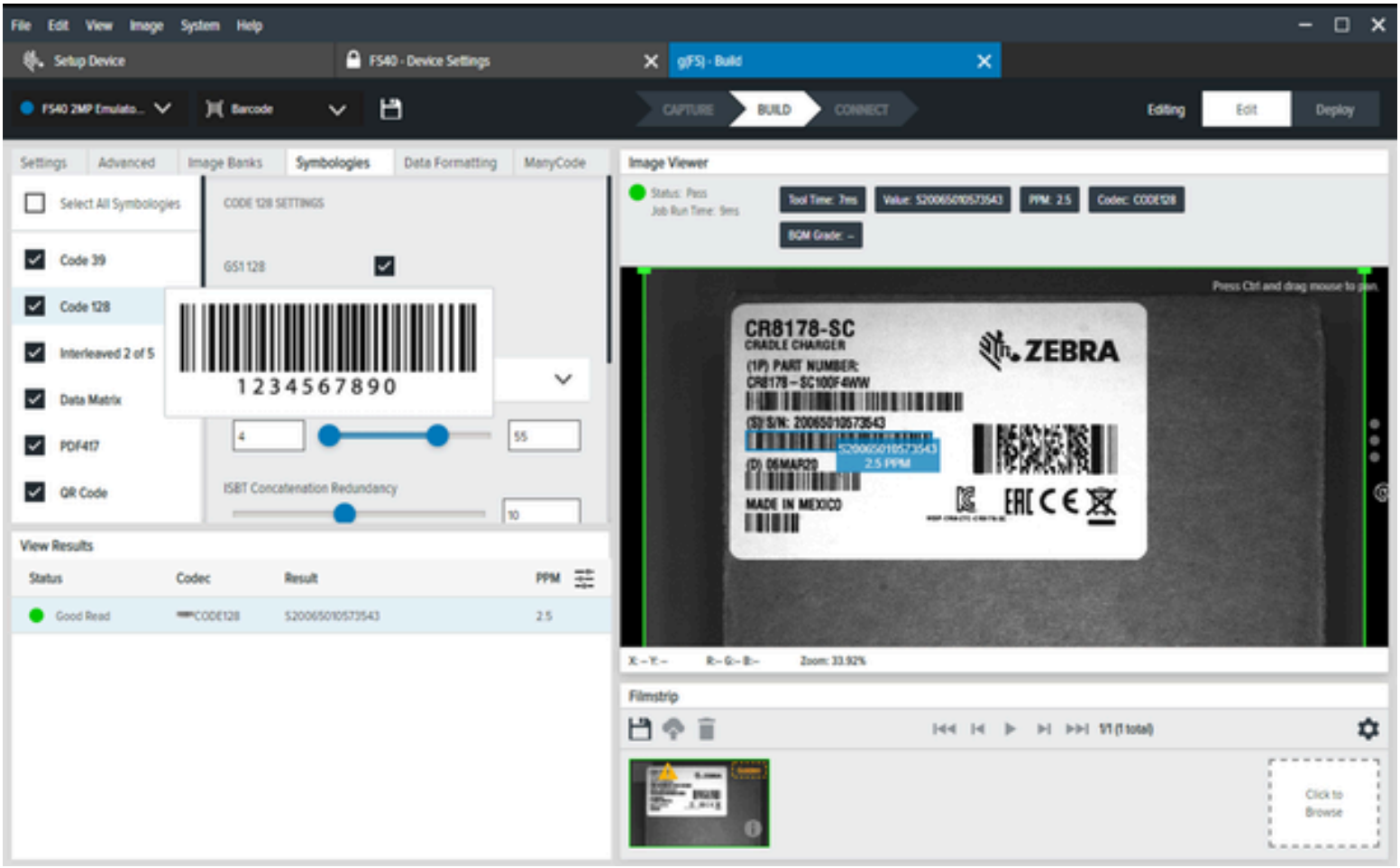


Table 32 Symbologies

Setting	Description
Code 39	Configurable settings include: • Trioptic - Trioptic Code 39 is a variant of Code 39 used in the marking of computer tape cartridges. Trioptic Code 39 symbols always contain six characters. • Full ASCII Conversion - Code 39 Full ASCII is a variant of Code 39 that pairs characters to encode the full ASCII character set. **NOTE:** Trioptic Code 39 and Code 39 Full ASCII cannot be enabled simultaneously. • Set lengths for Code 39 to any length, one or two discrete lengths, or lengths within a specific range. **NOTE:** If Code 39 Full ASCII is enabled, Length Within a Range or Any Length are the preferred options. • Check Digit Verification - enable to check the integrity of all Code 39 symbols to verify that the data complies with the specified check digit algorithm. Only Code 39 symbols that include a modulo 43 check digit are decoded. **NOTE:** Check Digit Verification is useful when the Code 39 symbols contain a Modulo 43 check digit. • Convert to Code 32 - Code 32 is a variant of Code 39 used by the Italian pharmaceutical industry. **NOTE:** Code 39 must be enabled for this parameter to function.

Table 32 Symbologies (Continued)

Setting	Description

Code 128	Configurable settings include:
	• Enable GS1128
	• ISBT 128 - ISBT 128 is a variant of Code 128 used in the blood bank industry.
	NOTE: If necessary, the host must perform the concatenation of the ISBT data.
	• Set lengths for Code 128 to any length, one or two discrete lengths, or lengths within a specific range.
	NOTE: When setting lengths, enter a leading zero for single-digit numbers.
	• ISBT Concatenation Redundancy - if ISBT Concatenation is set to Autodiscriminate, use this parameter to set the number of times the device must decode an ISBT symbol before determining that there is no additional symbol.
	• Emulate UCC128
	• Select ISBT Concatenation - select an option for concatenating pairs of ISBT code types.
	  • Disable ISBT Concatenation - the device does not concatenate pairs of ISBT codes it encounters.
	  • Enable ISBT Concatenation - there must be two ISBT codes in order for the device to decode and perform concatenation. The device does not decode single ISBT symbols.
	  • Autodiscriminate ISBT Concatenation - the digital scanner decodes and concatenates pairs of ISBT codes immediately. If only a single ISBT symbol is present, the digital scanner must decode the symbol the number of times set in ISBT Concatenation Redundancy.
	• Ignore Code128 FNC4 - enable to strip the Code 128 <FNC4> character from the decode data.

Table 32 Symbologies (Continued)

Setting	Description
Interleaved 2 of 5 (I 2 of 5)	Configurable settings include: • Set lengths for I 2 of 5 to any length, one or two discrete lengths, or lengths within a specific range. **NOTE:** When setting lengths, enter a leading zero for single-digit numbers. • Check Digit Verification - check the integrity of all I 2 of 5 symbols to verify the data complies with either the specified Uniform Symbology Specification (USS), or the Optical Product Code Council (OPCC) check digit algorithm. • Transmit Check Digit - scan the appropriate barcode to transmit I 2 of 5 data with or without the check digit. • Convert Interleaved 2 of 5 Length to EAN13 - convert 14-character I 2 of 5 codes to EAN-13 and transmit them to the host as EAN-13. **NOTE:** To accomplish this, the I 2 of 5 code must be enabled, and the code must have a leading zero and a valid EAN-13 check digit. • Febraban (Interleaved 2 of 5 Length) - Febraban is an I 2 of 5 barcode of length 44 that requires special check characters to be inserted in the transmitted data stream. When enabled, the I 2 of 5 internal check digit calculation and transmission is disabled.

Data Matrix	Configurable settings include:
	• Decode Data Matrix Mirror Images - use one of the following barcodes to select an option for decoding mirror image Data Matrix barcodes:
	  • Regular Only - decodes regular images only.
	  • Mirrored Only - decodes mirrored images only.
	  • Both Regular and Mirrored - decodes both regular and mirrored images.
	• Inverse Data Matrix - sets the Data Matrix inverse decoder setting.
	  • Regular Only - decodes regular Data Matrix barcodes only.
	  • Inverse Only - decodes inverse Data Matrix barcodes only.
	  • Both Regular and Inverse - decodes regular and inverse Data Matrix barcodes.
	• Enable or Disable GS1 Datamatrix.

PDF417	Configurable settings include: • Enable MicroPDF.

Table 32 Symbologies (Continued)

Setting	Description

QR Code	Configurable settings include:
	• Enable MicroQR (Enabled by Default).
	• Enable GS1 QR.
	• Select Linked QR Mode - use one of the following barcodes to select a linked QR mode.
	  • Linked QR Only - does not decode individual QRs from a set of linked QR codes.
	  • Individual QR With Headers - decodes individual QR codes from a set of linked QR codes, preserving the header information and data.
	  • Individual QR No Headers - decodes individual QRs from a set of linked QR codes and transmits data only without header information.

Table 32 Symbologies (Continued)

Setting	Description
UPC/EAN	General UPC/EAN settings include: • Select UPC/EAN/JAN Supplementals. • 2 Digital Supplemental. • 5 Digital Supplemental. Configurable UPC-A settings include: • Enable UPC-A - preamble characters are part of the UPC symbol, and include Country Code and System Character. • Select UPC-A Preamble - select the appropriate option to transmit a UPC-E preamble compatible with the host system. • Transmit UPC-A Check Digit - transmits data with or without the UPC-A check digit. Configurable UPC-E settings include: • Enable UPC-E - preamble characters are part of the UPC symbol, and include Country Code and System Character. • Select UPC-E Preamble - select the appropriate option for transmitting a UPC-E preamble to match the host system. • Transmit UPC-E Check Digit - transmits data with or without the UPC-E check digit. • Convert UPC-E to UPC-A - converts UPC-E (zero suppressed) decoded data to UPC-A format before transmission. Configurable UPC-E1 settings include: • Enable UPC-E1. • Select UPC-E1 Preamble - the preamble parameter that transmits the system character and country code applies to the converted barcode. • Transmit UPC-E1 Check Digit - transmits the data with the UPC-E1 check digit. • Convert UPC-E1 Check Digit to UPC-A - converts UPC-E1 (zero suppressed) decoded data to UPC-A format before transmission. Configurable EAN-13/JAN-13 settings include: • Enable EAN-13/JAN13 Configurable EAN-8/JAN8 settings include: • Enable or Disable EAN-8/JAN8 • Enable EAN-8/JAN8 Extend

Table 32 Symbologies (Continued)

Setting	Description
Code 93	Configurable settings include: • Select Code 93 Length - the length of a code refers to the number of characters (for example, human-readable characters), including check digit(s) the code contains. Set lengths for Code 93 to any length, one or two discrete lengths, or lengths within a specific range. **NOTE:** When setting lengths, enter a leading zero for single-digit numbers.

DotCode	Configurable settings include:
	• Select DotCode Inverse - scan one of the following barcodes to select a DotCode Inverse decoder setting. Setting options are:
	  • Regular Only - decodes DotCode barcodes with normal reflectance only.
	  • Inverse Only - decodes DotCode barcodes with inverse reflectance only.
	  • Both Regular and Inverse - decodes both regular and inverse DotCode barcodes.
	• Select DotCode Mirror - use one of the following barcodes to select a DotCode Mirror decoder setting:
	  • Never - decodes non-mirrored DotCode barcodes only.
	  • Always - decodes mirrored DotCode barcodes only.
	  • Auto - decodes both mirrored and non-mirrored DotCode barcodes.
	• Determine DotCode ECC Erasure - set the maximum number of erasures in DotCode codewords before passing the codewords for error correction.
	NOTE: The value range is from 4 to 20. The default value is 10. A higher value means DotCode barcodes are easier to decode, but the chance of a decode error is higher.

MaxiCode	There are no configurable settings for Maxicode.
Mailmark	Enable or disable Mailmark.

Aztec	Configurable settings include:
	• Select Inverse Aztec - set the Aztec inverse decoder setting, options include:
	  • Regular Only - decodes regular Aztec barcodes only.
	  • Inverse Only - decodes inverse Aztec barcodes only.
	  • Both Regular and Inverse - decodes both regular and inverse Aztec barcodes.

Table 32 Symbologies (Continued)

Setting	Description
MSI	Configurable settings include: • Select MSI Length - the length of a code refers to the number of characters (for example, human-readable characters), including check digit(s) the code contains. Set MSI lengths to any length, a single discrete length, or a range of lengths. **NOTE:** When setting lengths, enter a leading zero for single-digit numbers. • MSI Check Digits - when using MSI symbols, one check digit is mandatory and always verified by the device. The second check digit is optional. If the MSI codes include two check digits, scan the Two MSI Check Digits barcode to verify the second check digit. • Transmit Check Digit - transmit MSI data with or without the check digit. • Select MSI Check Algorithm: two algorithms are available for verifying the second MSI check digit. Select the parameter that corresponds to the algorithm used to encode the check digit. • MSI Reduced Quiet Zone - enable or disable decoding MSI barcodes with reduced quiet zones

Table 32 Symbologies (Continued)

Setting	Description
CODABAR	Configurable settings include: • Select Codabar Length - the length of a code refers to the number of characters (for example, human-readable characters), including check digit(s) the code contains. Set Codabar lengths to any length, a single discrete length, or a range. **NOTE:** When setting lengths, enter a leading zero for single-digit numbers. • CLSI Editing - enable to strip the start and stop characters and insert a space after the first, fifth, and tenth characters of a 14-character Codabar symbol. This parameter is helpful if the host system requires this specific data format. **NOTE:** Symbol length does not include start and stop characters. • NOTIS Editing - enable this parameter to strip the start and stop characters from a decoded Codabar symbol. This parameter is helpful if the host system requires this specific data format. • Upper or Lower Case Start/Stop Characters - select whether to detect upper case or lower case Codabar start/stop characters. • Select Security Level - the device offers four levels of decode security for delta bar codes, including UPC/EAN and Code 93. Select increasing levels of security for decreasing levels of barcode quality. **NOTE:** There is an inverse relationship between security and the aggressiveness of the scanner; choose only the level of security necessary for any given application. • Mod 16 Check Digit Verification - check the Codabar Mod 16 check digit to verify that the data complies with the specified check digit algorithm. • Transmit Codabar Check Digit - select whether or not to transmit the Codabar check digit(s).
Discrete 2 of 5	Select Length • Any • Discrete • Range (4-55)

Table 32 Symbologies (Continued)

Setting	Description

GS1 Databar	Configurable settings include:
	• Omnidirectional
	• Limited
	• Expanded
	• Transmit Codabar Check Digit - select whether or not to transmit the Codabar check digit(s).
	Options:
	• Convert GS1 Databar to UPC/EAN/JAN
	• Limited Margin Check
	  • Level 1 - this setting does not require a clear margin and complies with the original GS1 standard, yet can result in a misdecode of DataBar limited barcodes when scanning UPC symbols that start with digits 9 and 7.
	  • Level 2 - this setting automatically detects risks. This level of margin check can result in a misdecode of Databar Limited barcodes when scanning UPC symbols. If a misdecode occurs, the scanner operates in Level 1 or 3.
	  • Level 3 - this setting reflects the newly proposed GS1 standard and requires a five-times trailing clear margin.
	  • Level 4 - this setting extends beyond the standard GS1 requirement.
	• Security Level
	  • Level 0 - the scanner operates in its most aggressive state while providing sufficient security in decoding most in-spec barcodes.
	  • Level 1 - this setting eliminates most misdecodes.
	  • Level 2 - this setting applies greater barcode security requirements if Security Level 1 does not sufficiently eliminate misdecodes.
	  • Level 3 - this setting applies the highest security requirements.
	NOTE: This option is an extreme measure against misdecodes and significantly impairs the scanner's decoding ability. If this level of security is required, it is recommended to use high-quality barcodes.

Table 32 Symbologies (Continued)

Setting	Description

Pharmacode	Configurable settings include:
	• Bar Count (3-16)
	• Orientation
	  • Horizontal
	  • Vertical
	• Direction
	  • Normal
	  • Reverse

Best Practices for Using Pharmacode

Pharmacode (Laetus) is a highly sensitive barcode that is prone to false positives when scanned from an image captured by a barcode reader.

- Disable other Barcode Symbolgies
  - Pharmacode can easily be misinterpreted if multiple barcode symbologies are enabled simultaneously. It is advisable to disable all other symbologies when scanning Pharmacode to prevent interference and misreadings.
- Minimize Background in the Scan Area
  - Avoid capturing unnecessary background elements, as random patterns, text, or lines may be mistakenly identified as a valid Pharmacode.
  - Ensure high contrast between the barcode and its background for better recognition.
- Keep the Barcode Area Clear of Text and Graphics
  - Nearby text, lines, or graphical elements may be misinterpreted as part of the barcode, leading to incorrect readings.
  - Maintain a quiet zone around the barcode to enhance scanning accuracy.
- Implement Verification Algorithms
  - Use software validation checks to ensure that the scanned value matches expected parameters (fox example, specific length, numerical range.
- Avoid Excessive Light and Reflections
  - Glossy or reflective surfaces (for example, laminated packaging) can cause distortions in the barcode image, leading to scanning errors.
  - Adjust the angle of illumination to reduce glare and improve readability.
- Ensure Proper Printing Quality
  - Poorly printed Pharmacodes with smudges, inconsistent bar widths, or faded ink can lead to scanning failures.
  - Print Pharmacodes using high-precision techniques and verify compliance with Laetus specifications.

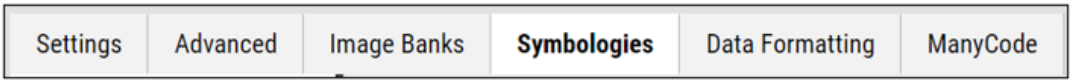
- Optimize Scan Resolution
  - Avoid blurry or low-quality images, as they can cause false positives or incorrect readings.

Using Pharmacode

Pharmacode, also known as the pharmaceutical binary code, is primarily used in the pharmaceutical industry to represent numeric data. A pharmacode is a one-dimensional barcode that consists of a series of vertical bars with varying widths. It is a type of barcode symbology widely used in the pharmaceutical industry for packaging control, medication identification and tracking purposes. It is a specific implementation of barcode technology designed to encode numerical data.

NOTE: Pharmacode can be misinterpreted if multiple barcode symbologies are enabled simultaneously. It is advisable to disable all other symbologies when scanning Pharmacode to prevent interference and misreadings.

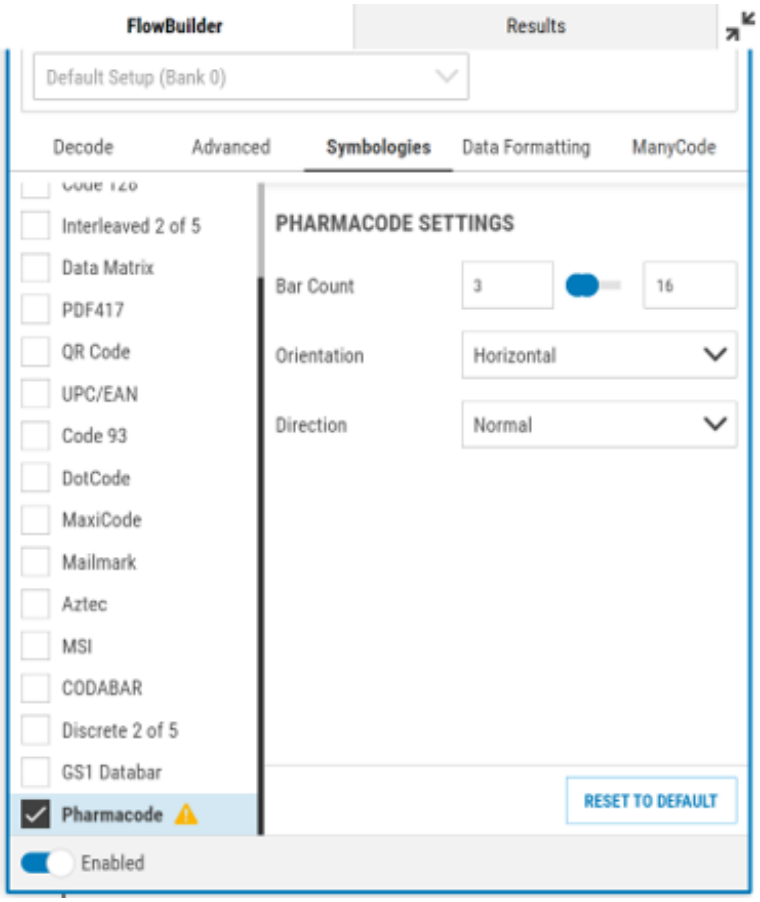
1. Access the **Symbologies** tab.



2. Select **Pharmacode** from the symbologies list.

NOTE: If you are using another symbology in addition to Pharmacode, observe the caution recommending against the use of any other symbology while using Pharmacode, as it is known to misinterpret barcodes when attempting multiple decodes simultaneously.

3. Configure the **Pharmacode** settings.



- Bar Count
  - 3-16 (Default)
- Orientation
  - Horizontal (Default)
  - Vertical
- Direction
  - Normal (Default)
  - Reverse

Using ManyCode

Use ManyCode to decode multiple barcodes simultaneously and determine their sorting arrangement.

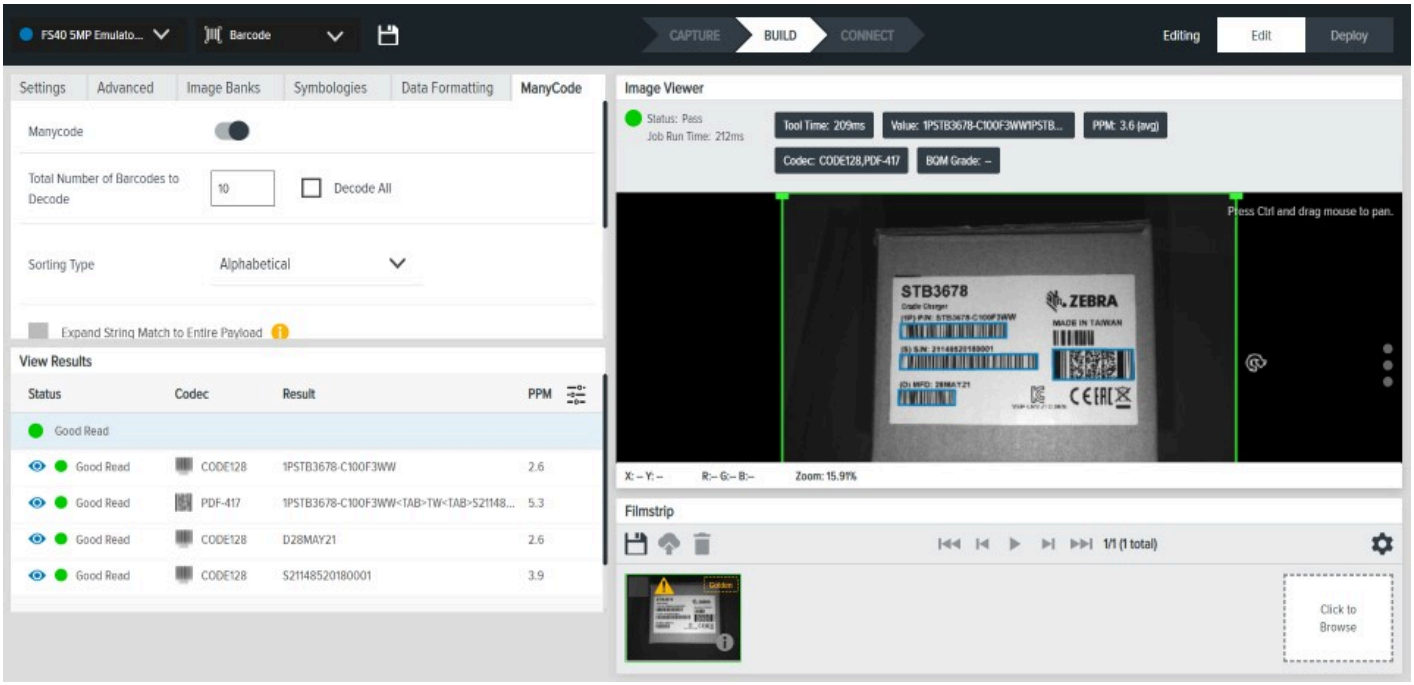


Table 33 ManyCode Tools

Setting	Description
ManyCode (Disable/Enable)	Enabling this setting allows the barcode to read multiple barcodes in a single iteration.
Total Number of Barcodes to Decode	Limits the number of barcodes to attempt during a single ManyCode scan iteration. For example, if you have eight barcodes in your field of view but set this to five, only the first five barcodes read are decoded, and the scan iteration ends.
Decode All	Explicitly defines no limit for the number of barcodes to attempt to read in a single scan iteration.
Sorting Type	Define how barcode results are sorted: • Alphabetical • First Decoded • Top to Bottom • Left to Right
Expand String Match to Entire Payload	The tool searches for a string match in the entire payload when enabled.
Enable Partial Results	If the number of decodes is less than the **Total Number of Barcodes to Decode** setting, enabling this setting still outputs these barcodes.

Table 33 ManyCode Tools (Continued)

Setting	Description
Enable Decode of Identical Symbols	By default, identical (duplicate) barcodes are only read once. When enabled, all duplicate barcodes are read.
Exhaustive Search Attempt	Sets the underlying ManyCode barcode algorithm to prioritize read performance over speed.
Apply Across Level Selection	This option is only active in Level Continuous mode while Read Multiple Barcodes is enabled.

Using Machine Vision Tools

Deploying Machine Vision Jobs

To build and deploy a vision scanning (VS) job, start by selecting a machine vision tool and dragging it onto the FlowBuilder. Using FlowBuilder, stack additional tools onto the workflow or configure the intended results to deploy the Job. Use the QuickDraw tool to streamline the creation of a specific toolset.

Common Machine Vision Tool Settings

Some machine vision tools share common settings. Refer to the table below to understand how to configure them based on your use case.

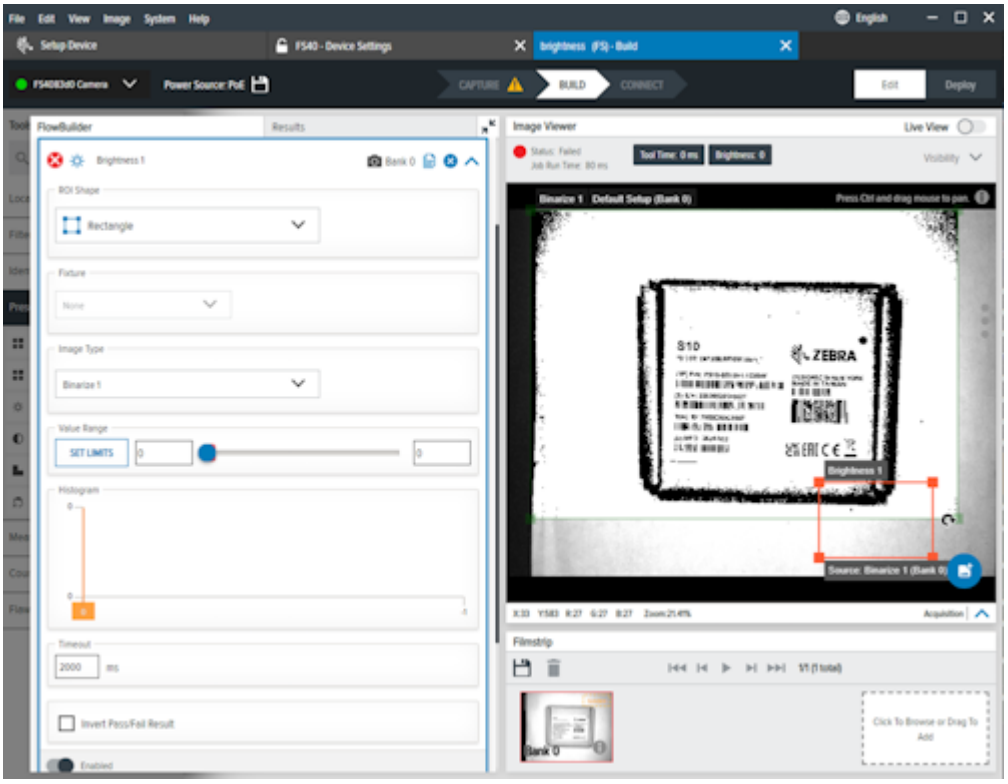
Table 34 Common Machine Vision Tool Settings

Setting	Description
ROI Type	Select a rectangular or circular Region of Interest (ROI).
Histogram	Displays the number of pixels between two values in a user-defined value range.
Timeout	Set a time limit to stop the execution of the inspection tool, producing a failed result.
Inverse Pass/Fail	Inverts the output result when enabled.

Using Image Types

When using the **Image Type** drop-down menu to use a filter or tool output as the source image, the ROI of the tool must be fully contained within the ROI of the source images. If the tool's ROI exceeds the source's ROI, the tool automatically issues a failing result.

In the following image, the **Brightness** tool uses the **Binarize** tools output as the source image. As a result, the ROI of the **Brightness** tool must be fully contained within the **Binarize** tools source ROI. If there is an overlap between the two ROIs, the **Brightness** tool issues a fail.



Locate Tools

Locate tools find a single occurrence of a predefined template on an image by comparing object edges.

Locate Object

Locate Object finds a specific pattern in a filmstrip based on an edge inside a user-defined region of interest.

NOTE: Aurora Focus imposes a 1GB size limit on all models.

If the total size of all models within a specific region exceeds this limit, use one of the following strategies to reduce the size:

-
- Increase edge contrast to keep only the most important edges (reduce noisy edges).
- Reduce scale (if possible)
- Reduce rotation (if possible)

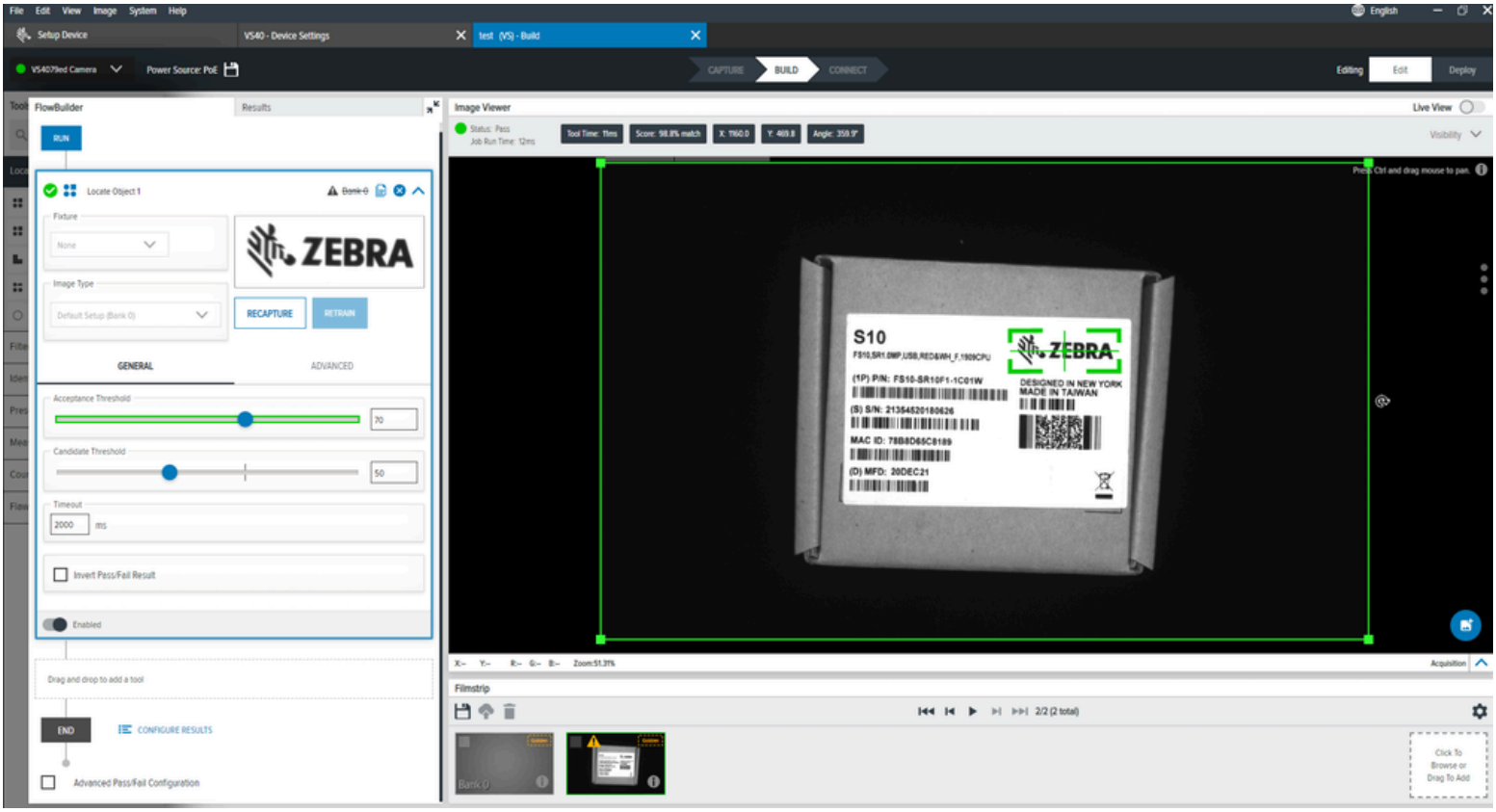


Table 35 Locate Object Settings

Setting	Description
Acceptance Threshold	Determines the minimum score of the valid object occurrence.
Candidate Threshold	Filters out objects below the acceptable threshold in the intermediate phases of the algorithm execution. Note that increasing value can improve performance. However, low-quality objects may not be found.

General Locate Object Settings

Configure General Locate Object settings such as acceptance threshold, candidate threshold, and rotation threshold.

Table 36 General Locate Object Settings

Setting	Description
Acceptance Threshold	The minimum match score required to be considered a passing match. If multiple object match scores exceed this minimum, the highest matching object score is used as the final match.
Candidate Threshold	The threshold for a match of the trained pattern to a pattern in the acquired image.

Table 36 General Locate Object Settings (Continued)

Setting	Description
Rotation Threshold	The minimum match score required to be considered a passing match. If multiple object match scores exceed this minimum, the highest matching object score is used as the final match.

Advanced Locate Object Settings

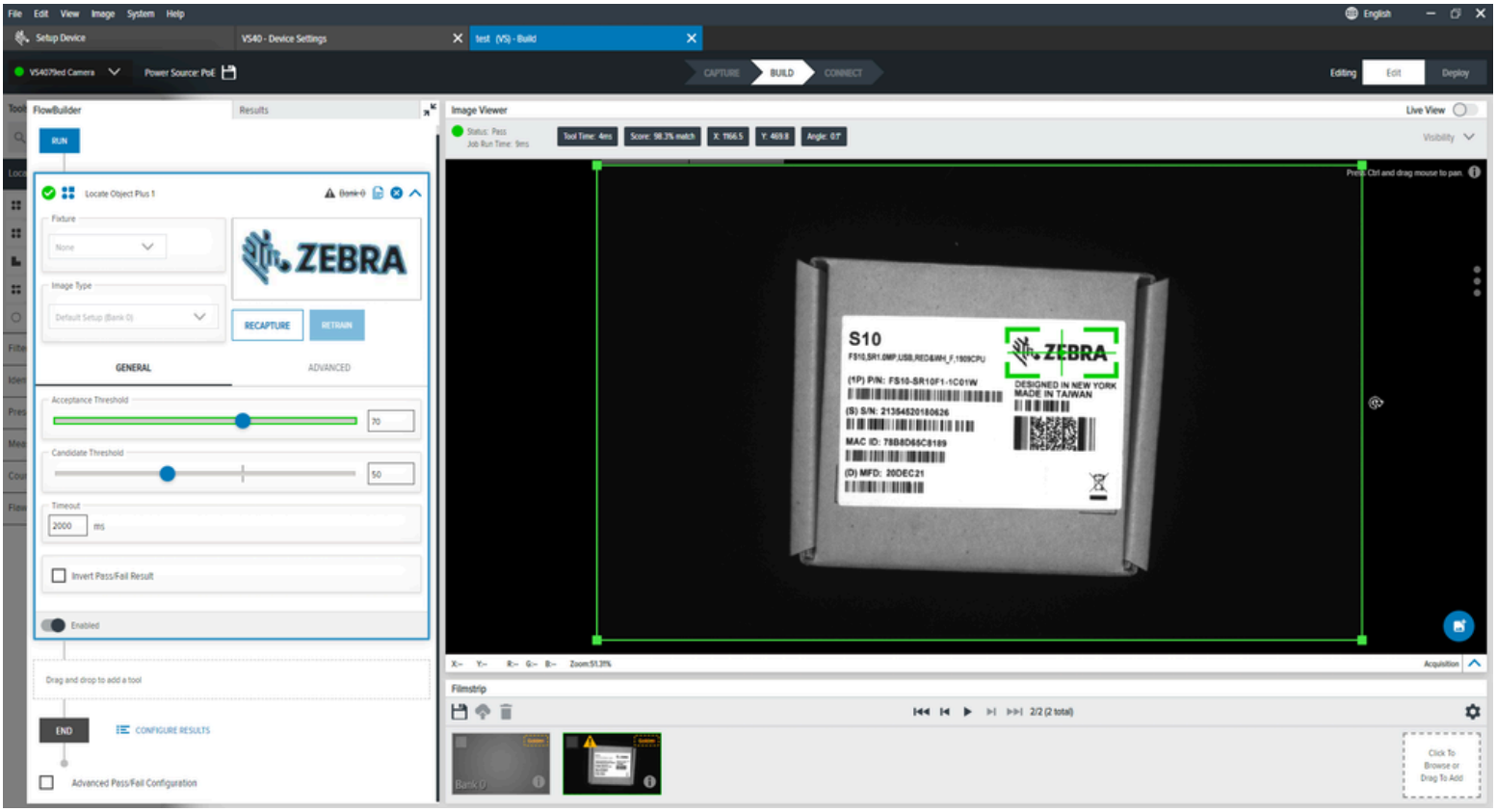
Configure Advanced Locate Object settings such as performance control, noise, rotation threshold, and scale deviation.

Table 37 Advanced Locate Object Settings

Setting	Description
Performance Control	Select the best coarseness and search type combinations for a more performant search.
Noise	This filter removes pixel-level noise while preserving edge data. • Off - the object edges have no noise. • Low - the object edges have a low level of noise. The object shape and the model shape are nearly identical. • High - the object edges have a considerable noise level, or the object shape slightly differs from the model shape.
Rotation Threshold	The minimum match score required to be considered a passing match. If multiple object match scores exceed this minimum, the highest matching object score is used as the final match.
Allow Scale Deviation	Allows locating objects slightly smaller or bigger than those used during model creation.

Locate Object Plus

Locate Object Plus finds specific patterns in a specified region of interest and evaluates them based on advanced characteristics such as minimum edge contrast and scale factor.



Advanced Locate Object Plus Settings

Configure Advanced Locate Object Plus settings such as performance control, noise, rotation threshold, minimum edge contrast and scale factor.

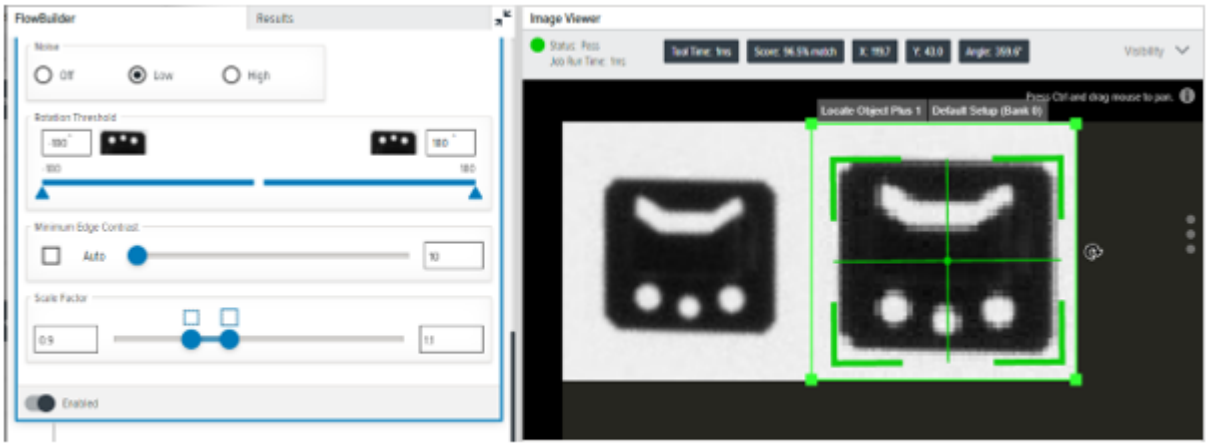
Table 38 Advanced Locate Object Plus Settings

Setting	UI Element
Performance Control	Use the selector to choose the best coarseness and search type combinations for a more performant search.
Noise	Use this filter to remove pixel-level noise while preserving edge data.
Rotation Threshold	A minimum match score is required to be considered a passing match. If multiple object match scores are above this minimum, the highest matching object scores are used as the final match.
Minimum Edge Contrast	Manually sets the minimum contrast in the acquired image to match the trained patterns.

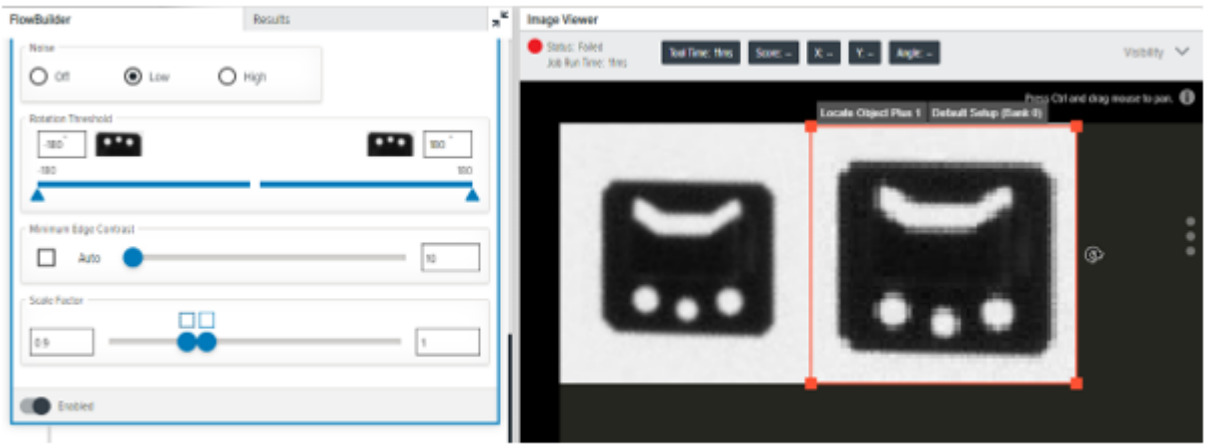
Table 38 Advanced Locate Object Plus Settings (Continued)

Setting	UI Element
Scale Factor	The Minimum Scale Factor and Maximum Scale Factor parameters determine the range of template scales that will be considered in the matching process. It enables locating objects that are slightly smaller or bigger than the object used during model creation. A wide range of possible scales introduces significant overhead (both in memory usage and computing time). As a result, it is recommended to limit the range whenever possible.

Smaller object used to model creation

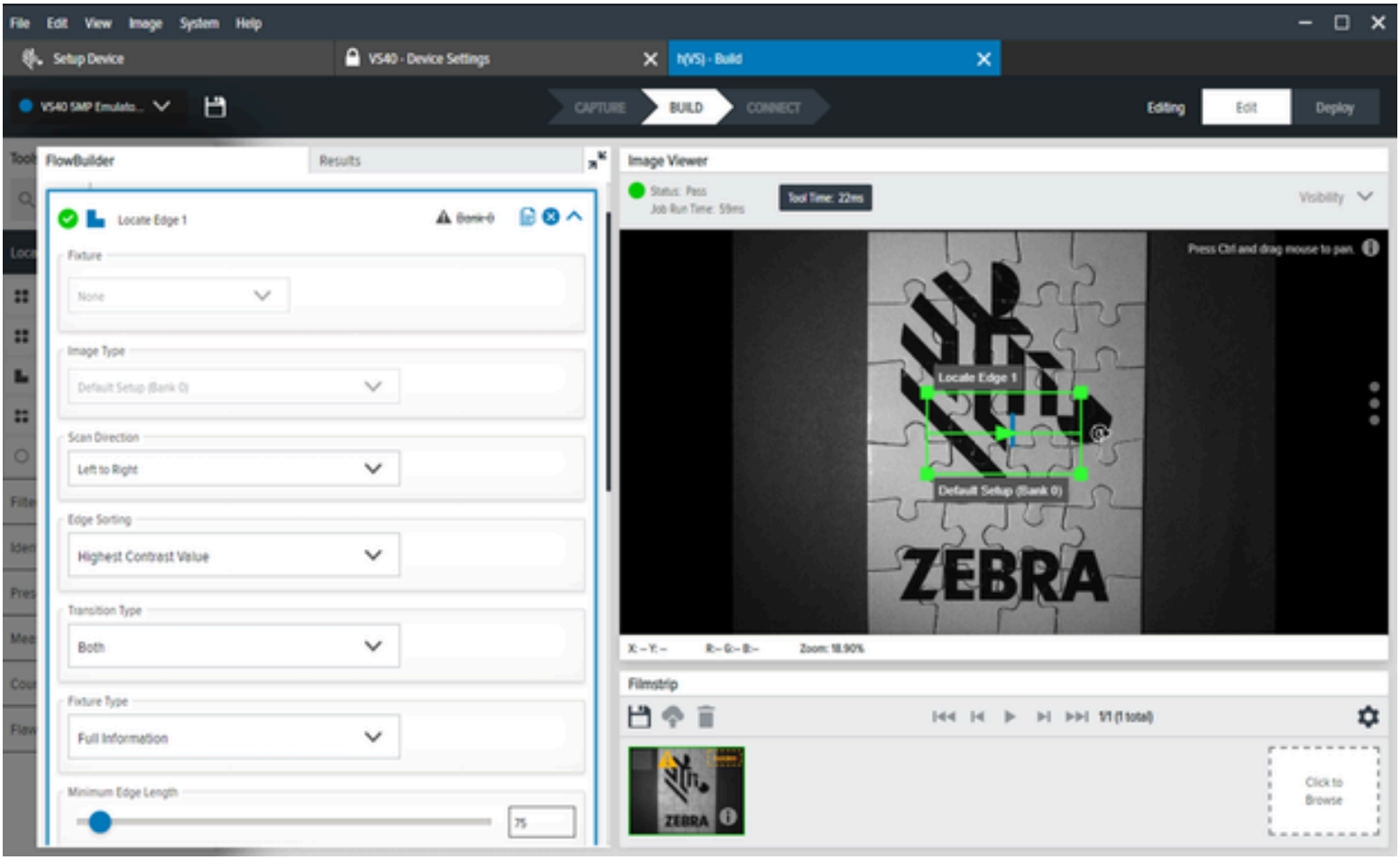


Max Scale Factor = 1.1



Locate Edge

Locate Edge identifies transitions based on the highest contract in a user-defined region of interest.



General Locate Edge Settings

Configure General Locate Edge settings such as fixture, image type, scan direction, edge sorting, transition type, fixture type, minimum edge length, maximum gap, skew tolerance, edge contrast and edge profile.

Table 39 General Locate Edge Settings

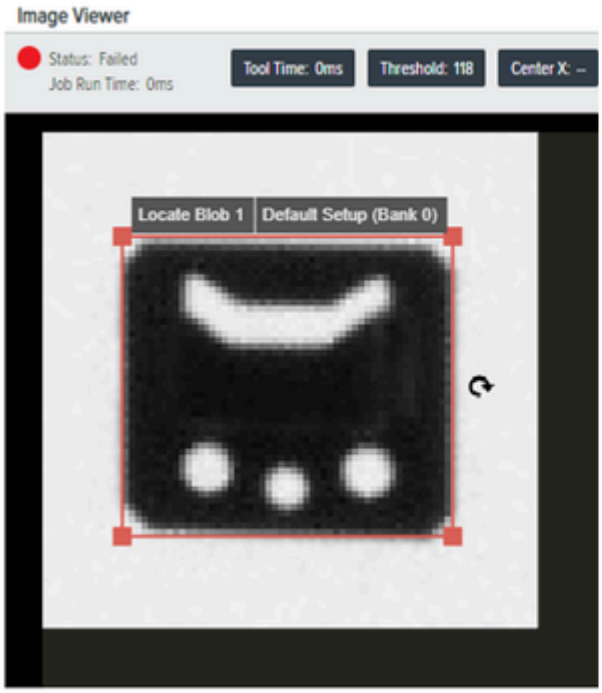
Setting	Description
Fixture	Select a previous tools' result to be used to position this tool's ROI.
Image Type	Select the type of image that the inspection type will use.
Scan Direction	Sets the direction that the tool uses when searching for edges.
Edge Sorting	Narrow down a collection of found edges to a single found edge.
Transition Type	Select the type of edge transition used to find the edge.
Fixture Type	Include all information, position only, or Y position only.
Minimum Edge Length	The minimum length in pixels for an edge to use.
Maximum Gap	The maximum size in pixels to consider in gaps for an edge.
Skew Tolerance	Degrees to attempt to measure a line if it is not straight.
Edge Contrast	Threshold acceptance to consider a line.
Edge Profile	Displays the contrast score of the features across a region of interest.

Locate Blob

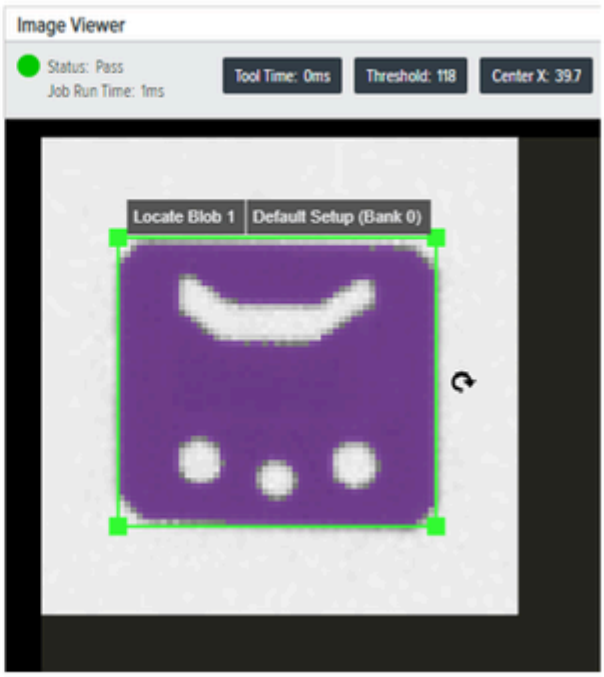
A blob is a set of connected light-or dark-colored pixels. The Locate Blob tool detects blobs that pass specific filter parameters within a region of interest.

This filter is helpful for quickly segmenting an image. The Locate Blob tool performs a series of operations on the image, including; thresholding using the **Threshold** parameter and removing holes by setting **Fill Holes**.

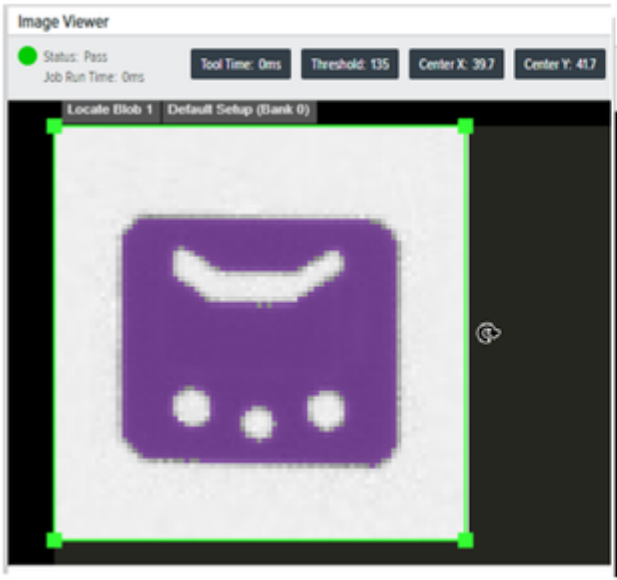
Filter Boundary blobs by setting **Allow Boundary** blobs.



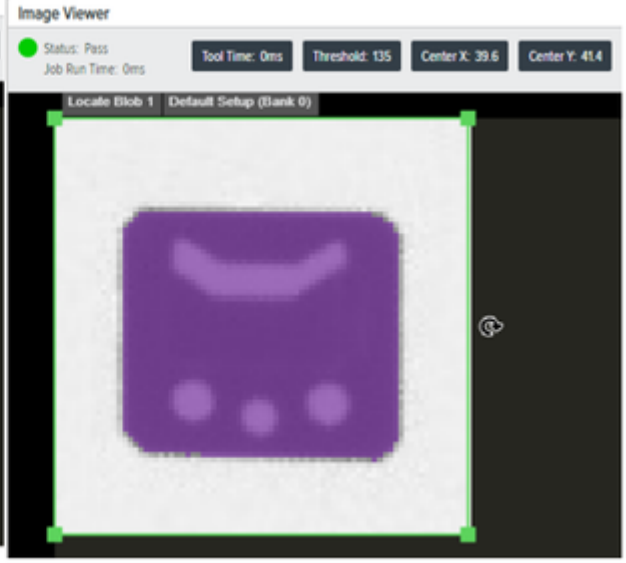
Allow Boundary Blobs = False



Allow Boundary Blobs = True



Fill Holes = False



Fill Holes = True

General Locate Blob Settings

Configure General Locate Blob settings such as fixture, image type, threshold, histogram result, filters and sorting.

Table 40 General Locate Blob Settings

Setting	Description
Fixture	Select a previous locate tool result that is used to position the tool's ROI.
Image Type	Select the image that the inspection tool will use.
Threshold	Select a threshold for black or white blobs. This value is automatically set to 117.
Histogram Result	Visual representation of the number of pixels found at each grayscale level.
Filters	Apply filters to set the criteria used to consider a blob as valid.
Sorting	Defines the priority for selecting the blob to return.

Locate Circle

Locate Circle finds a circle model inside a user-defined region of interest that fits the specified parameters.

General Locate Circle Settings

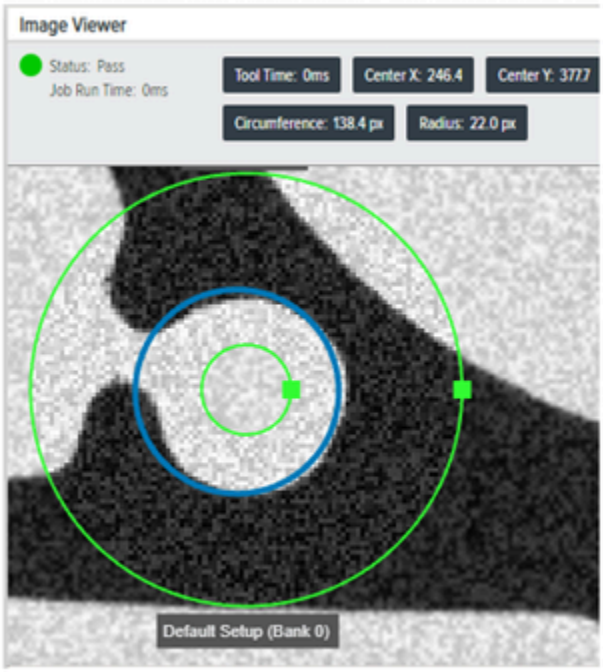
Configure General Locate Circle settings such as find by, transition type, maximum edge magnitude, edge profile, and scale calibration.

Table 41 General Locate Circle Settings

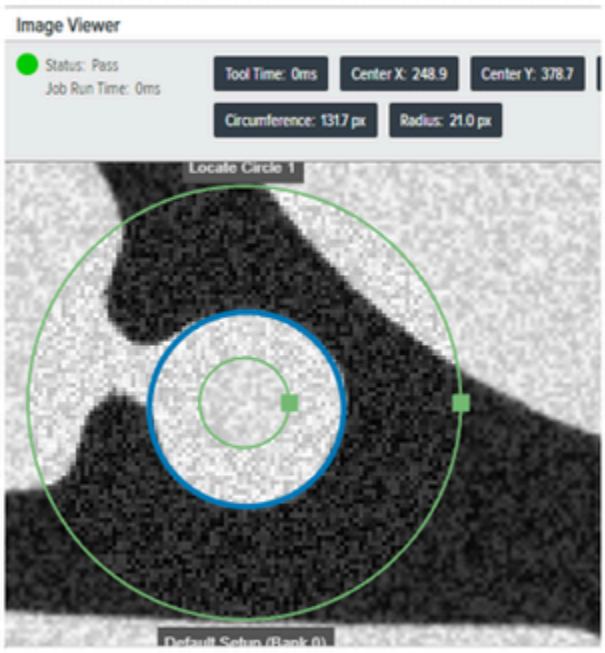
Setting	Description
Find By	Describes which result is reported as filter output: • Best Score - searches for the best match to the detected edges within a given ROI. • Largest Circle - returns the largest circle that matches the edges found in a given ROI. • Smallest Circle - returns the smallest circle that matches the edges found in a given ROI.
Transition Type	Determines what is considered an edge in a given image. • Blended - any change in pixel intensity determines an edge. • Both - changes from dark to bright pixels or changes from dark to bright pixels determine an edge. • Dark to Light - changes from dark to bright pixels determine an edge. • Light to Dark - changes from bright to dark pixels determine an edge.
Maximum Edge Magnitude	The minimum acceptable edge strength.
Edge Profile	Displays the contrast score of the features across the region of interest.
Scale Calibration	Calibrates pixel values to engineering units.

Advanced Locate Circle Settings

When **Enable Outlier Suppression** is enabled, the resulting output resembles the output example below:



Enable Outlier Suppression = False



Enable Outlier Suppression = True

Table 42 Advanced Locate Circle Settings

Setting	Description
Scan Width	Defines the width of the area neighboring each scanning line that should be considered when calculating the results in pixels.
Scan Count	Sets the number of scanning segments used around the circumference when defining the circle edge.
Maximum Incompleteness	Sets the percentage of points in the circle edge that are allowed to be missing while still providing a passing result.
Enable Outlier Suppression	Allows the suppression of influence of values far from most others on the result using Tukey's M-estimator.

Filter Tools

Filter Tools are facilitated by the use of a kernel. A kernel is repeatedly centered at each pixel within the dimensions of the region that is being transformed. Every pixel is either added to the resulting region or not, depending on the operation-specific condition set on the minimum number of kernel pixels that have to overlap with actual input region pixels in the given position of the kernel.

Binarize

The Binarize Tool converts the image to monochrome.

The operation transforms each pixel value to the maximum or minimum level, creating a binary image. The result of the transformation depends on the pixel intensity:

- Pixel values in the range (MinValue, MaxValue) are transformed to the maximum level.

- Other pixel values are transformed to the minimum level.

Original Image

Result

Table 43 Binary Settings

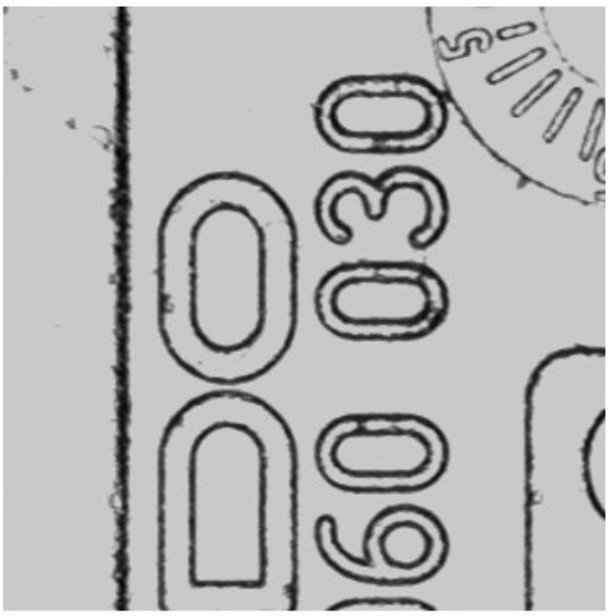
Setting	Description
Fixture	Select a previous tool's result to position this tool's ROI.
Image Type	Select which image to use with this tool.
Threshold	The algorithm detects the best threshold to use for the filter.
Histogram Result	Visual Representation of the number of pixels found at each greyscale level.

Dilate

The Dilate tool replaces each pixel with a maximum of pixels within the kernel. This tool also thickens bright features in an image and reduces dark features.

The operation replaces each pixel with the brightest pixel in its neighborhood, thus shrinking dark areas and expanding the bright ones.

The following images display a Dilate tool result with three Kernel Rows and three Kernel Columns.



Original Image

Result

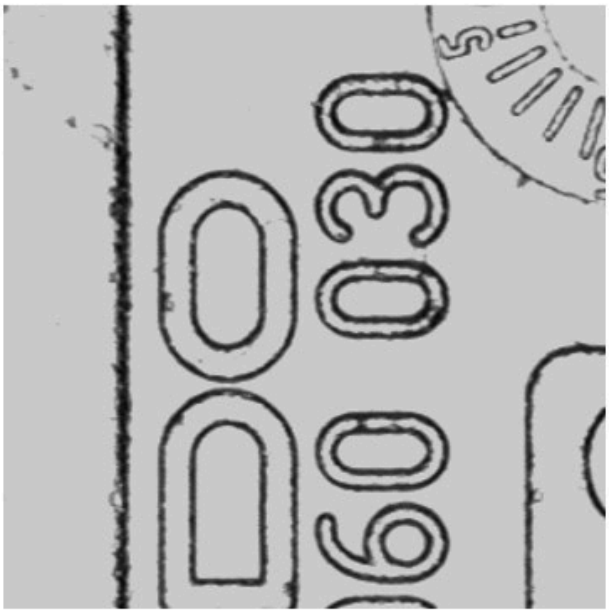
Table 44 Dilate Tool Settings

Settings	Description
Fixture	Select a previous locate tool's result that is used to position this tool's ROI.
Image Type	Select which image this inspection tool uses.
Kernel	Select the kernel values used to calculate the filter output.

Erode

The operation replaces each pixel with the darkest pixel in its neighborhood, thus shrinking bright areas and expanding the dark ones.

The following images display an Erode tool result with three Kernel Rows and three Kernel Columns.



Original image

Result

Table 45

Settings	Description
Fixture	Select a previous locate tool's result to position this tool's Region of Interest (ROI).
Image Type	Select which image this inspection tool will use.
Kernel	Select the kernel values used to calculate the filter output.

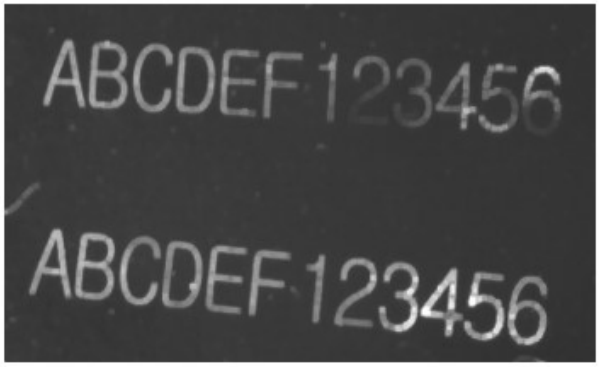
Open

The Open tool decreases the image's overall brightness and enhances the remaining bright features by contrast.

Removes small bright structures from an image (or fills in dark ones) by applying consecutive erosion and dilation.

The following images display an Open tool result with three Kernel Rows and three Kernel Columns:

Original image



Result

Table 46 Open Settings

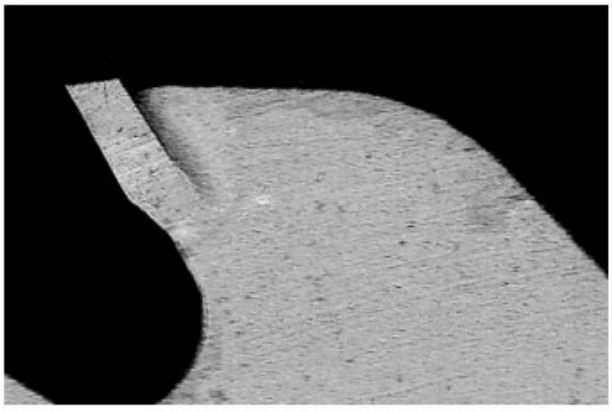
Settings	Description
Fixture	Select a previous locate tool's result to position this tool's Region of Interest (ROI).
Image Type	Select which image this inspection tool will use.
Kernel	Select the kernel values used to calculate the filter output.

Close

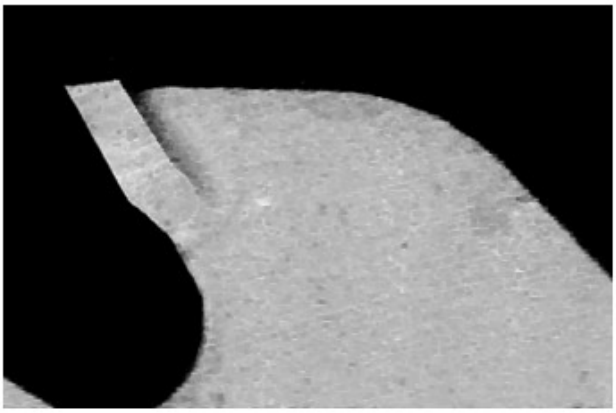
The Close tool increases the image's overall brightness and enhances the remaining dark features by contrast.

Removes small bright structures from an image (or fills in dark ones) by applying consecutive erosion and dilation.

The following image displays an Open tool result with three Kernel Rows and three Kernel Columns:



Original Image



Result

Table 47 Close Settings

Settings	Description
Fixture	Select a previous locate tool's result that is used to position this tool's Region of Interest (ROI).
Image Type	Select which image this inspection tool will use.
Kernel	Select the kernel values used to calculate the filter output.

Gradient Full

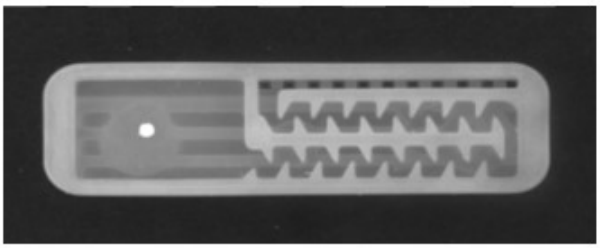
The Gradient Full tool brightens horizontal and vertical edges within the defined area of interest.

Method of estimation of the vectors' magnitude:

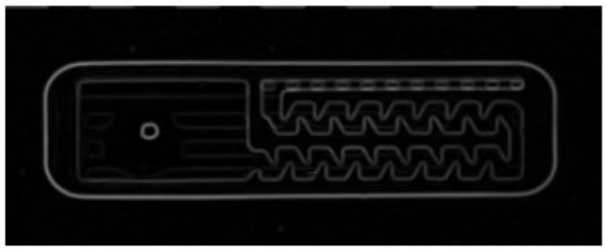
$$\text{Hypot-}\frac{\sqrt{x^2+y^2}}{4}$$

x- horizontal gradient component

y- vertical gradient component



Original Image



Result

Table 48 Gradient Full Settings

Settings	Description
Fixture	Select a previous locate tool's result to position this tool's Region of Interest (ROI).
Image Type	Select which image this inspection tool will use.
Scale	Increases the brightness of the edges in the output result. This input is used to scale the output edge brightness values.

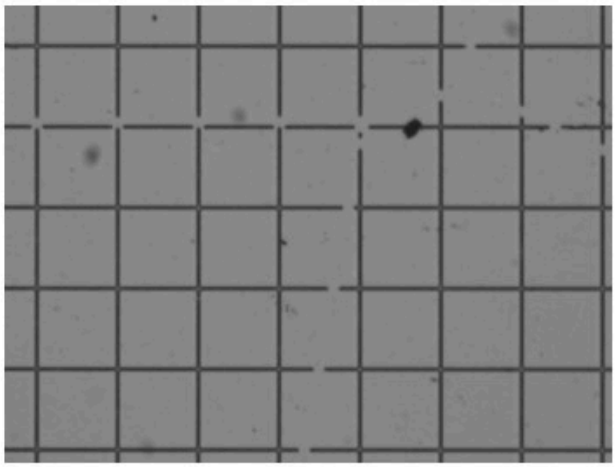
Gradient Horizontal

The Gradient Horizontal tool brightens horizontal edges within the defined region of interest.

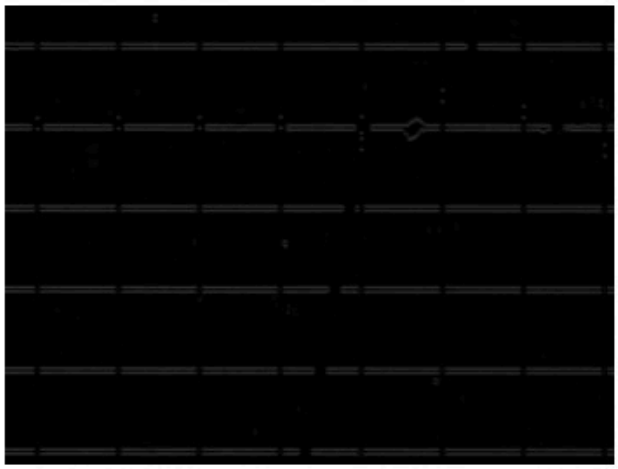
Method of estimation of the vectors' magnitude:

$$\text{Horizontal-}\frac{\sqrt{x^2}}{4}$$

x- horizontal gradient component



Original Image



Result

Table 49 Gradient Horizontal Settings

Settings	Description
Fixture	Select a previous locate tool's result to position this tool's Region of Interest (ROI).
Image Type	Select which image this inspection tool will use.
Scale	Increases the brightness of the edges in the output result. This input is used to scale the output edge brightness values.

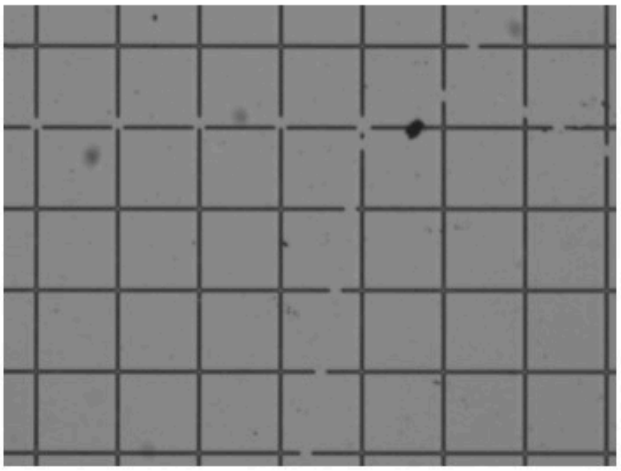
Gradient Vertical

The Gradient Vertical tool brightens vertical edges within the defined region of interest.

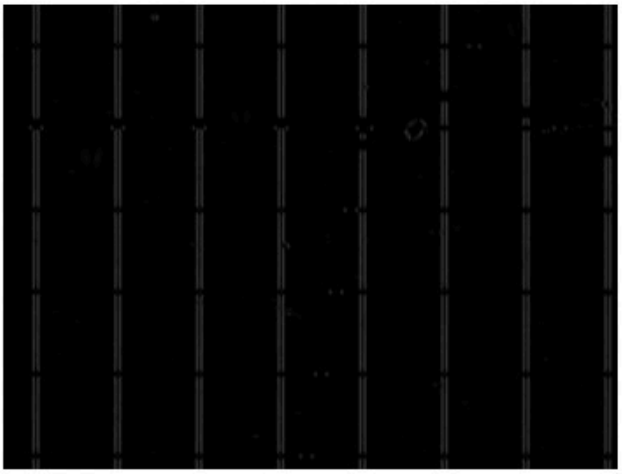
Method of estimation of the vectors' magnitude:

$$\text{Vertical-}\frac{\sqrt{y^2}}{4}$$

y- vertical gradient component



Original Image



Result

Table 50 Gradient Vertical Settings

Settings	Description
Fixture	Select a previous locate tool's result to position this tool's Region of Interest (ROI).
Image Type	Select which image this inspection tool will use.
Scale	Increases the brightness of the edges in the output result. This input is used to scale the output edge brightness values.

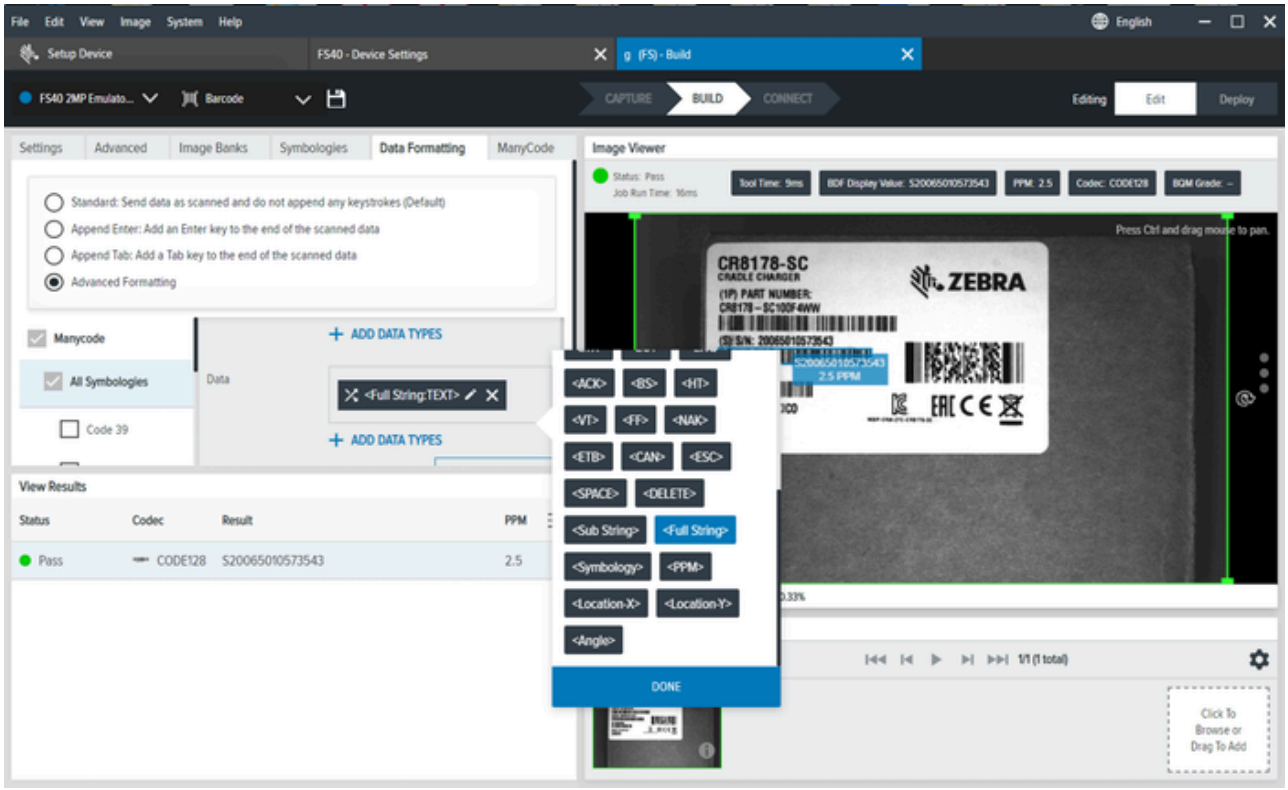
Identification Tools

Use the Identification Tools to read barcodes, DPM, Datacode, and Deep Learning OCR.

Using Barcode Quality Metrics for Identification Tools

To access and output various BQM statistics using the Advanced Data Formatting configuration:

1. Navigate to the **Data Formatting** tab.
2. Click the **Select Advanced Formatting** radio button.



3. Click **Add Data Types** in the **Data** section.
4. Scroll to observe various BQM-related statistics that you can send to your output.

<Angle>
<BQM Score>
<Decodability>
<Decode>
<Defects>
<Minimum Edge Contrast>
<Minimum Reflectance>
<Modulation>
<Symbol Contrast>
<Quiet Zone>
<Left Quiet Zone>
<Right Quiet Zone>
<Wide to Narrow Ratio>
<Inter Char Gap 2 Nar>
<Max Pixel Value In Extended Area>
<Maximum Reflectance>
<Minimum Reflectance>
<Unused Error Correction>
<Fixed Pattern Damage>
<Axial Nonuniformity>
<Grid Nonuniformity>
<Contrast Uniformity>
<Reflectance Margin>
<Horizontal Print Growth>
<Vertical Print Growth>

Read Barcode

Configure barcode decode settings such as Inverse 1D, ROI overlap, or ROI search.

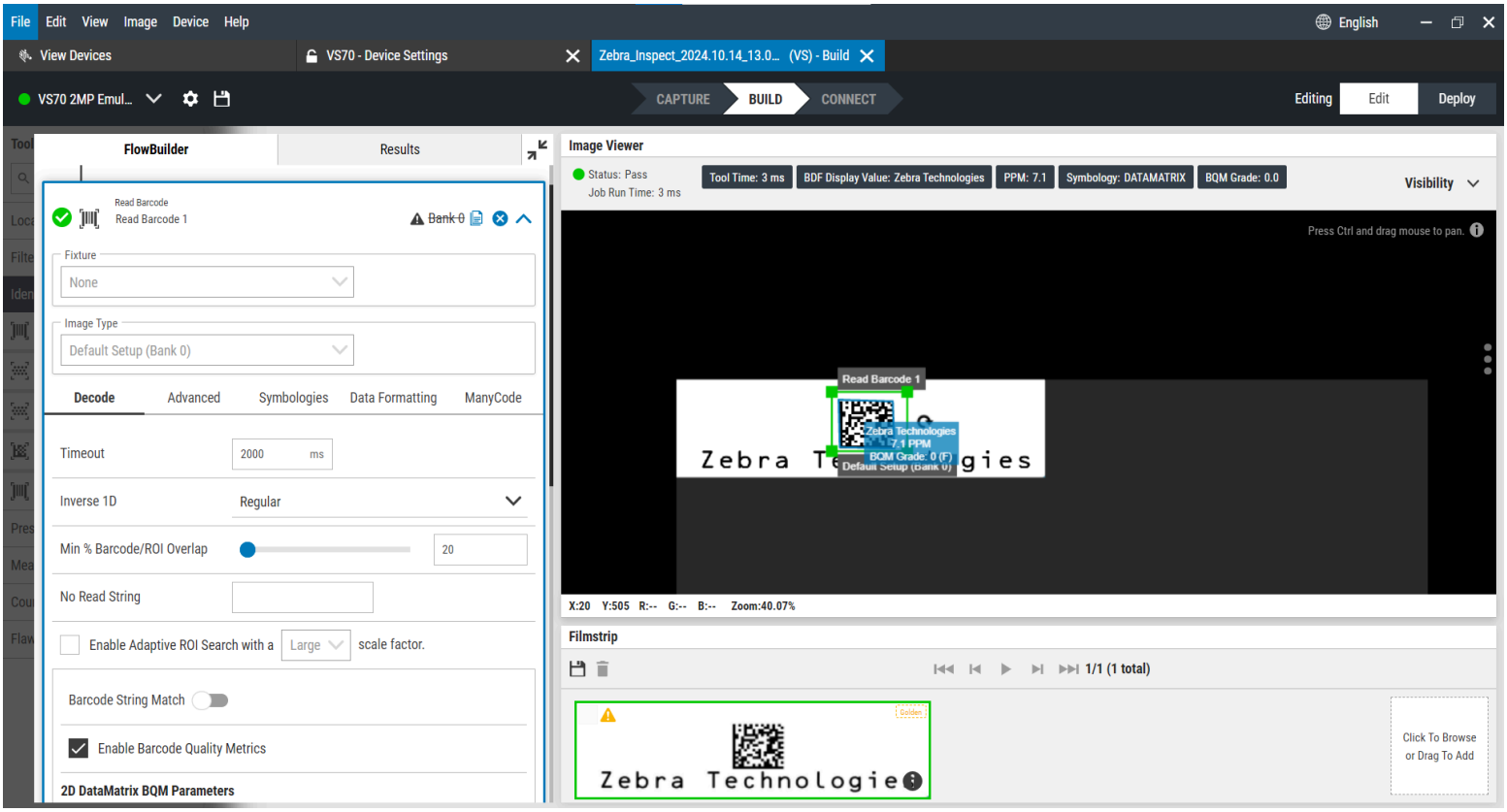


Table 51 Read DPM and Barcode Settings

Settings	Description
Inverse 1D	Choose the type of barcode to read.
Minimum Percentage Barcode/ROI Overlap	Define the minimum percentage a barcode needs to be contained within the Region of Interest (ROI) to decode. For 1D barcodes, this applies only to the horizontal (X) dimension. For 2D barcodes, this applies to horizontal and vertical (X & Y) dimensions.
No Read String	Define the data that should be output if no barcode is read (in place of barcode data in a successful read, as opposed to no output at all).

Table 51 Read DPM and Barcode Settings (Continued)

Settings	Description
Enable Adaptive ROI Search	Enabling Adaptive Region of Interest (ROI) allows the underlying barcode scanning algorithm to alter (shrink and reposition) the area of the ROI for subsequent reads based on the presence of past-read barcodes, providing better-performing barcode reads. The small/medium/large drop-down settings alter the generation of the adapted ROI component.
Barcode String Match	Enable this setting to define the pass/fail criteria based on the contents of the decoded barcode.
Enable Barcode Quality Metrics	Enable this checkbox to enable BQM outputs (Overall Grade).
2D DataMatrix BQM Parameters	Enable or disable parameter options to include in BQM calculations. **NOTE:** These settings are applicable to 2D DataMatrix barcodes only.
Minimum BQM Grade	Enable this option to use a threshold for issuing a pass or fail based on the BQM overall grade for each barcode scan. For example, by setting the threshold between C and D, if a barcode achieves a BQM grade of A, B, or C, the job/tool issues a pass. If the overall grade is a D or an F, the BQM grade is a fail.
Send Decode on Failure	By default, if a barcode is read but has a failing BQM grade (as defined by the **Minimum BQM Grade** threshold above), its data is not output (to either the **Deploy** screen or various outputs such as TCPIP/Serial/USB). When enabled, the decoded data is output even if the BQM Overall Grade is a failing grade, based on the **Minimum BQM Grade** threshold.

Read DPM

Configure DPM decode settings such as Inverse 1D, barcode string match, or ROI search.

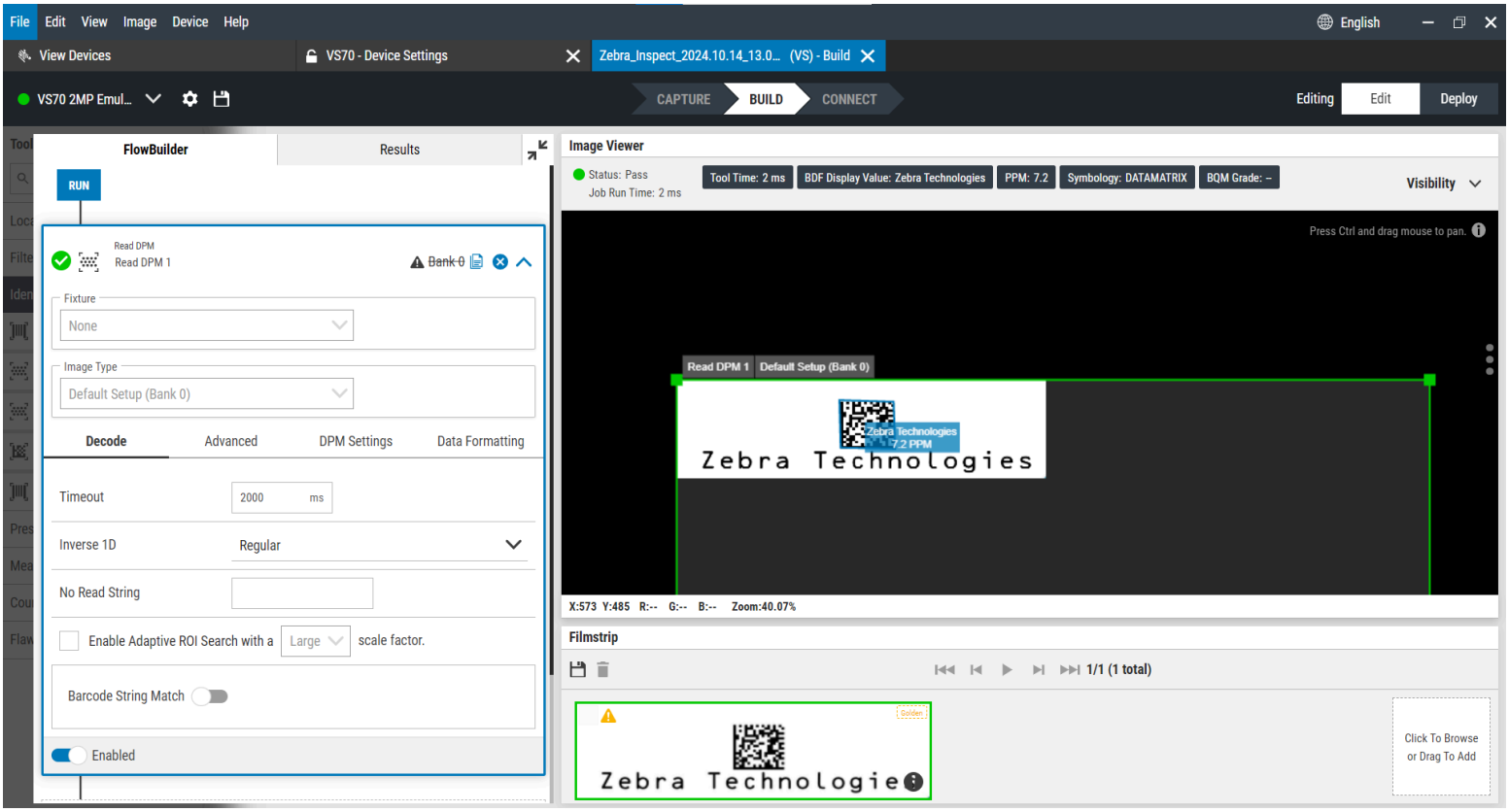


Table 52 Read DPM Settings

Settings	Description
Inverse 1D	Choose the type of barcodes to read.
No Read String	Define the data to be output if no barcode is read.
Enable Adaptive ROI Search	Enabling Adaptive Region of Interest (ROI) allows the underlying barcode scanning algorithm to alter (shrink and reposition) the area of the ROI for subsequent reads based on the presence of past-read barcodes. This provides better-performing barcode reads. The small/medium/large drop-down settings alter how the adapted ROI component is generated.
Barcode String Match	Enable this setting to define pass/fail criteria based on the contents of the decoded barcode.

Read DPM and Barcode

Configure DPM and barcode decode settings such as Inverse 1D, a no-read string, or barcode string match.

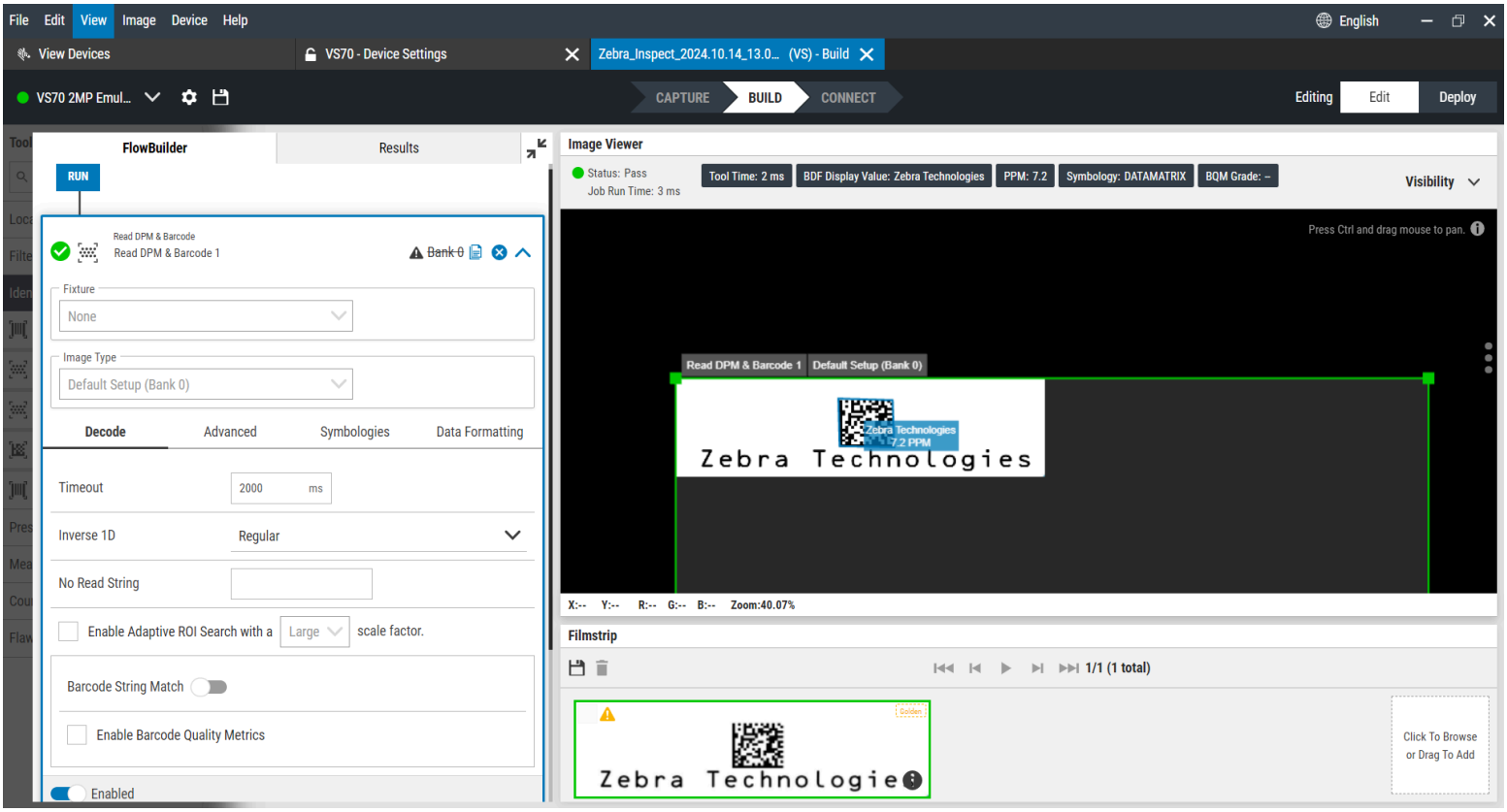


Table 53 Read DPM and Barcode Settings

Settings	Description
Inverse 1D	Choose what type of barcodes you want to be able to read.
No Read String	Define the data that should be output if no barcode is read (in place of barcode data in a successful read, instead of no output at all).
Enable Adaptive ROI Search	Enabling Adaptive Region of Interest (ROI) allows the underlying barcode scanning algorithm to alter (shrink and reposition) the area of the ROI for subsequent reads based on the presence of past-read barcodes. This provides better-performing barcode reads. Drop-down settings alter how the adapted ROI component is generated.
Barcode String Match	Enable this setting to define pass/fail criteria based on the contents of the decoded barcode.

Datacode

Configure Datacode settings such as No Read String, Module Size, or Search Strategy. This tool is helpful for reading 2D barcodes that are difficult to decode and 1D codes with selected symbology.

General Datacode Decode Settings

General Datacode decode settings include defining timeouts, symbologies, no-read string, module size, contrast threshold, polarity, and string match.

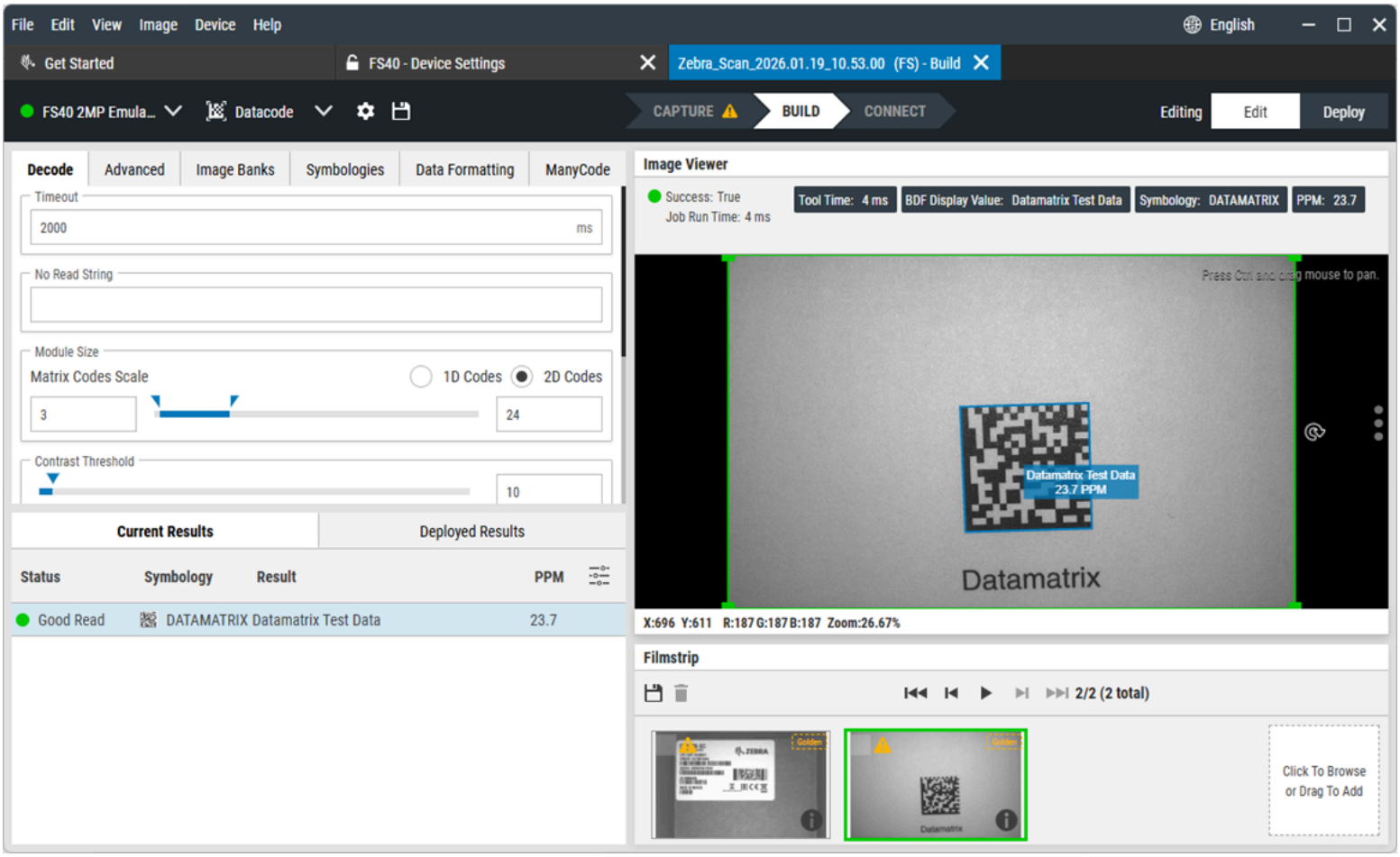


Table 54 General Datacode Decode Settings

Setting	Description
Timeout	Set a time limit to stop the execution of the inspection tool, resulting in a failure.
No Read String	Determine the string that is shown when there are no decoded values.
Module Size	The minimum and maximum module sizes.
Contrast Threshold	The threshold used for Datacode.By default, this is set to 10.

Table 54 General Datacode Decode Settings (Continued)

Setting	Description
Polarity	Select a polarity for the decode: • Any • Bright • Dark
Barcode String Match	Define a string that must be exact or contained within the barcode or datacode output for the tool to pass: • Disabled - no match string. • Exact String - string must match the exact string provided. • String Contains - code must contain the match string. • RegEx - enable to input complex match string conditions.

Advanced Datacode Decode Settings

Advanced Datacode decode settings include four sections: Detection, Matrix Codes, DPM, and Linear Codes. Detection settings include defining a detection strategy, a decoding strategy, a detection method, a finder tradeoff, allowing distortion, a perspective level, a code slant, background suppression, and codes with broken edges. Matrix Codes settings include defining code sizes (row and column count), mirrored codes, rectangular codes, and codes with broken corners. DPM settings include defining a mixed polarity, grid-based detection. Linear Codes settings include defining a length, high confidence, high blur, damage, no quiet zone, and addons.

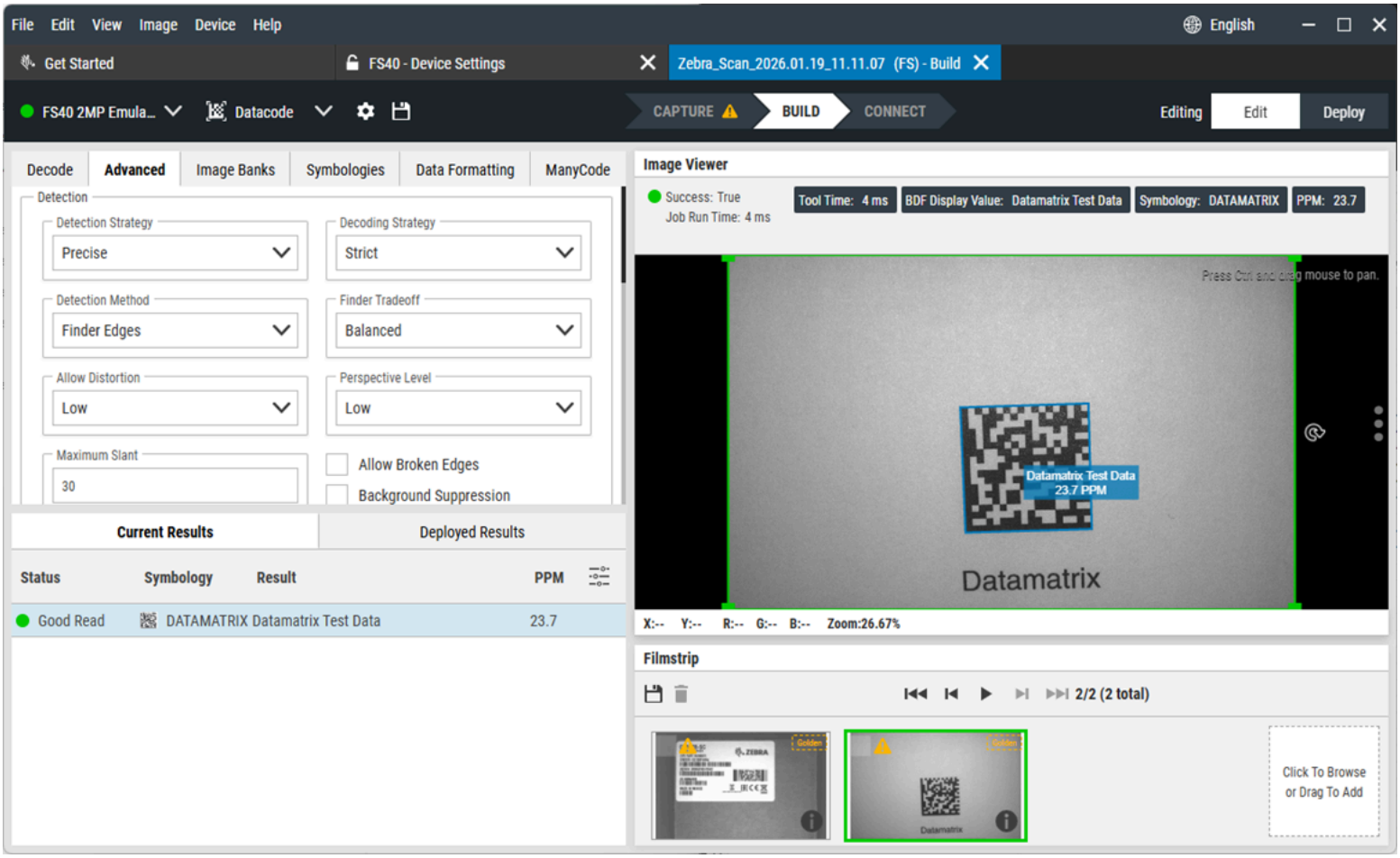


Table 55 Advanced Datacode Decode Settings

Setting	Description
Detection Strategy	Specify the step of image downsampling used for finding codes at different scales: · Super Fast · Fast · Precise · Strict · Extended
Decoding Strategy	Specify the precision of outline detection: · Super Fast · Fast · Precise · Strict · Extended

Table 55 Advanced Datacode Decode Settings (Continued)

Setting	Description
Detection Method	Specify how code candidates are located: • Finder Pattern • Finder Edges
Finder Tradeoff	Decide how to optimize parameters for weak or jagged edges: • Extra Sensitive • Sensitive • Balanced • Robust • Extra Robust
Allow Distortion	Specify a distortion level: • None • Low • Medium • High • Auto (slower)
Perspective Level	Specify a perspective distortion of the code grid: • None • Low • High
Maximum Code Slant	Specify the maximum deviation from the right angle in the corner of the Finder Pattern.
Expected Gap Size	Specify the distance between neighboring modules. • Zero • Small • Medium • Large
Maximum Code Slant	Specify the maximum deviation from the right angle in the corner of the Finder Pattern.
Allow Broken Edges	Employs an extended edge tracker that is able to skip missing pixels. Applies primarily to the FinderEdges method.
Background Suppression	Turns on an additional preprocessing step that removes low-contrast background areas.

Table 55 Advanced Datacode Decode Settings (Continued)

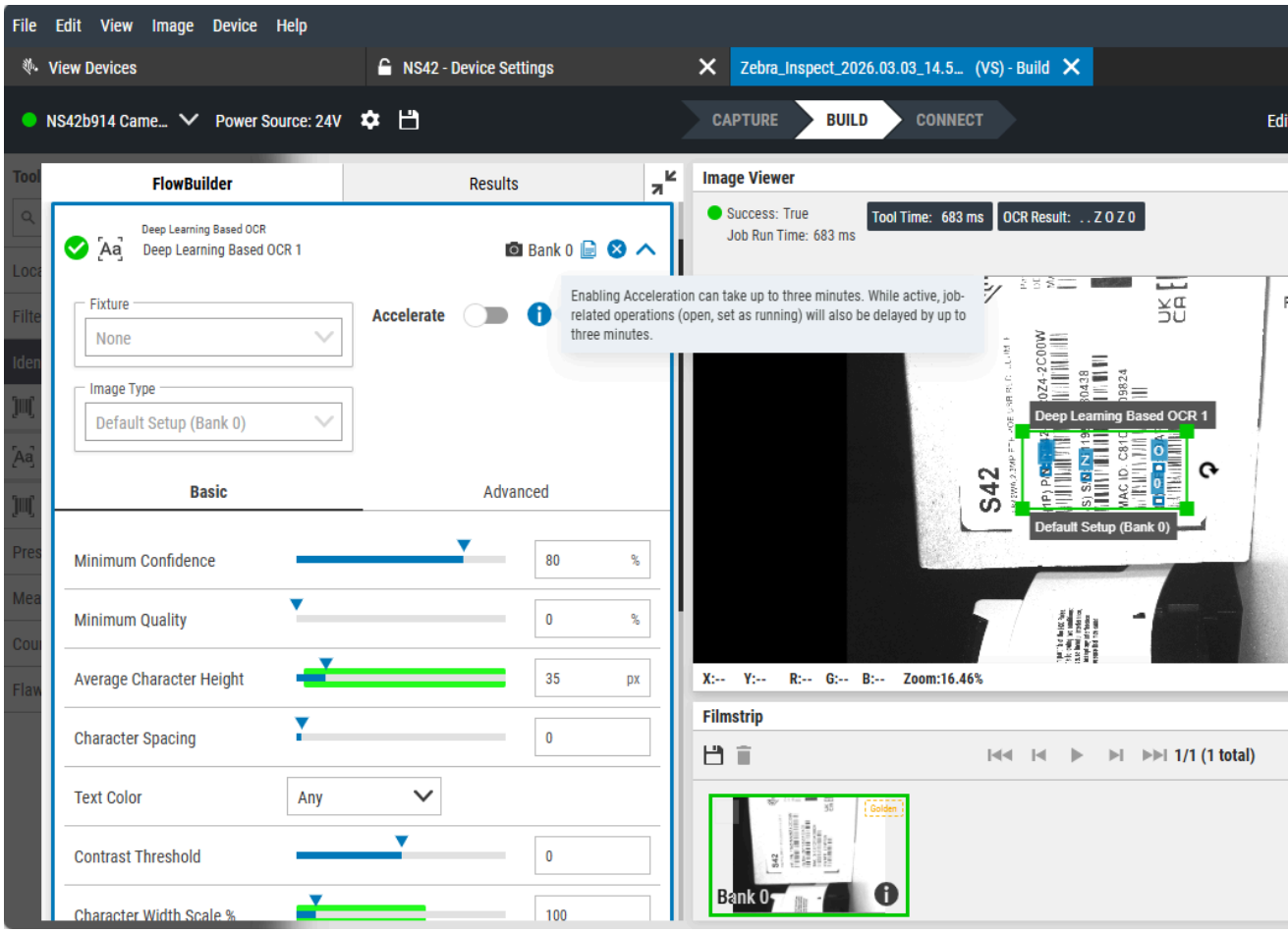
Setting	Description
Code Sizes	Limit the number of Data Matrix modules in a row and column.
Allow Mirrored	Allows codes to be mirrored/transposed.
Allow Rectangular	Allows codes to be rectangular
Allow Broken Corner	Tries to create a candidate even if the bottom-left corner of a Data Matrix code is occluded.
Enable DPM	Enables an algorithm for direct part marking.
Allow Mixed Polarity	Enables reading modules with both bright and dark pixels simultaneously.
Grid-based Detection	Enables reading codes with highly damaged Fixed Pattern (slow)
Dot Size	Optimizes the algorithm parameters for small, medium, or large dots, or for fully-printed (normal) codes. Applies to DotGrid and FinderPattern methods. Available options: • Small • Medium • Large • Full • Auto
Length	Barcode Length: • Any • Discrete - select a specific number as the barcode length. • Range - select a range of numbers as the barcode length.
Extra High Confidence	Enables filtering out detected codes with low score.
Allow High Blur	Enables high-blur digitizer.
Allow Damage	Enables reading of barcodes with dirt, holes or occlusions.
Allow No Quiet Zone	Enables reading codes without the obligatory empty areas before and after the bars.
Allow Add ons	Enables EAN and UPC add-ons 2 and 5

Using Accelerated Tools

Deep Learning OCR and Anomaly Detection tools can be accelerated to speed up a specified tool during job execution and reduce overall tool time.

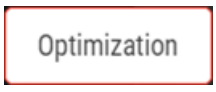
NOTE: Acceleration is available on FS42 and NS42 devices only.

1. Enable acceleration by sliding the **Accelerate** toggle to the right.



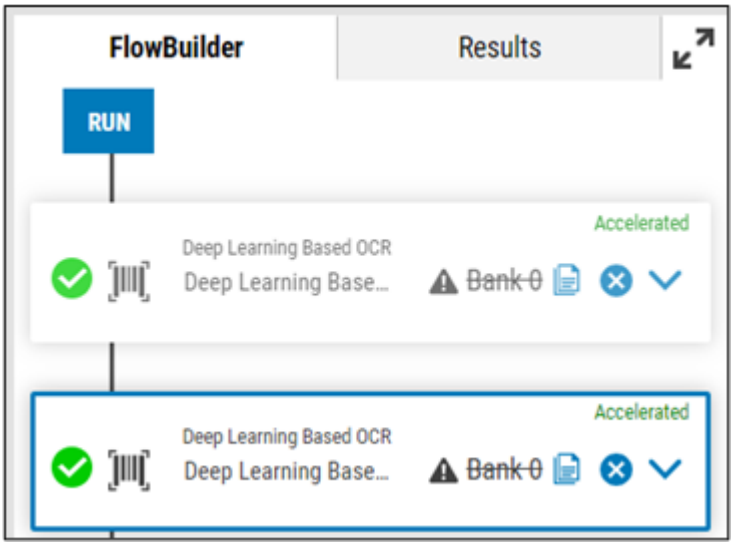
NOTE: In this mode, the Region of Interest (ROI) can be moved or rotated but not resized. Tool settings **Average Character Height** and **Character Width Scale %** are not editable while in Accelerate mode. Move the slider to the left to disable Accelerated mode and edit tool settings.

2. After **Accelerate** is enabled, the **Optimization** overlay displays, and all settings are blocked and transferred to the device.



NOTE: For optimal performance, jobs should contain only one accelerated tool. Jobs containing more than one accelerated tool can negatively impact performance and increase execution time.

3. All accelerated tools are marked on the **FlowBuilder** by the green **Accelerated** indication.



NOTE: If a job is saved with the **Accelerate** setting enabled and later opened on a device that does not support this setting, it is automatically switched back to its standard state without acceleration.

Deep Learning Optical Character Recognition (OCR)

The Deep Learning OCR tool reads text from images using Deep Learning.

This tool locates and recognizes characters without additional training:

- Horizontally-oriented
- Height between 85% and 115% of Char Height (in pixels)
- Contain Latin letters (upper case or lower case), digits, or one of: !#$%&()*+,-./:;<=>?@[]^_`{|}~'"\€£¥

Use the Deep Learning OCR region of interest to limit the analyzed area, which can lead to improved performance. Moreover, it may be used to adjust text not displayed horizontally.

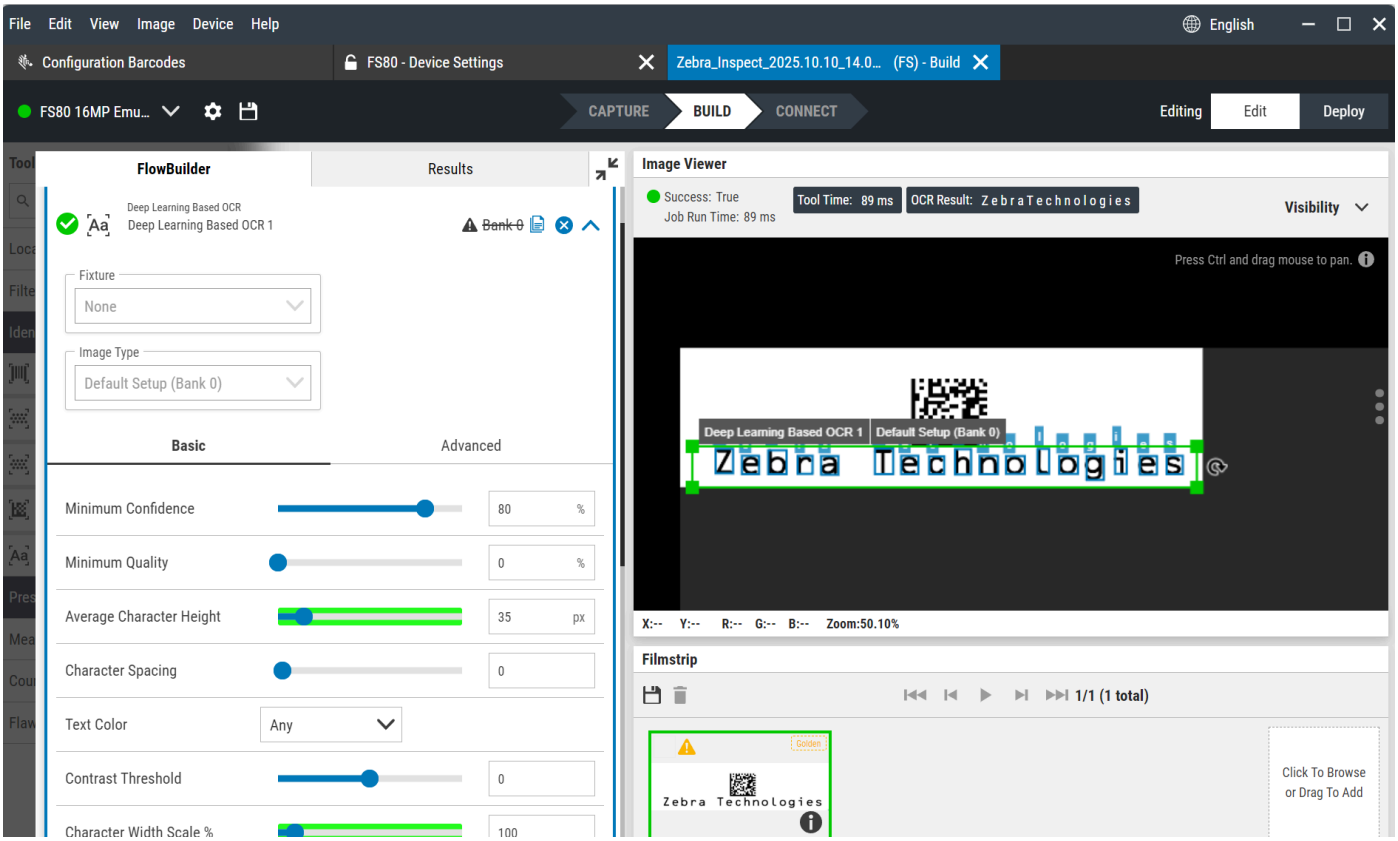


Table 56 Deep Learning OCR Settings

Setting	Description
Minimum Confidence	The Minimum Confidence parameter may change a character's minimum score. By default, this threshold is set to 80%. **NOTE:** Adjusting this value lower can help include decodes of slightly lesser confidence to get the desired output. For example, the above example is too strict and does not decode various characters. Lowering this value from 99 to 98 yields a favorable result.
Minimum Quality	Sets a minimum required quality for a character to include in the algorithm or output.

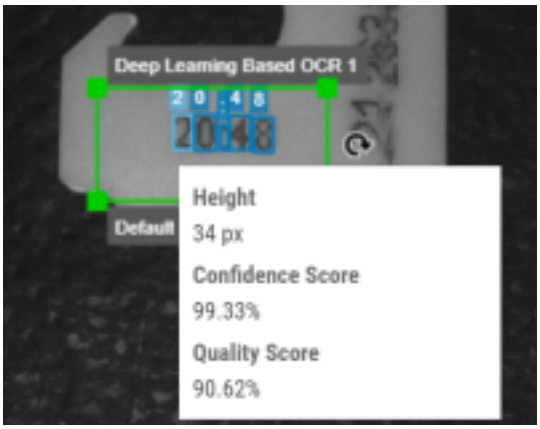
Table 56 Deep Learning OCR Settings (Continued)

Setting	Description
Average Character Height	The Average Character Height should be set to the average height of characters (specifically, capital letters) in the analyzed area. For example, if an image contains two kinds of characters: one is 24 pixels high and the second is 40 pixels high, the Average Character Height should be set to 32, irrespective of the number of characters of each kind. Use the slider to select a value close to the average pixel height of text trying to be decoded. Performance improves when this value is closer to the average character height (in pixels). • Minimum Value: 8 • Default Value: 25 • Maximum Value: 200
Character Spacing	Distance between characters is denoted as a fraction of CharHeight.
Text Color	Set a required polarity for a character to be returned. • Bright: Only characters with contrast greater than Contrast Threshold are returned. • Dark: Only characters with contrast lower than Contrast Threshold are returned. • Any: Only characters with contrast lower than Contrast Threshold or greater than Contrast Threshold are returned. Default Value: Any
Contrast Threshold	Sets a threshold for a contrast of found characters. Default value: 0
Character Width Scale Percentage	Scales the image width by the given factor (%). • Minimum Value: 10% • Default Value: 100% • Maximum Value: 1000%
Character Range	Enable Character Range to limit the set of recognized characters. This string must be formatted according to the following rules: • Allowed characters must be separated with commas. • For ease of use, a continuous range of letters or digits may be written as starting_character-ending_character, for example: A-Z or 1-6. • Comma and backslash have to be prepended with a backslash. For example, Character Range equal to A-F,g-o,0-9,X,Y,Z,-,\\,\, recognizes only ABCDEFXYZghijklmno0123456789-\, characters.

Table 56 Deep Learning OCR Settings (Continued)

Setting	Description
String Match	Defines a string that must be contained within the decoded OCR output for the tool to pass (for example, a substring). By default, the tool passes if the match string is contained in the resulting OCR output. This allows additional characters to be added before and after the string. Enable the RegEx checkbox and use the syntax `^<stringToExactMatch$` to implement the exact match behavior. RegEx can also be used for complex string match logic. Refer to the RegEx section for more details.
Timeout	Sets a time after which the tool fails.
Invert Pass/Fail Result	Flips the results of this tool.

NOTE: Obtain character height, confidence score, and quality score by hovering over the characters of the decoded image.



Advanced Deep Learning OCR Settings

Configure Advanced Deep Learning OCR settings such as character gap percentage, vertical misalignment percentage, minimum characters to create a line, flatten, and grammar rules.

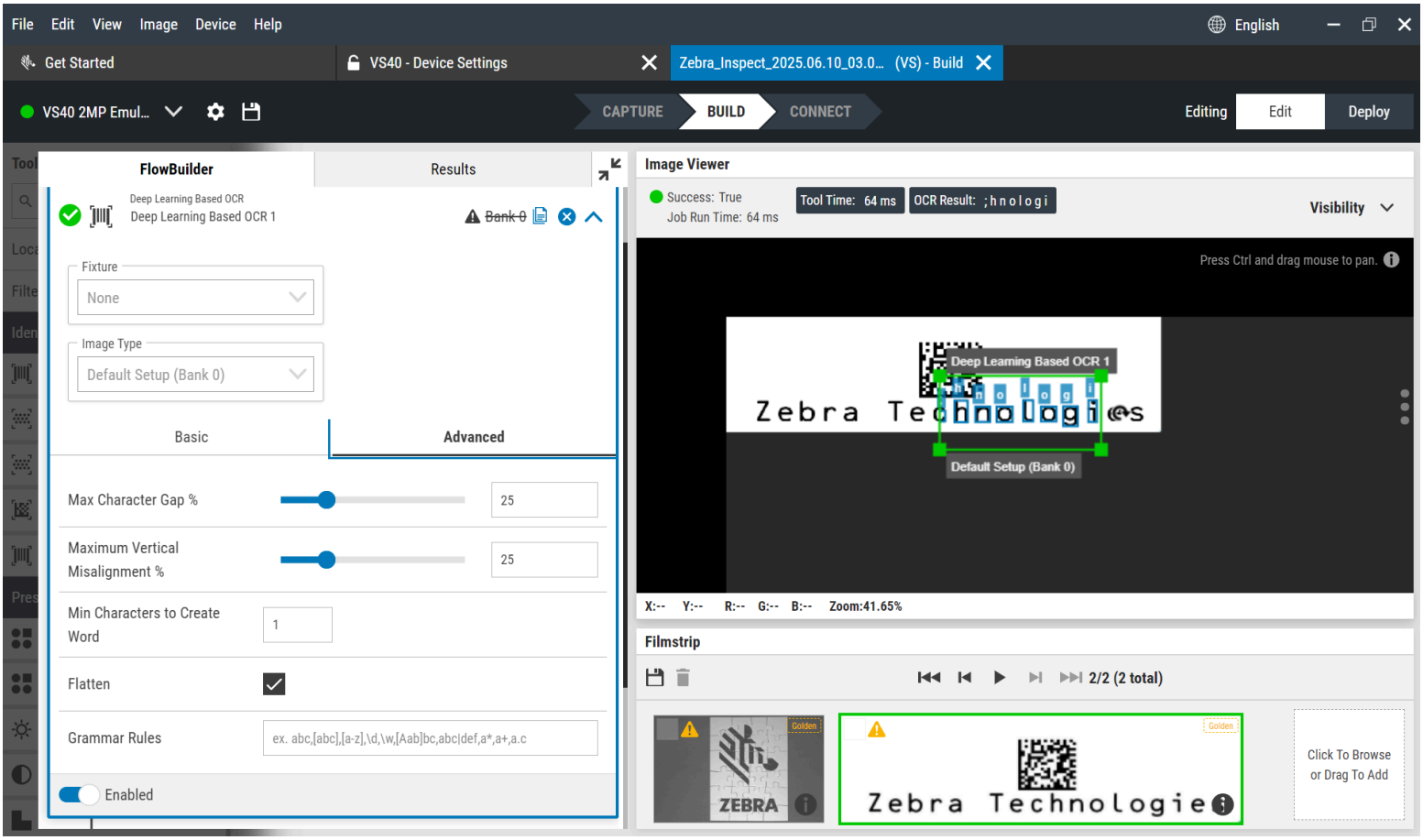


Table 57 Advanced Deep Learning OCR Settings

Setting	Description
Maximum Character Gap Percentage	The maximum horizontal gap between joint characters' boxes is denoted as a percentage of A character height.
Maximum Vertical Misalignment Percentage	The maximum vertical misalignment between joint character boxes is denoted as a percentage of A character height.
Minimum Characters to Create a Line	Determine the minimum number of characters to create a line. Default value: 1 **NOTE:** A line is commonly referred to as a Block or Word for the utilization of this tool.

Table 57 Advanced Deep Learning OCR Settings (Continued)

Setting	Description
Flatten	If true, this feature concatenates the words on the line into a single result string. Otherwise, each word is a separate result string.
Grammar Rules	Use grammar rules to check the structure of the text for grammar, character, and formatting constraints, and abbreviations or acronyms.

Pattern Elements detected by Grammar Rules can be:

- Individual Characters - Escaping operational characters: Use a backslash to treat operational characters as normal: \\, *, \?, \., \+, \-, \], \[, \), \(.
- Character Class - a set of characters enclosed within square brackets []. It allows you to match any one character from the specified set. For example:
  - List of characters: [abc] Matches any one of the characters a, b, or c.
  - Range: [a-z] Matches any one of the characters from a to z.
  - Mix of them: [a-zA-Z12] Matches any one of the characters from a to z, A to Z, and 1,2.
  - Predefined character classes:
    - \d is equivalent to [0-9]
    - \w corresponds to [a-zA-Z0-9_]
    - . (dot) matches any single character (\w plus special characters)

NOTE: Inside a character class, the following characters require escaping with a backslash //, \\, \-, \]. For example, [a.*|] is a valid pattern that matches the characters: a,.,*,|.

- Chain - an extended string created by concatenating individual characters and character classes. For example:
  - abc - matches text abc
  - [Aa]bc - matches texts: abc and Abc
  - \dabc - matches texts: 0abc, 1abc, ..., 9abc
- Alternative - used to match one pattern or another. It's a sequence of chains separated by the pipe symbol | in round brackets (). For example:
- Special Operators - can modify or repeat the preceding expression. For example:
  - * (star): means zero or more occurrences of the preceding expression (in particular, ".*" means any sequence), but tries to match as many characters as possible.
  - + (plus): means one or more occurrences of the preceding element, maximizing the number of characters matched.
  - ? (question mark): means zero or one occurrence of the preceding element, with a preference for one.
  - *? (lazy star): means zero or more occurrences of the preceding expression, but tries to match as few characters as possible.
  - +? (lazy plus): means one or more occurrences, but minimizes the number of characters matched.

NOTE: Special operators can be added after the pattern element; for example: [ABC]*, [0-9]?, (ABC|DEF)+. However, special operators cannot be used inside alternative pattern elements.

Presence/Absence Tools

Use Presence/Absence tools to detect specific objects or patterns within a Region of Interest (ROI).

Object Presence Absence

The Object Presence Absence tool verifies the presence or absence of a specific pattern or object within a Region of Interest (ROI).

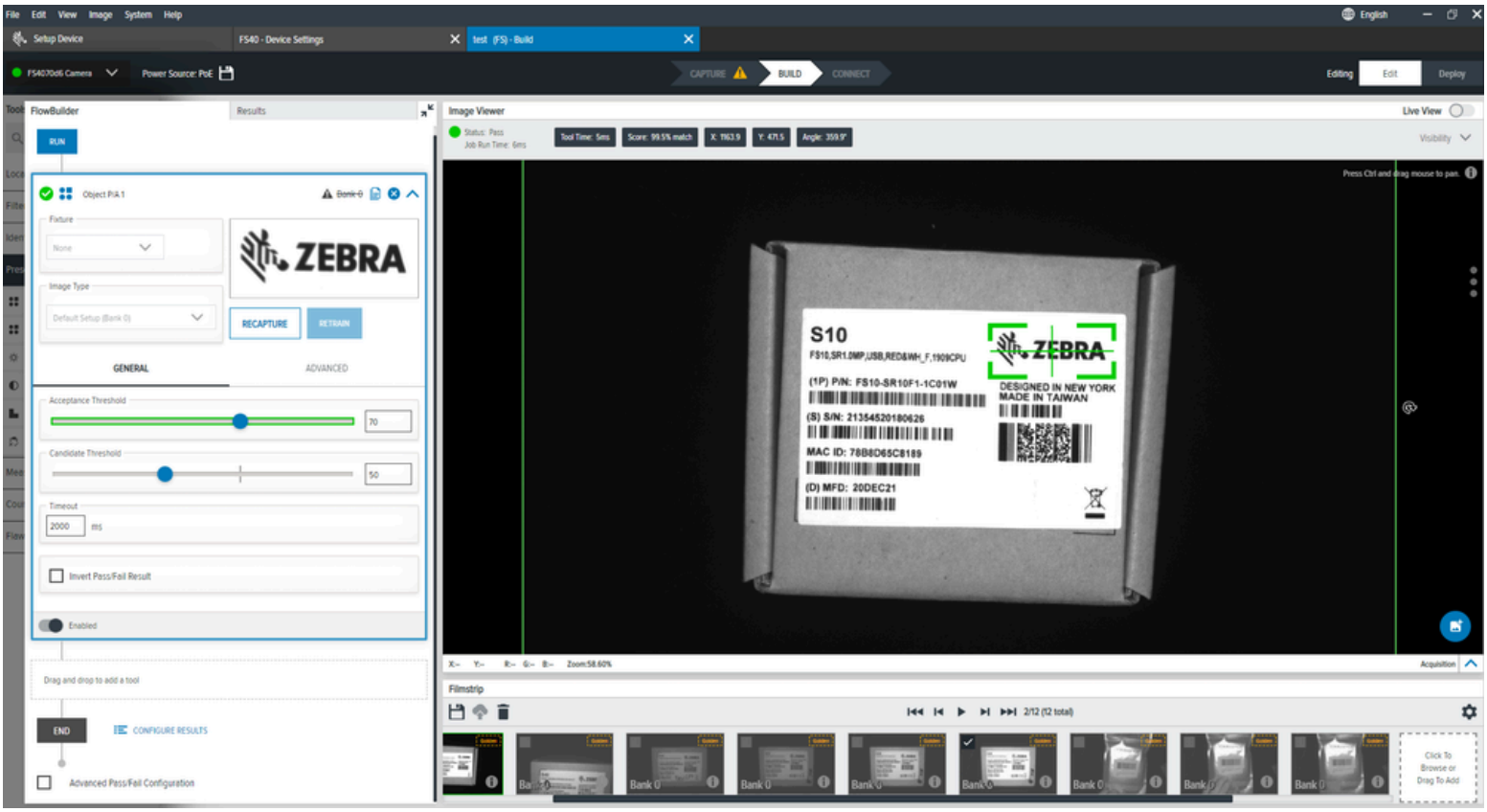


Table 58 General Object Presence Absence Settings

Settings	Description
Acceptance Threshold	Select a minimum match score required to be considered a passing match. If multiple object match scores exceed this minimum, the highest matching object score is used as the final match.
Candidate Threshold	The threshold for a match of the trained pattern to a pattern in the acquired image.

Advanced Object Presence Absence Settings

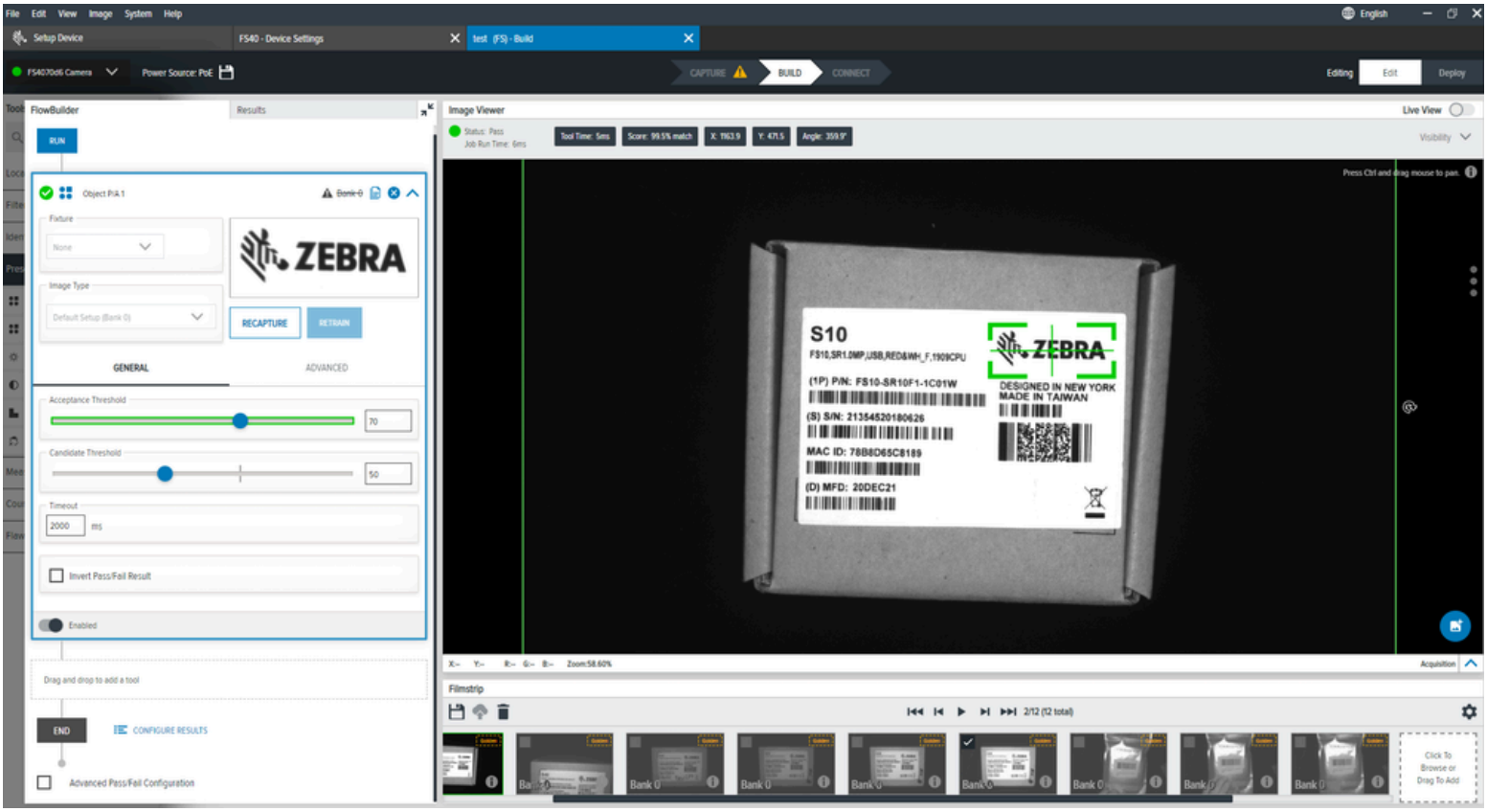
Configure Advanced Object Presence Absence settings such as performance control, noise, rotation threshold, and scale deviation.

Table 59 Advanced Object Presence Absence Settings

Settings	Description
Performance Control	A selector to choose the best coarseness and search type combinations for a more performant search.
Noise	This filter removes pixel-level noise but preserves edge data.
Rotation Threshold	A minimum match score is required to be considered a passing match. If multiple object match scores are above the minimum, the highest matching object score is not used as the final match.
Allow Scale Deviation	When enabled, this setting allows the location of objects slightly smaller or larger than the object used during the creation of the model.

Object Plus Presence Absence

The Object Plus Presence Absence tool finds specific patterns in a Region of Interest (ROI) and evaluates them based on advanced characteristics such as minimum edge contrast and scale factor.



Settings	Description
Acceptance Threshold	The minimum match score required to be considered a passing match. If multiple object match scores exceed this minimum, the highest matching object score is used as the final match.
Candidate Threshold	The threshold for a match of the trained pattern to a pattern in the acquired image.

Brightness

The Brightness tool calculates the average pixel brightness value in a Region of Interest (ROI).

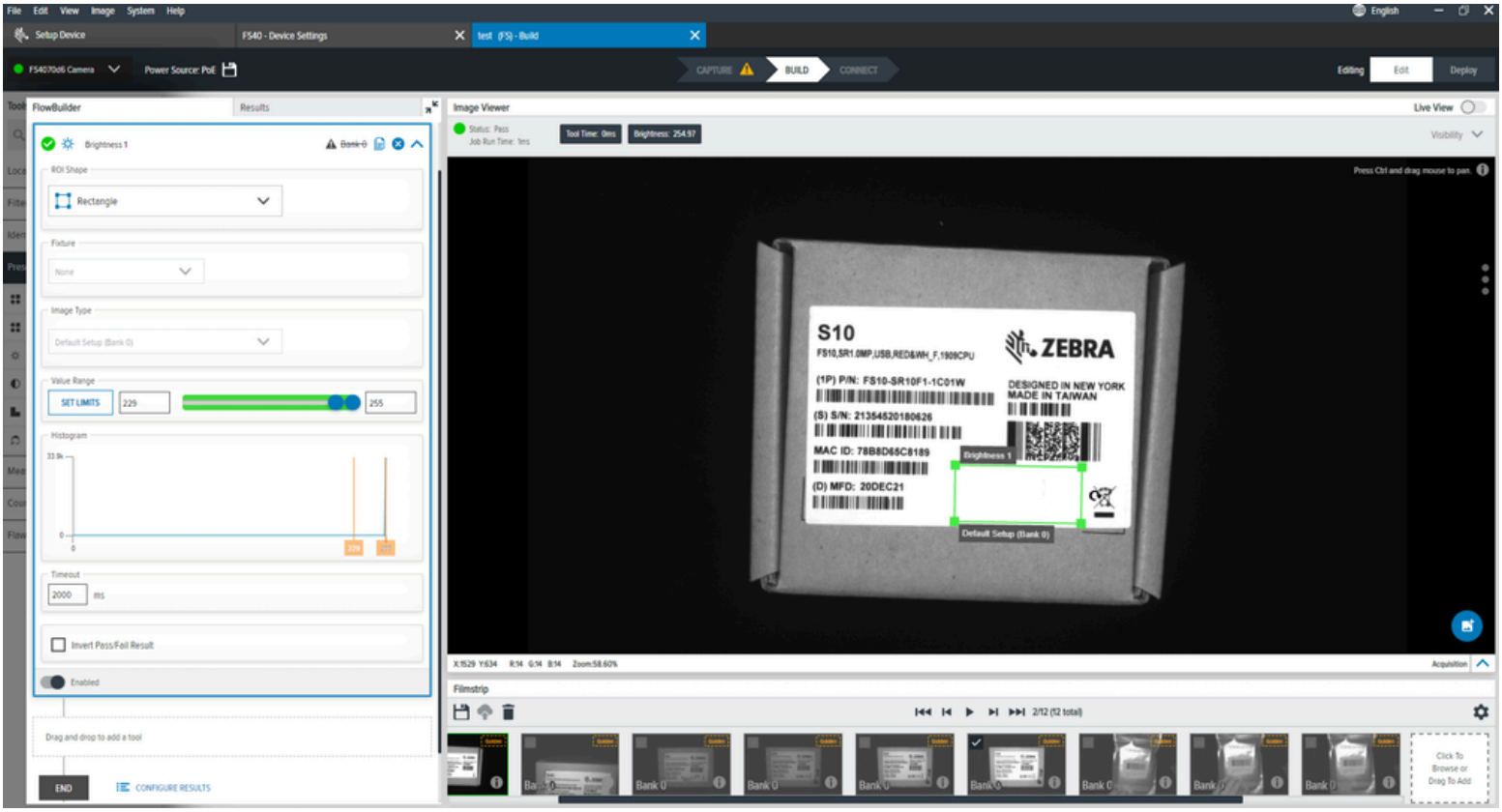


Table 60 General Brightness Settings

Settings	Description
Fixture	Determine the fixture for the ROI shape.
Image Type	Select which image this inspection tool should use.
Value Range	Set the minimum and maximum values.
Histogram	Shows the number of pixels at each value.

Contrast

The Contrast tool calculates the maximum and minimum pixel intensity difference in a Region of Interest (ROI).

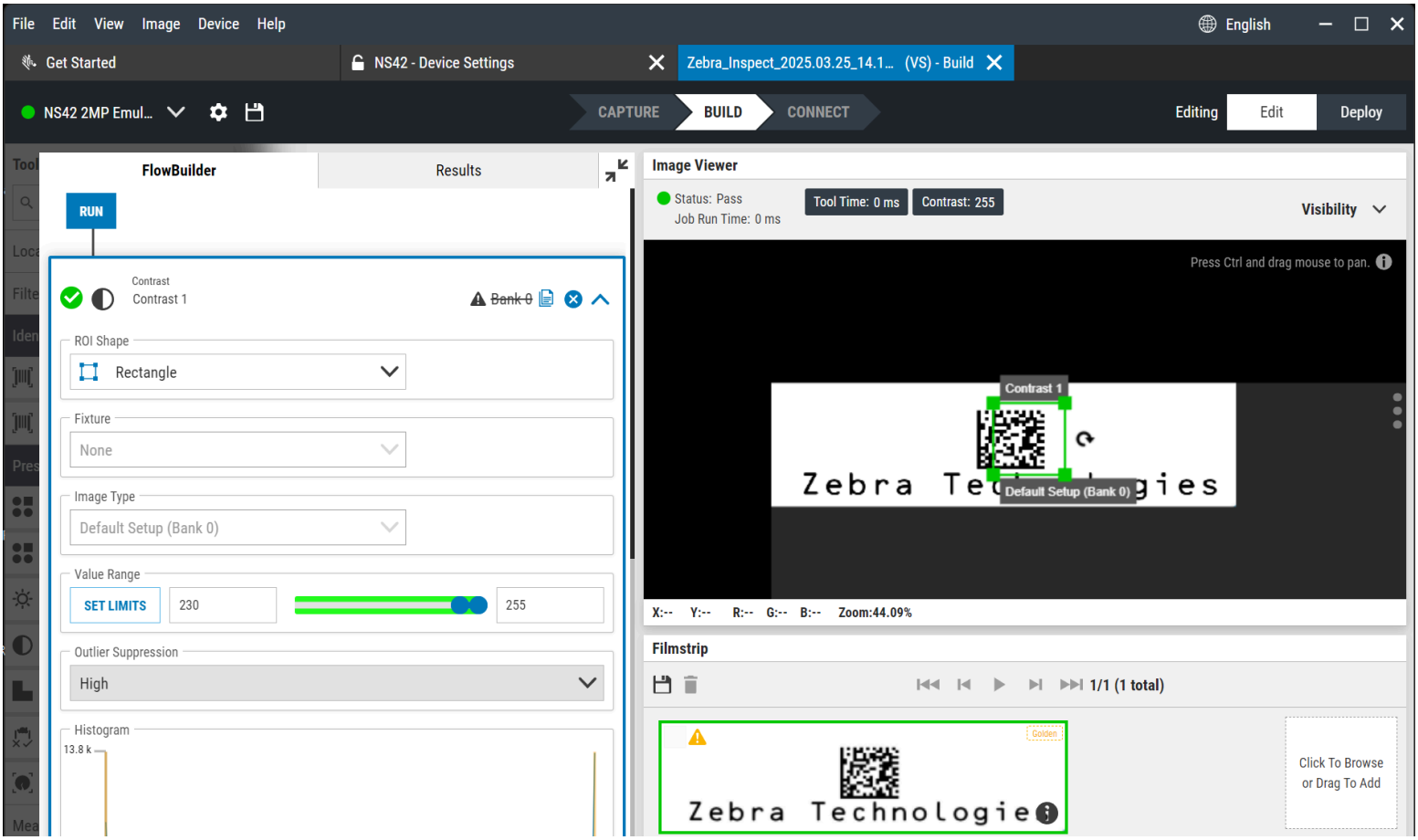


Table 61 Contrast Settings

Settings	Description
ROI Shape	Select a Rectangle or Circle shape for the Region of Interest (ROI).
Fixture	Select the output from a locate or scanning tool's previous result to position the current tool's Region of Interest (ROI).
Image Type	Select which image this inspection tool should use.
Value Range	Set the minimum and maximum values.
Outlier Suppression	Specify the number of extreme pixel values to exclude when calculating the final contrast. Its meaning is the percentage of outliers (noise) we suppress while calculating the contrast value.
Histogram	Shows the number of pixels at each value.

Using Outlier Suppression

Outlier Suppression is the percentage of the noise suppressed when calculating the contrast value. The following histograms provide visual examples of different levels of suppression:

Figure 3 High Suppression (5%) / Contrast: 20

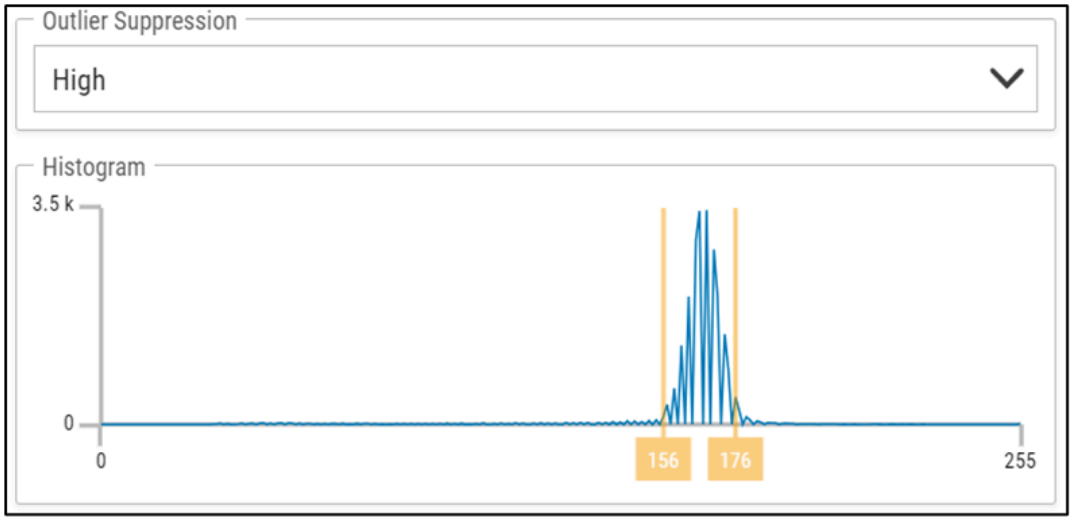


Figure 4 Medium Suppression (2%) / Contrast: 83

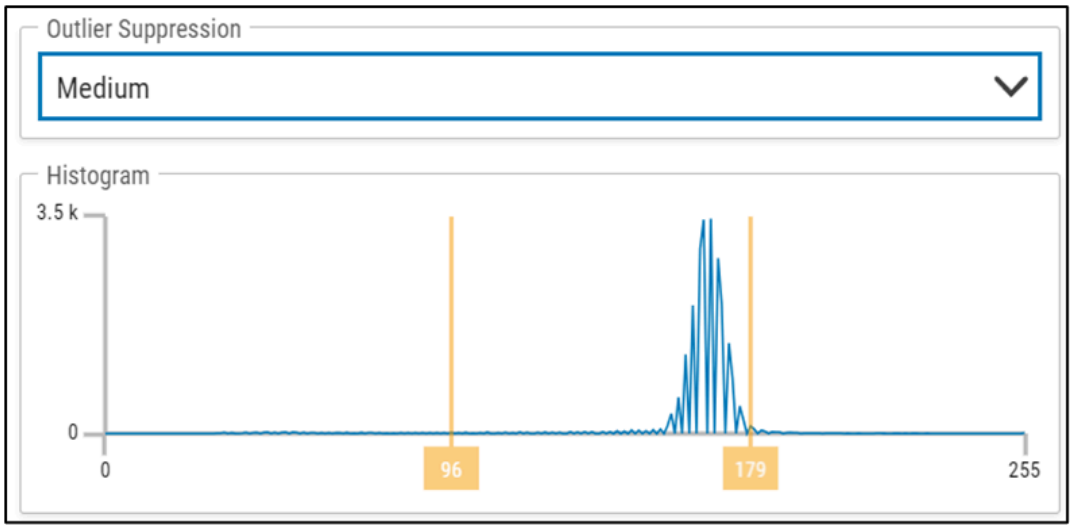


Figure 5 Low Suppression (1%) / Contrast: 180

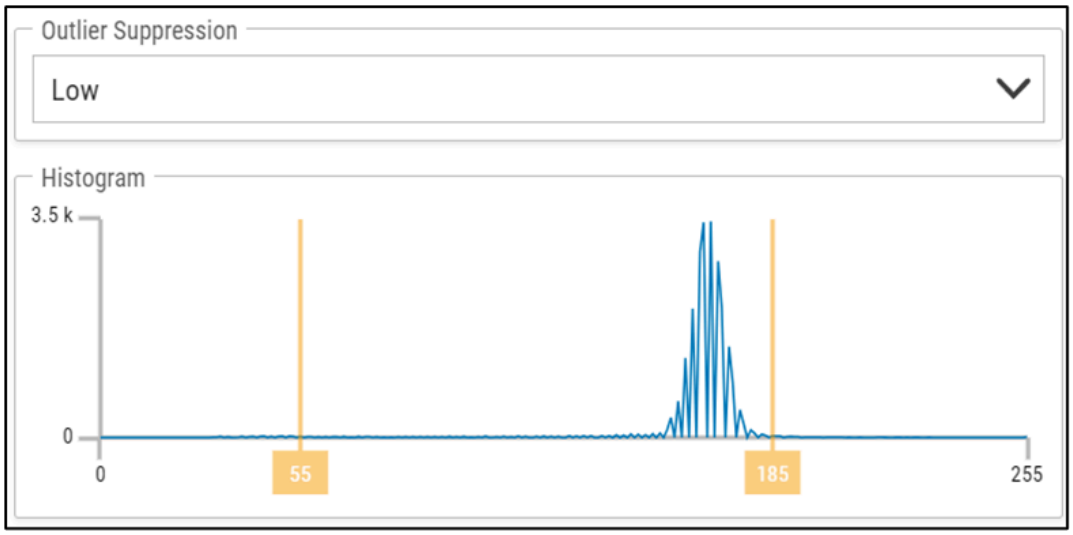
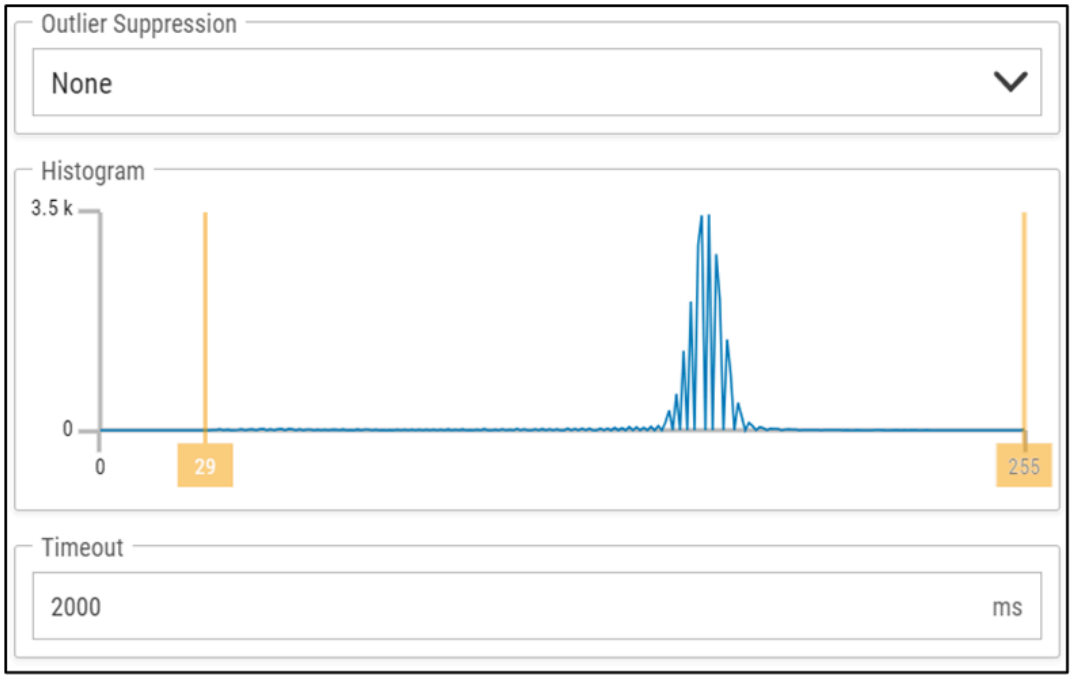


Figure 6 No Suppression (0%) / Contrast: 226



Edge Detect

The Edge Detect tool identifies transitions based on the highest contrast in a Region of Interest (ROI).

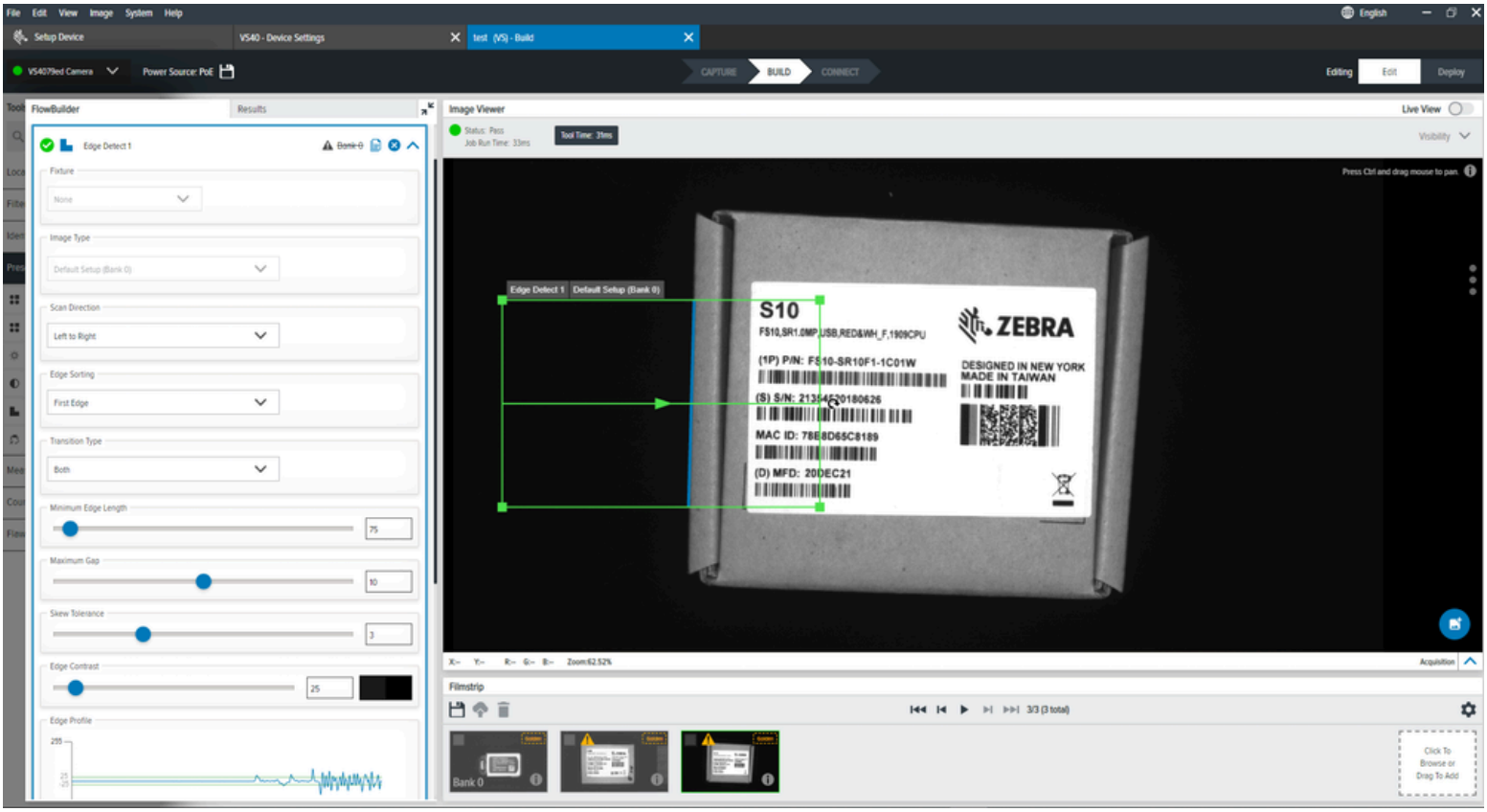


Table 62 Edge Detect Settings

Settings	Description
Fixture	Select a previous locate tool result to position this tool's Region of Interest (ROI).
Image Type	Select which image type this tool uses.
Scan Direction	Set the direction that the tool uses when searching for edges.
Edge Sorting	Refine a collection of found edges to a single best edge.
Transition Type	Select the type of edge transition used to find the edges.
Minimum Edge Length	The minimum length in pixels of an edge to use.
Maximum Gap	The maximum size in pixels to consider in gaps of the edge.
Skew Tolerance	Degrees to attempt to match the skew of the line if it is not straight.
Edge Contrast	Threshold acceptance to consider a line.

Table 62 Edge Detect Settings (Continued)

Settings	Description
Edge Profile	Displays the contrast score of the features across the Region of Interest (ROI).

Blob Presence Absence

The Blob Presence/Absence tool verifies a blob's presence or absence within a Region of Interest (ROI).

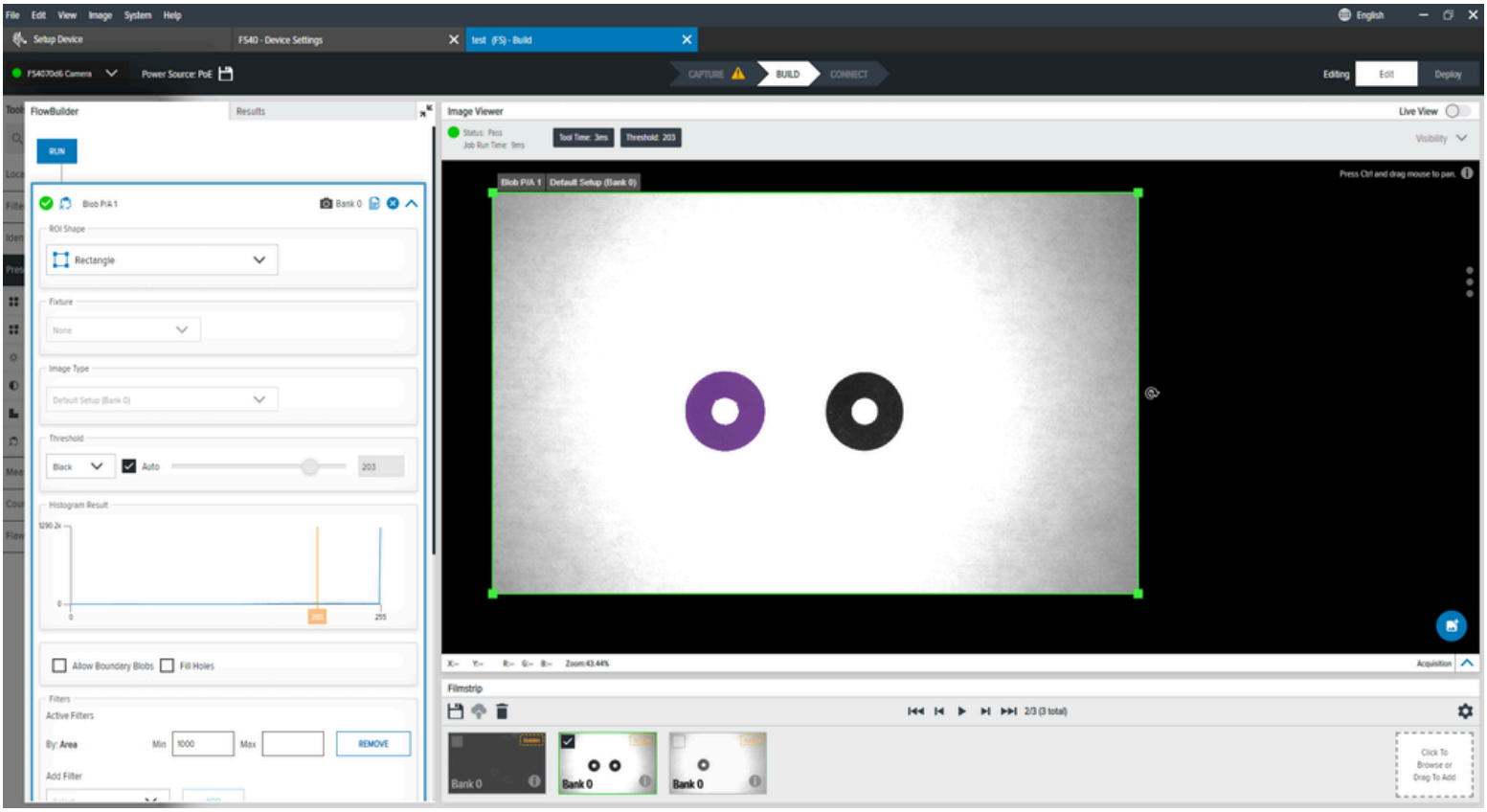


Table 63 Blob Presence Absence

Setting	Description
Image Type	Select which image this tool should use.
Threshold	The algorithm detects the best threshold for the tool to use.
Histograph Result	Sets the number of pixels at each value.
Filters	Apply filters to set the criteria used to consider a blob valid for this tool.
Sorting	Define the priority for selecting a blob to return.

Using Anomaly Detection

Aurora Deep Learning Editor is required to run Anomaly Detection in Aurora Focus. It requires a license stored on the device or a valid emulator license key.

NOTE: Anomaly Detection is available on FS42 and NS42 devices only.

Acquiring Images

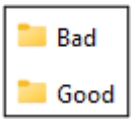
Images can be acquired and saved via FTP or directly onto the device.

1. If you acquire the images live in an industrial environment, use Image Saving via FTP to copy the image from the FTP server to your PC. For more information on setting up image saving, refer to the FTP File Saving section in the Aurora Focus User Guide.
2. If you save the images directly onto the device, access them using the Aurora Focus Web HMI. For more information, refer to the Accessing the Web HMI section in the Aurora Focus User Guide.

Preparing Images

It is important to properly organize and label the images when training the model.

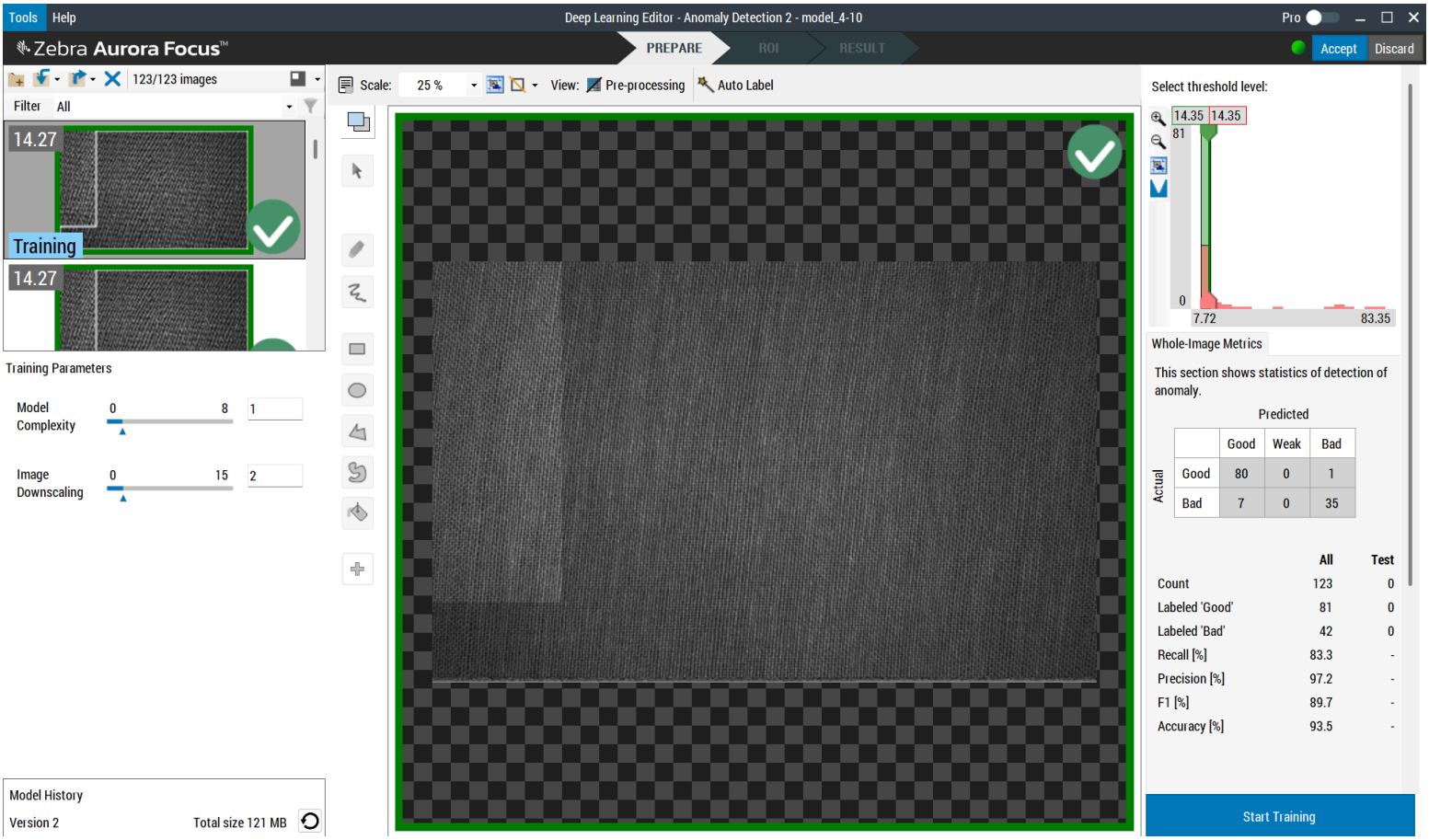
1. Collect a series of images using periodic single-shot mode in Aurora Focus.
2. Save the images to the local host/PC.
3. Create separate folders for Bad and Good images for testing purposes.



Using Aurora Deep Learning Editor

Aurora Focus and Deep Learning Editor are directly connected and require compatible versions.

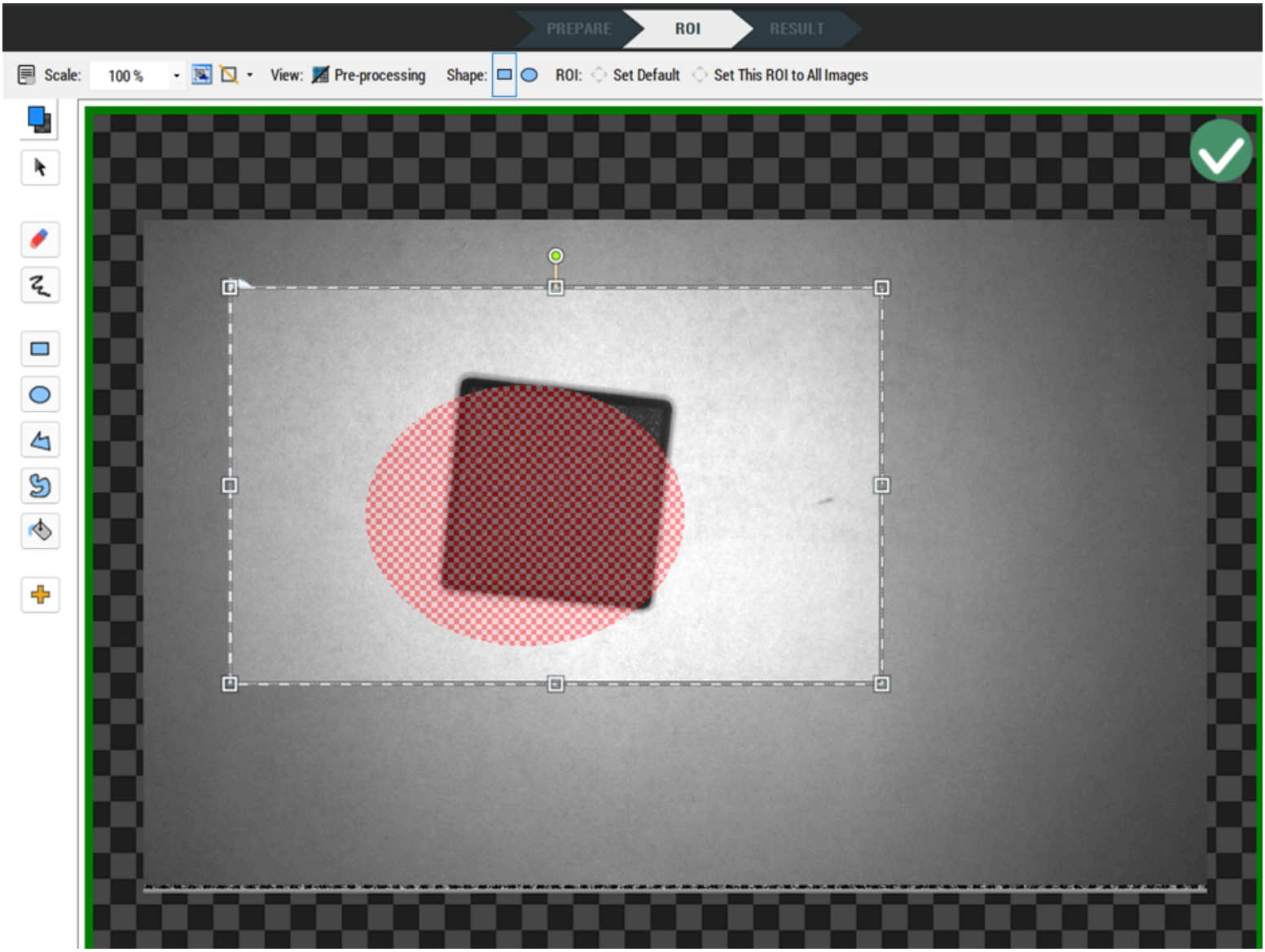
1. Click  to add new images to the editor, or click  to add a new image and mark it as good or bad.



2. Observe the **Training Parameters** to understand the complexity and downscaling of the model. Both contribute to the training time and the working time of the model.
   - Model Complexity - as density increases, the precision of the heatmap data visualization also increases, which in turn increases execution time.
   - Image Downscaling - each level reduces the width and height of the image by a factor of 2.

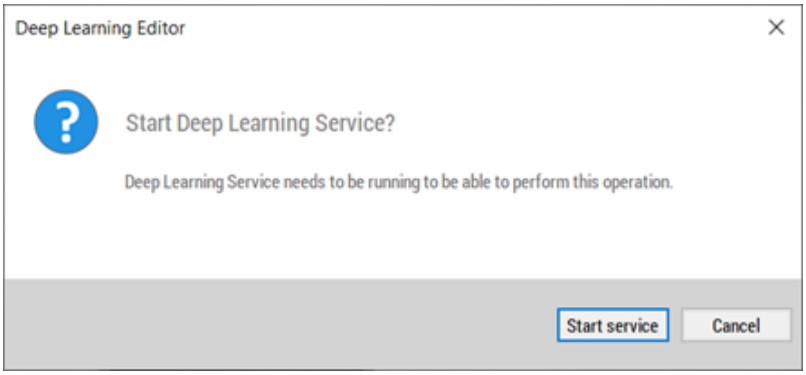
NOTE: Go to Using Pro Mode to understand how to use advanced parameters.

3. Use the **ROI** tab to configure the Region of Interest (ROI). The size, shape, and location of the ROI can be changed to fit every image. Drawing tools help to mark the quiet zone inside the ROI. Fields outside the marked ROI are not calculated.



4. After configuring the settings, click **Start Training** to start training the model.

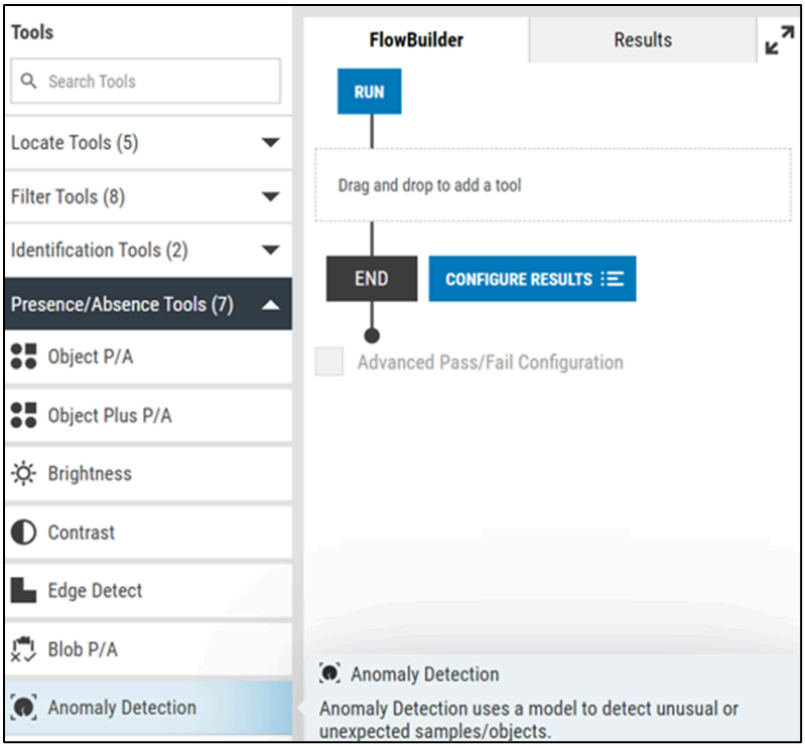
NOTE: If the Deep Learning Editor is being run for the first time, the first click launches the Deep Learning Service, and the second starts generating the model. The Deep Learning Service runs for approximately 30 minutes after disconnecting the device or stopping the emulator.



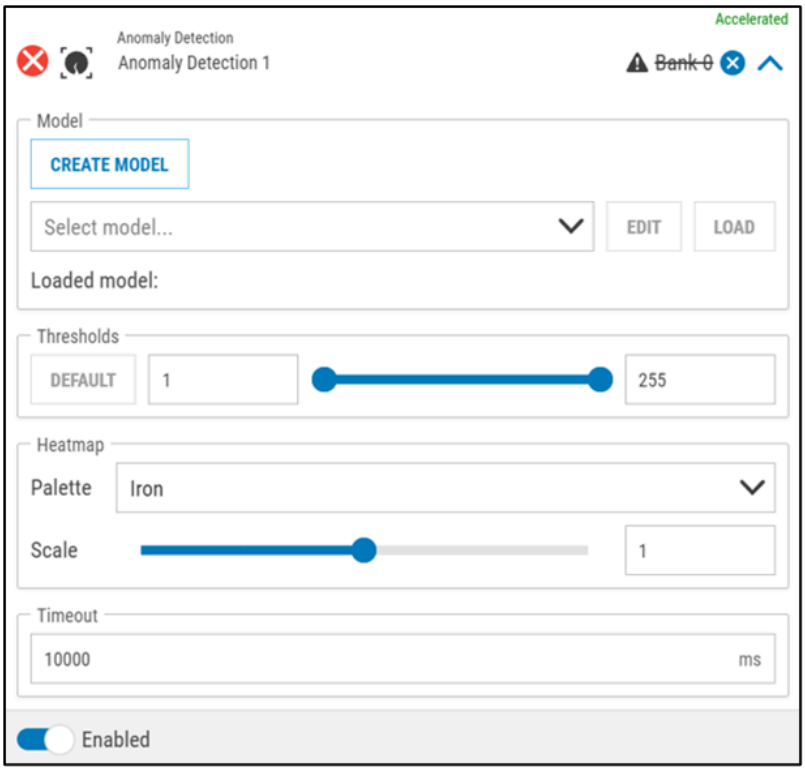
Creating a Model

The Anomaly Detection tool in Aurora Focus uses the Aurora Deep Learning Editor.

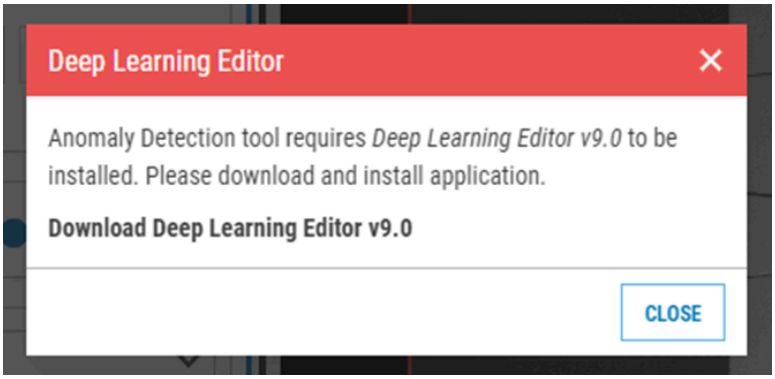
1. Add the **Anomaly Detection** tool from **Presence/Absence** tools list.



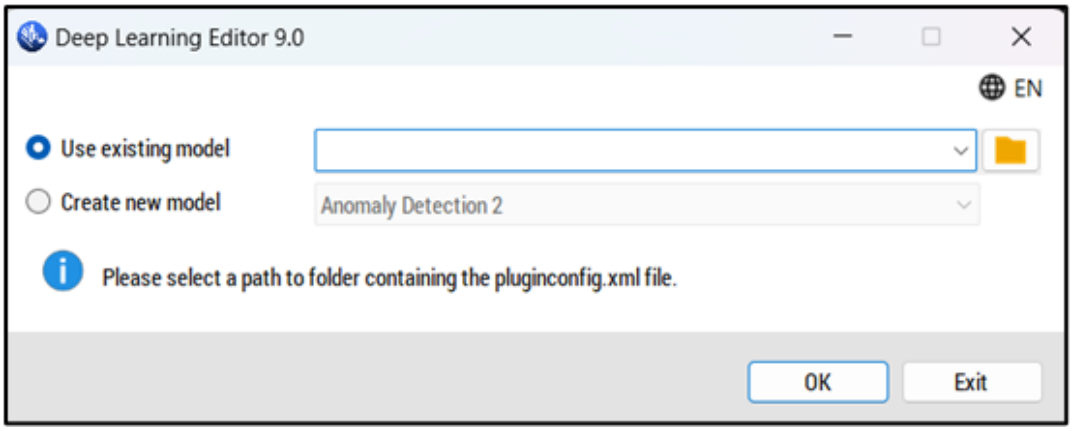
2. Click **Create Model** to use an existing model or create a new one.



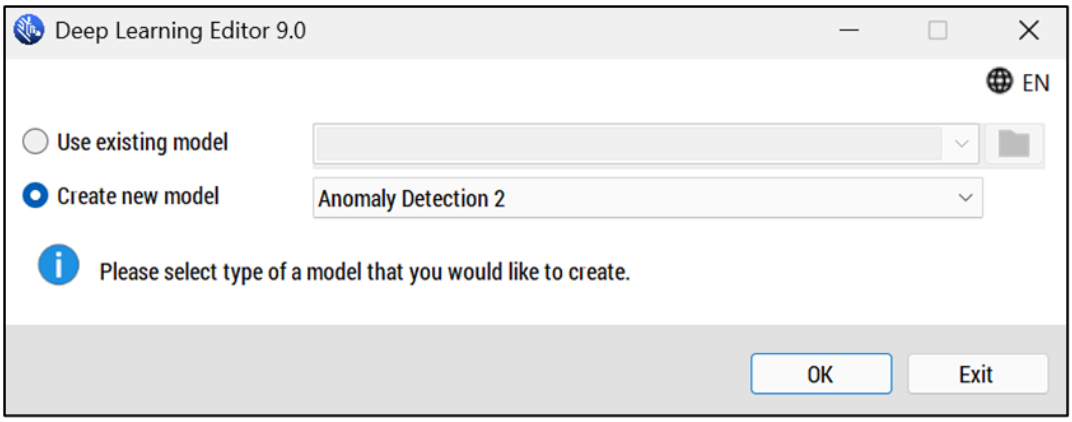
NOTE: Anomaly Detection requires the latest version of Aurora Deep Learning Editor. To download the latest version, open Aurora Focus while connected to an FS42 or NS42 device or emulator and add the Anomaly Detection tool. The dialog box provides a direct link to download the latest version of Deep Learning Editor.



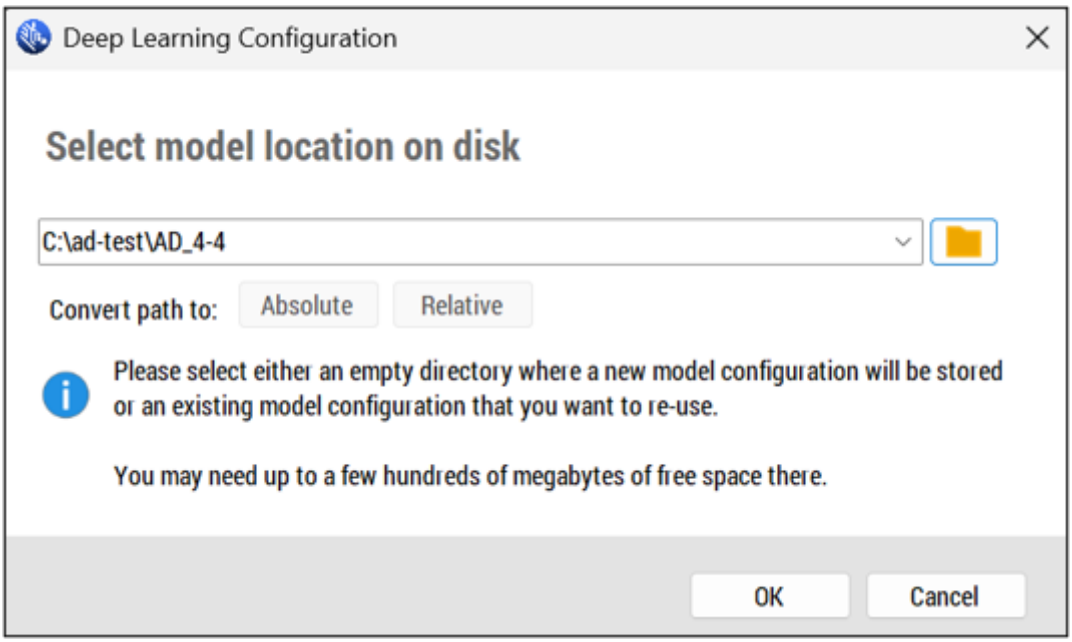
3. If you are using an existing model, click and navigate to the location of the model on your PC, and click the folder icon.



4. If you are creating a new model, select the model type from the dropdown and click **OK**.



5. Click to select a location for the model on your PC.



NOTE: Model editing is done in Deep Learning Editor.

Understanding Thresholds

Threshold values inform the classification of the images in the dataset by providing scores and confidence levels.

After the training phase, scores are calculated for every training sample and presented as a histogram; good samples are marked with green bars, and bad samples with red bars. All images with scores between 0-T1 are marked good, and images above T2 are classified as bad. If the score is between T1 and T2, the result is [Confident: No].

Training with many samples from both groups is recommended to achieve a more robust threshold.

- Images that have scores within 0-T1 are marked as Good, [Confident: Yes]
- Images that have scores within T1-T are marked as Good, [Confident: No]
- Images that have scores within T-T2 are marked as Bad, [Confident: No]
- Images that have scores within T-T2 are marked as Bad, [Confident: Yes]
- If both thresholds are equal (T1=T2=T), there is no area of uncertainty. Results are marked as Confident: Yes.

NOTE: The image viewer in Aurora Focus provides information on Tool Time, Confident, and Score. [Confident: No] indicates the score is close to the threshold. In this case, perform another inspection.

Figure 7 Job Fail, Confident: Yes

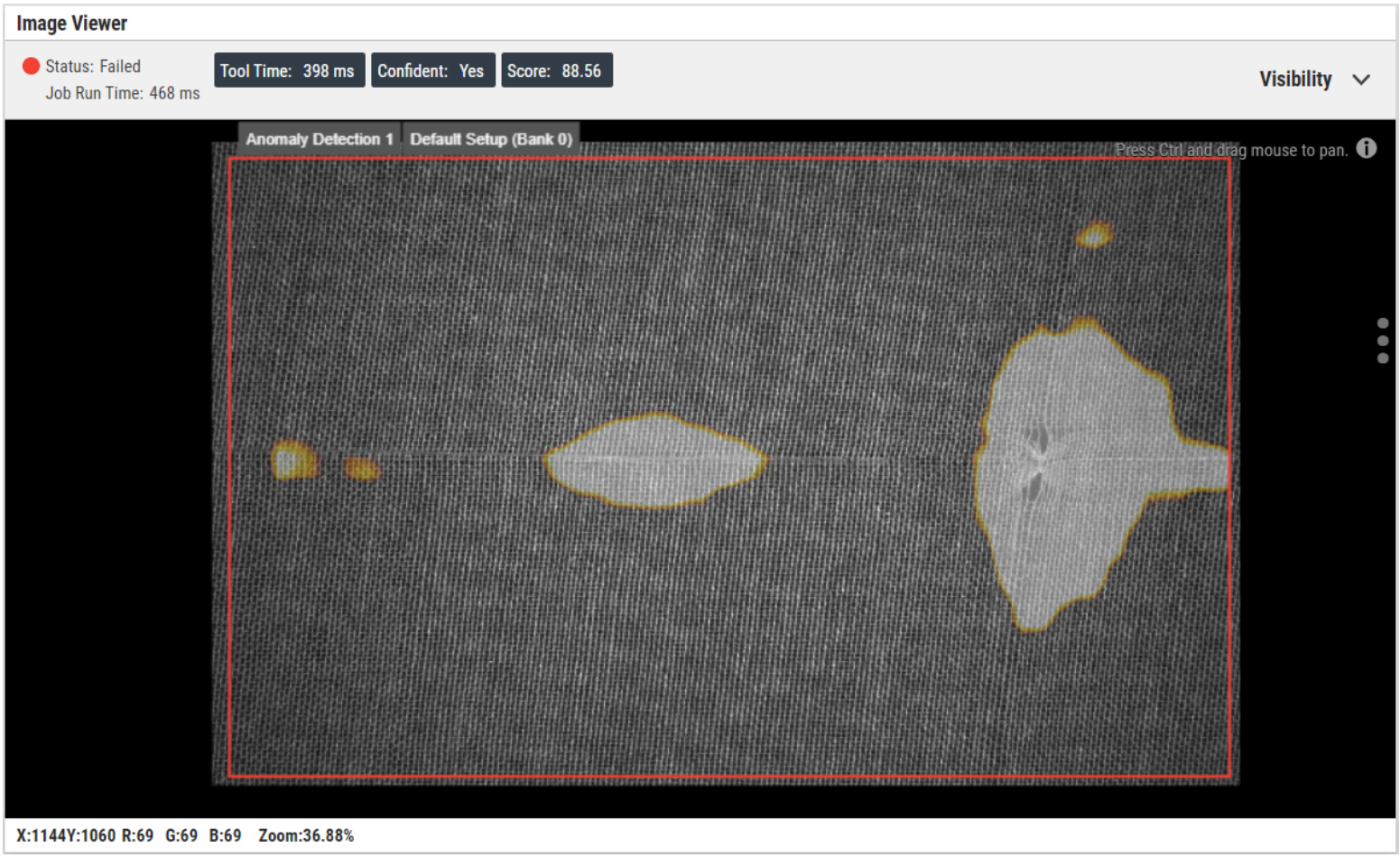


Figure 8 Job Pass, Confident: No

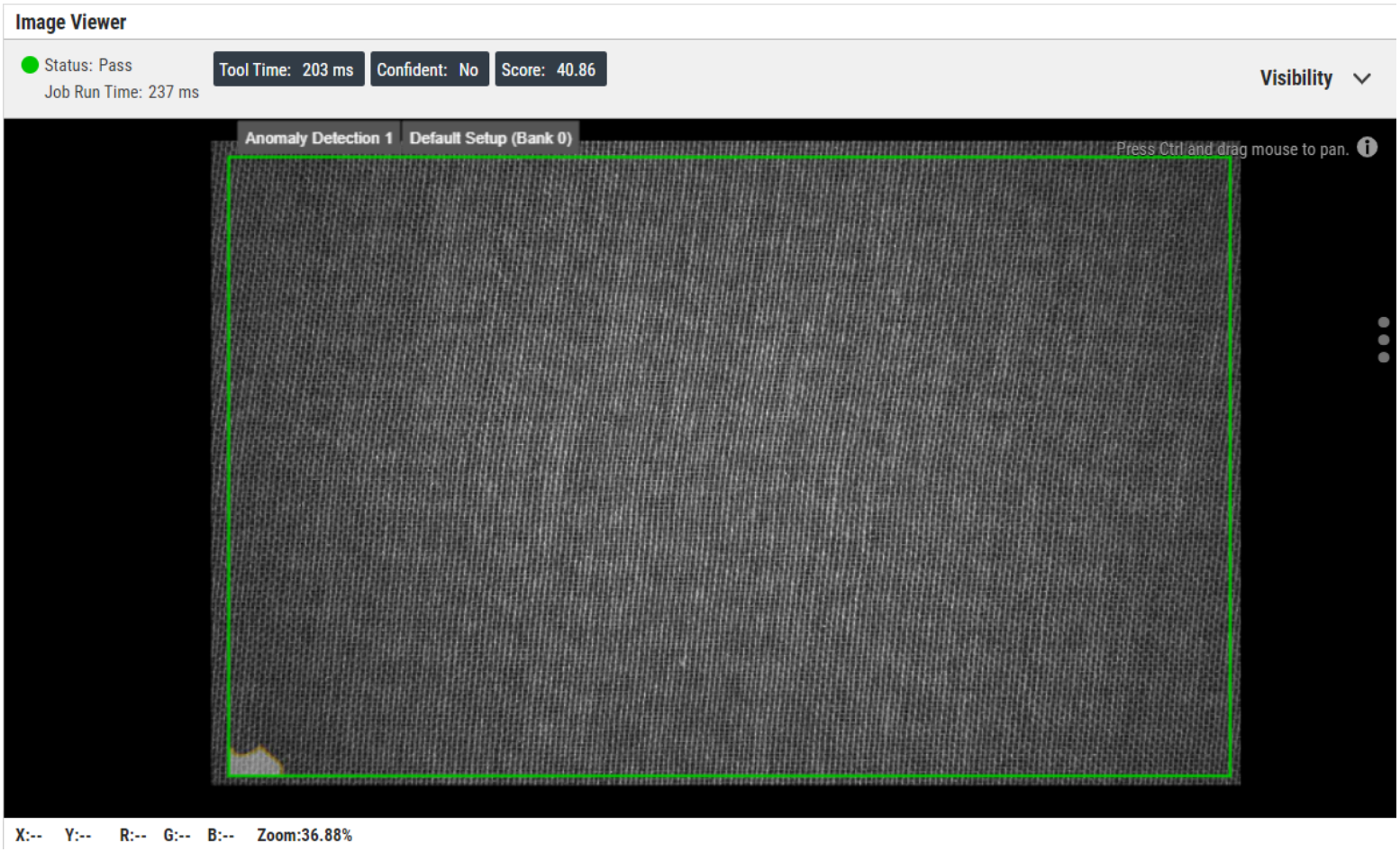
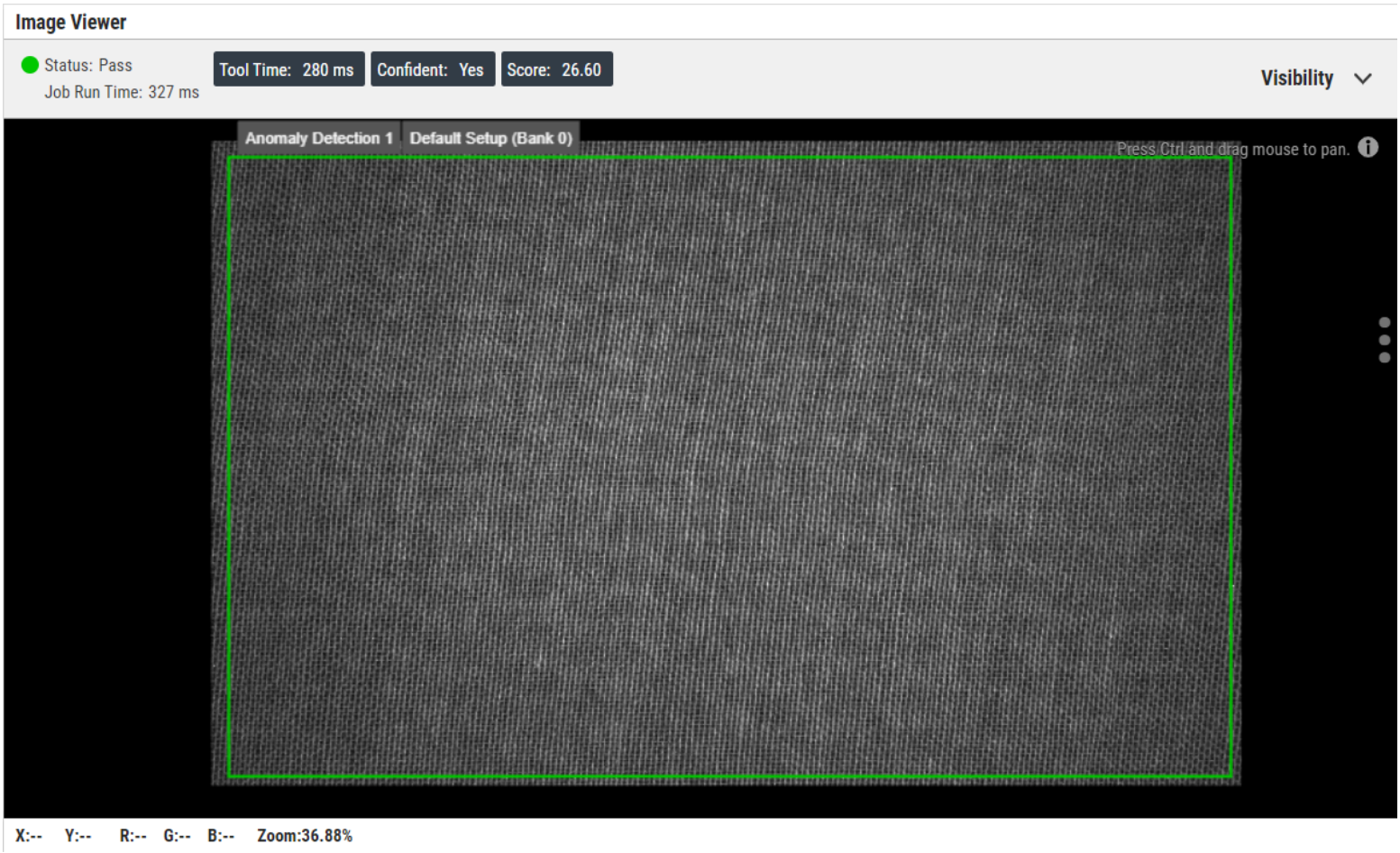


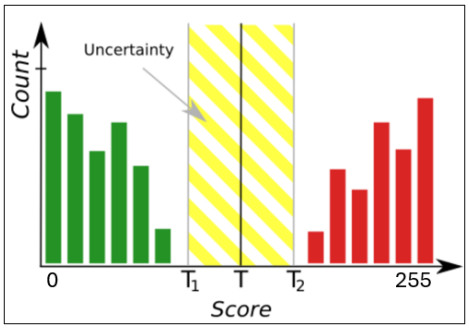
Figure 9 Job Pass, Confident: Yes



The histogram tool displays green bars representing correct samples and red bars representing anomalous samples. T (the midpoint of T1 and T2) marks the main threshold, and T1 and T2 define the area of uncertainty [Confident: No].

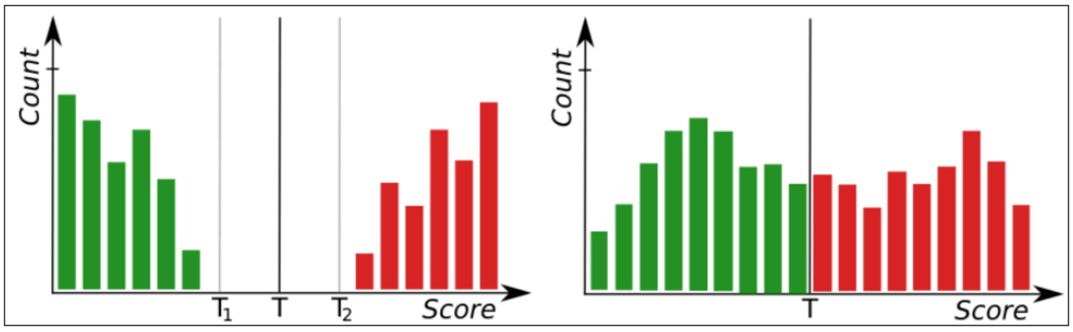
The following histogram displays well-separated groups, indicating that the model has good accuracy:

Figure 10 Uncertainty Threshold



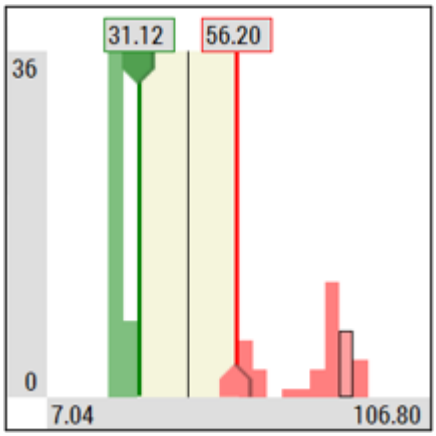
The following histogram displays groups in close proximity, indicating that the model has poor accuracy:

Figure 11 Good and Bad Model Accuracy Examples



The following histogram provides a real-world example in Aurora Deep Learning Editor:

Figure 12 Aurora Deep Learning Editor Example



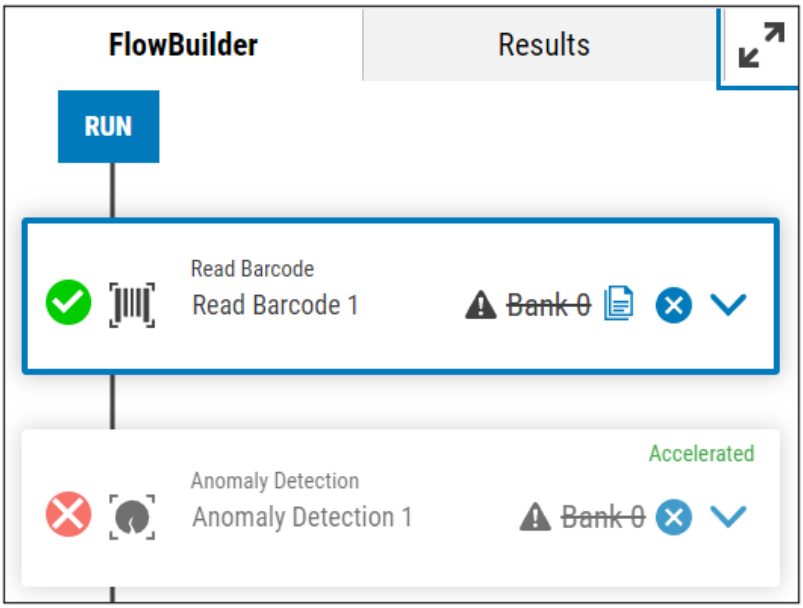
This example shows 36 images in the model with a clear separation between Good and Bad images. The range of scores for Good images is 7.04-31.12, and the range for Bad images is 56.20-106.80. The yellow separation between Good and Bad images indicates that the model has good accuracy.

Fixturing in Anomaly Detection

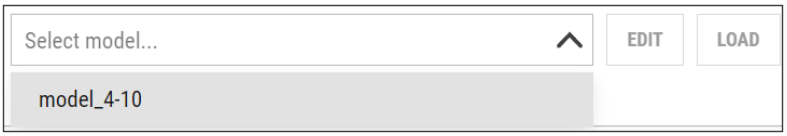
Use fixturing in Anomaly Detection to dynamically control the Region of Interest (ROI) using a fixture from a Locate or Read Barcode tool.

NOTE: Models created in Aurora Focus 9.3 and above can use fixturing.

1. Choose a Locate or **Read Barcode** tool and add it to the **FlowBuilder**.

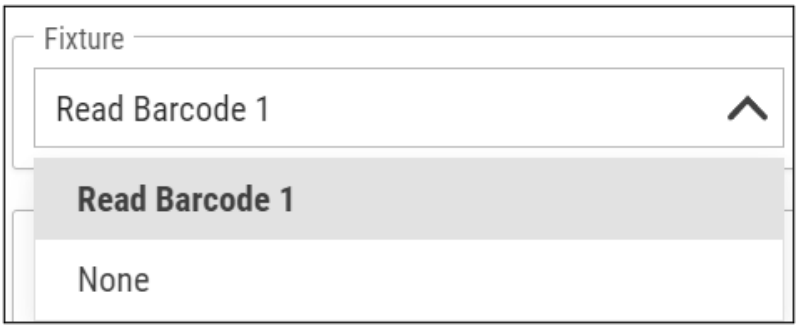


2. Select a model from the **Model** drop-down and click **Load**.



NOTE: Fixturing is not available until after the model loads.

3. Select a **Fixture** from the drop-down and select a fixture source.



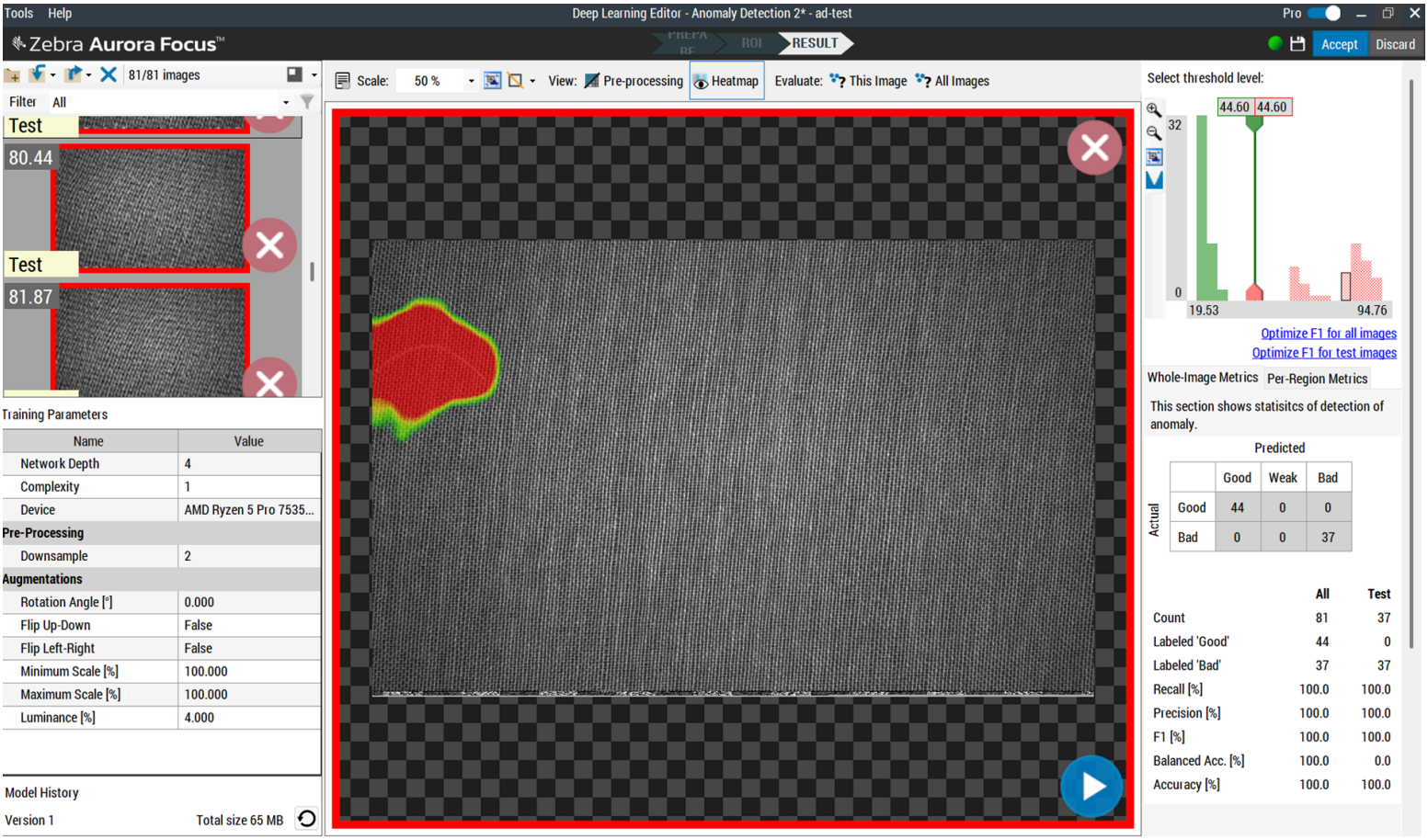
4. Once the fixture is selected, the status indicator changes to Set.

Using Pro Mode

Enable Pro Mode to access advanced parameters such as Network Depth, Complexity, and Device.

1. Toggle the button to the right to enable Pro Mode 

.
2. Observe the following additional parameters:

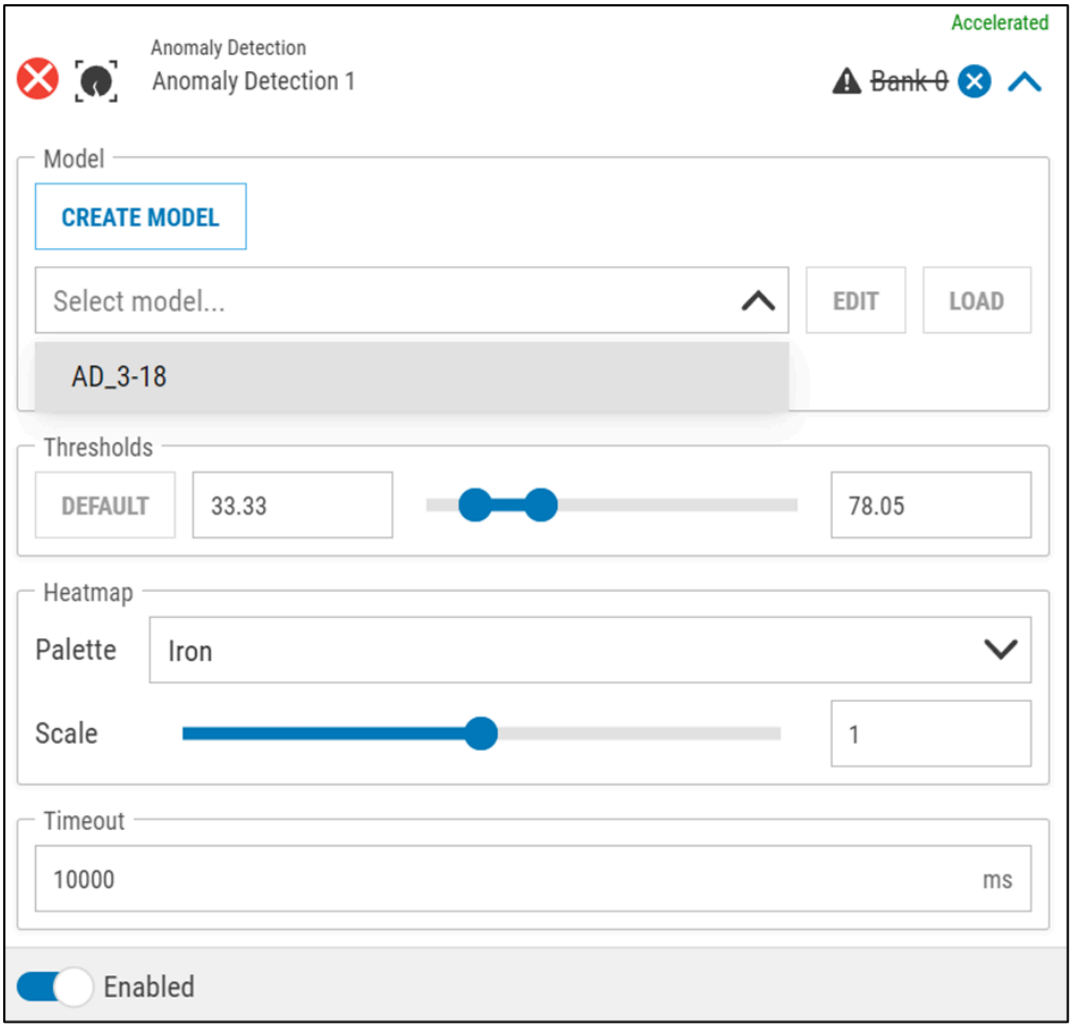


- Network Depth - the size of the network's internal memory. Use higher depth for images containing complex patterns or a wide variety of objects.
- Complexity - higher density provides more precise heatmaps but can increase execution time.
- Device - provides information on the device the training is being performed on.
- Pre-Processing
  - Downsample - each level reduces the width and height of the images by a factor of two.
- Augmentations
  - Rotate Angle - rotate samples by a random angle between 0 and the parameter value.
  - Flip Up-Down - reflect samples along the x-axis.
  - Flip Left-Right - reflect samples along the y-axis.
  - Minimum Scale Percentage - minimum value of random scale factor applied to samples.
  - Maximum Scale Percentage - maximum value of random scale factor applied to samples.
  - Luminance Percentage - change brightness of samples by a random percentage (between -ParameterValue and +ParameterValue) of pixel values (0-255).

Editing and Deploying Models in Aurora Focus

Load an existing model in Aurora Focus to edit and deploy on a set of images.

1. Select the model from the drop-down menu.



   a) Click **Load** to deploy the model on a set of images in Aurora Focus.

   b) Click **Edit** to edit the model in Deep Learning Editor.

2. Configure the model settings as necessary:

   - The Pass and Fail state of the tool is defined by the Threshold T (mid-point of T1 & T2), which is defined in the Deep Learning Editor. For additional information on thresholds, go to Understanding Thresholds .
   - Heatmap - change the visualization settings of the heatmap that appears when reviewing results.

Measurement Tools

Use measurement tools to measure the distance between two objects, the width of an object, or the diameter of a circle in the region of interest.

Distance

The Distance tool determines the space between two specifically defined objects or patterns in an image.

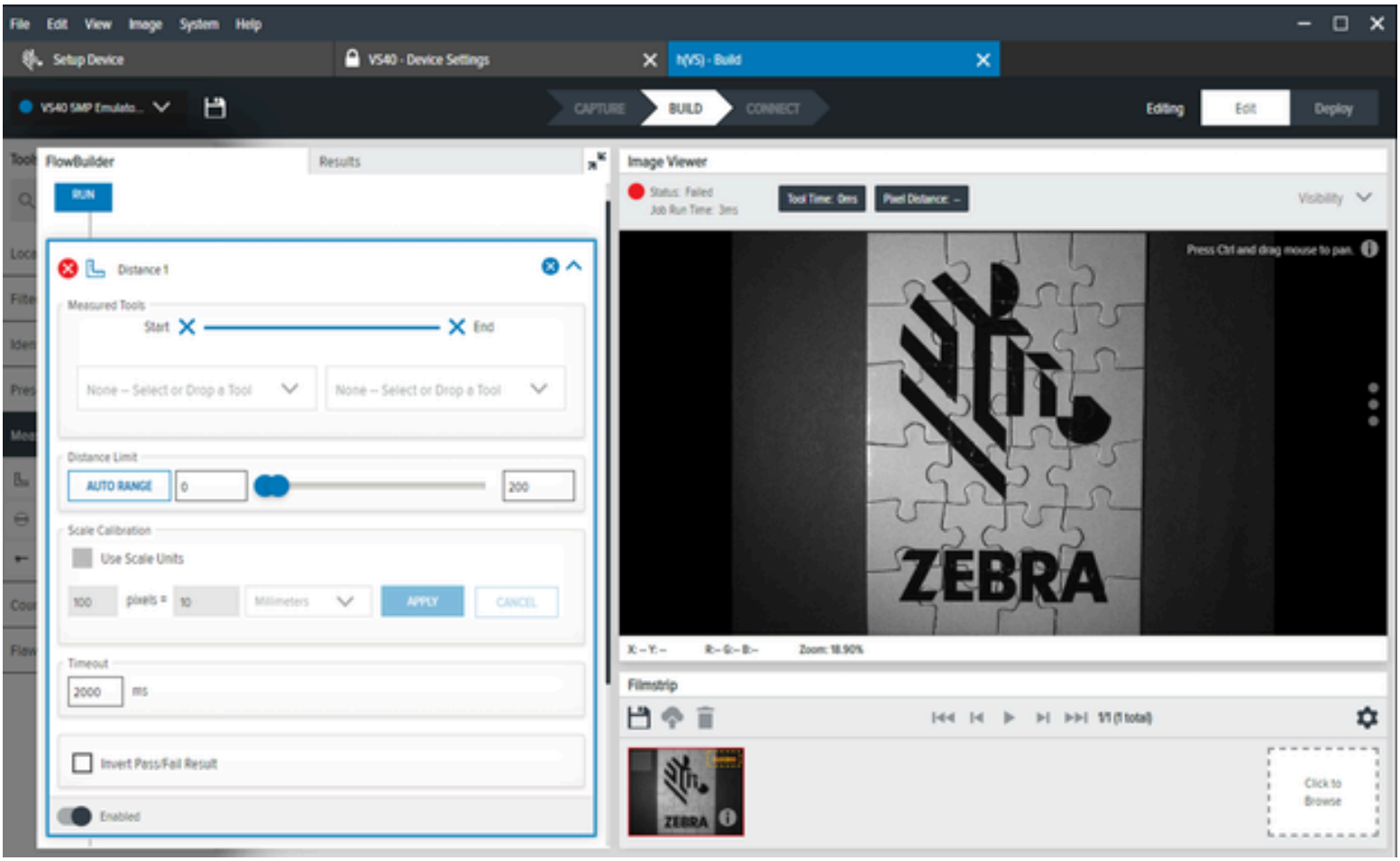


Table 64 Distance Tool Settings

Settings	Description
Measured Tools	There are four different options for where to measure edges: • Left - measures from left or top of an edge. • Middle - measures from the middle of an edge • Right - measures from right/bottom of an edge • Perpendicular/Bottom - creates a measuring line perpendicular to this edge, extending until it intersects with another edge. Select an edge on one side.
Distance Limit	Determine the minimum and maximum values.
Scale Calibration	Calibrate pixel values to engineering units measured in centimeters, millimeters, microns, or inches.

Circle Diameter

The Circle Diameter tool measures the diameter of a circle located in the Region of Interest (ROI).

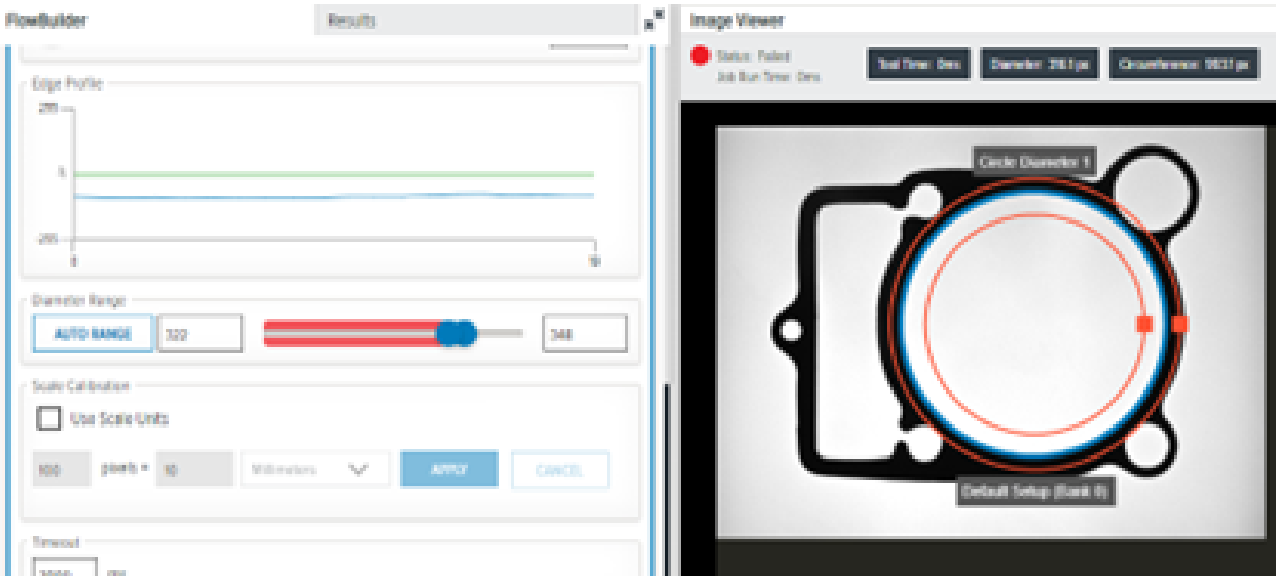


Table 65 Circle Diameter Settings

Settings	Description
Find By	Define the approach used to search for the circle: • Best Score • Largest Circle • Smallest Circle
Transition Type	Select the type of edge transition used to find the circle: • Both • Blended • Dark to Light • Light to Dark
Minimum Edge Magnitude	Sets the minimum contrast value used to define the edge of the circle.
Edge Profile	Displays the contrast score of the features across the Region of Interest (ROI).
Diameter Range	The Diameter range parameter defines the diameter value for which the tool returns a pass or fail status.
Scale Calibration	Calibrates pixel values to engineering units.

Measure Object Width

The Measure Object Width tool measures the width of an object present in an image.

The tool performs a series of scans along Scan Count parallel scan segments constructed from Measure Object Width Region of Interest (ROI). The obtained points are then used for computing two parallel segments using a slightly modified segment-fitting routine. The Outlier Suppression parameter supports the process. Finally, the stripe widths and fitted segments' direction compute the object width.

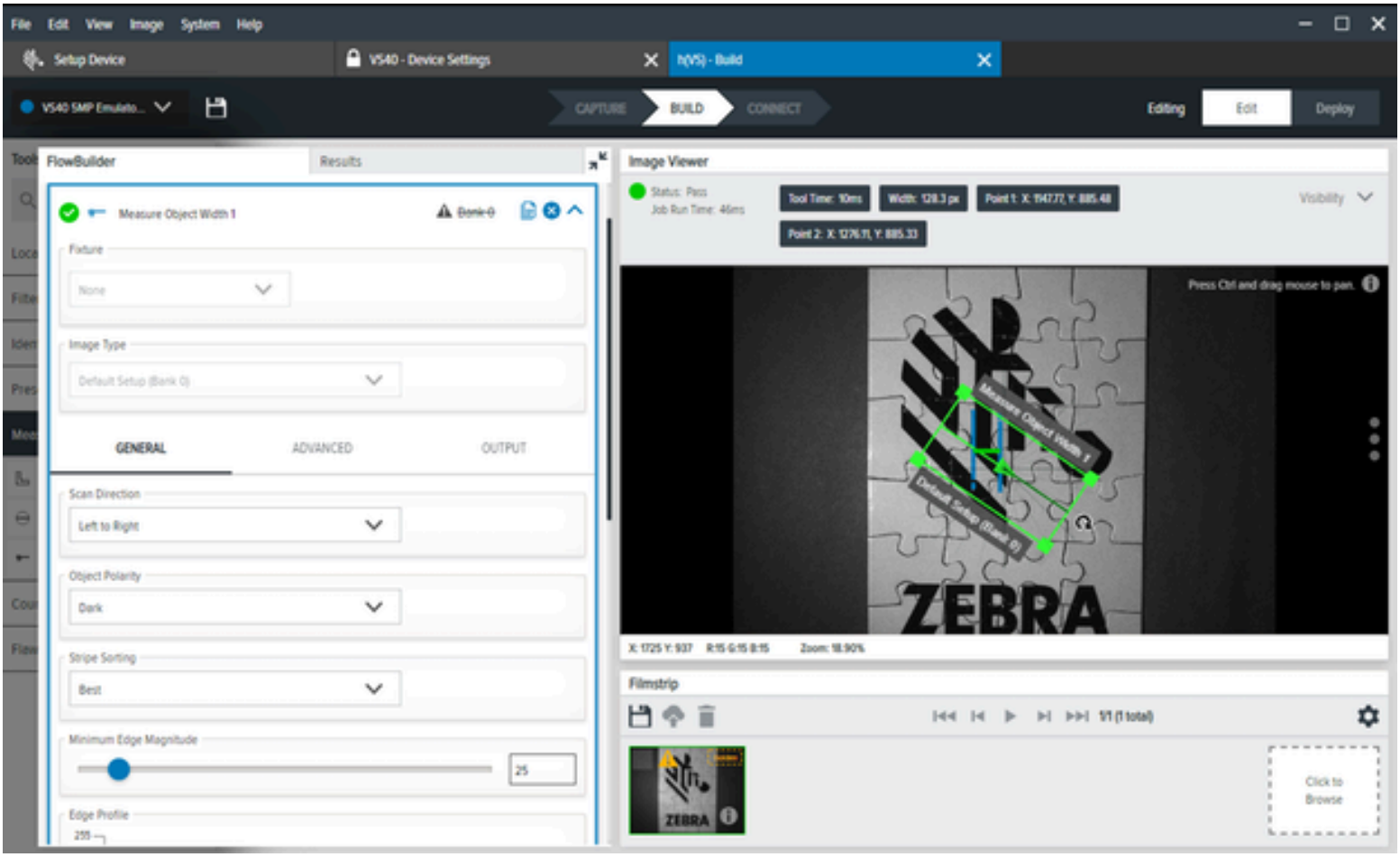


Table 66 Measure Object Width Settings

Setting	Description
Scan Direction	Set the direction that the tool uses when searching for edges.
Object Polarity	Determine the contrast type for the target object.
Stripe Sorting	Determine the mode of selection for the edges of the object.
Minimum Edge Magnitude	Set the minimum contrast value used to define object edges.
Edge Profile	Displays the contrast score of the features across the Region of Interest (ROI).

Table 66 Measure Object Width Settings (Continued)

Setting	Description
Distance Limit	Configure the minimum and maximum distance values.
Scale Calibration	Calibrate scale values to engineering units.

Advanced Measure Object Width Settings

Configure Advanced Measure Object Width settings such as scan width, scan count, and object outlier suppression.

Table 67 Advanced Measure Object Width Settings

Setting	Description
Scan Width	Defines the minimum number of pixels to consider when searching for an edge.
Scan Count	Sets the number of scanning segments used across the Region of Interest (ROI) to find object edges.
Object Outlier Suppression	When enabled, this setting reduces the impact of outlier points found on edges by suppressing the influence of values significantly different from most others on the result using Tukey's M-estimator.

Counting Tools

Counting tools determine the instances of particular objects within the Region of Interest (ROI).

Pixel Count

The Pixel Count tool provides the number of pixels of a user-specified value (0-255) within a Region of Interest (ROI).

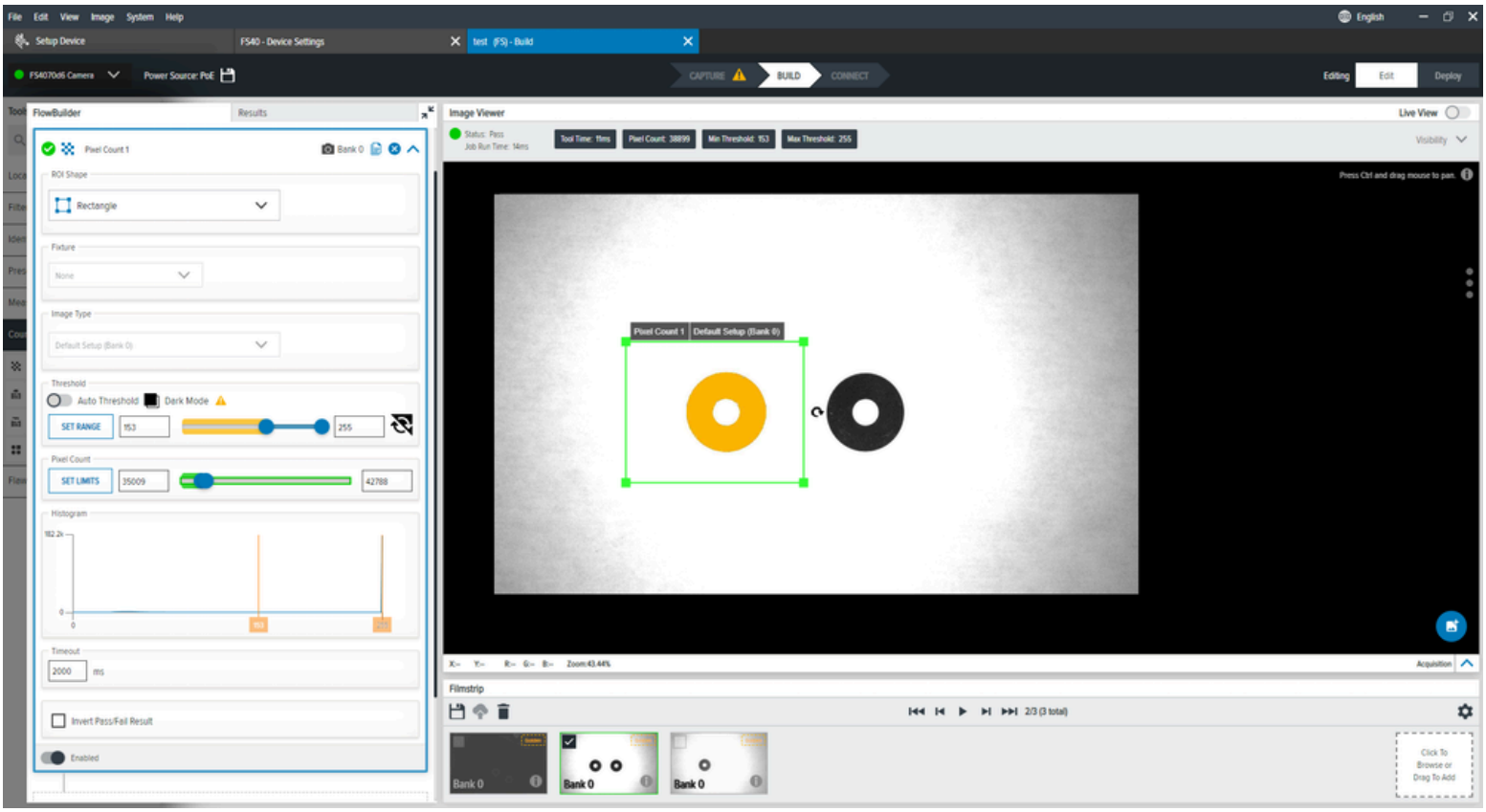


Table 68 General Pixel Count Settings

Setting	Description
Threshold	The minimum and maximum threshold values for Pixel count.
Pixel Count	The minimum and maximum pixel count values are considered for passing.
Histogram	Shows the number of pixels at each value.

Blob Count

The Blob Count tool counts the number of blobs that pass certain filter parameters within a Region of Interest (ROI).

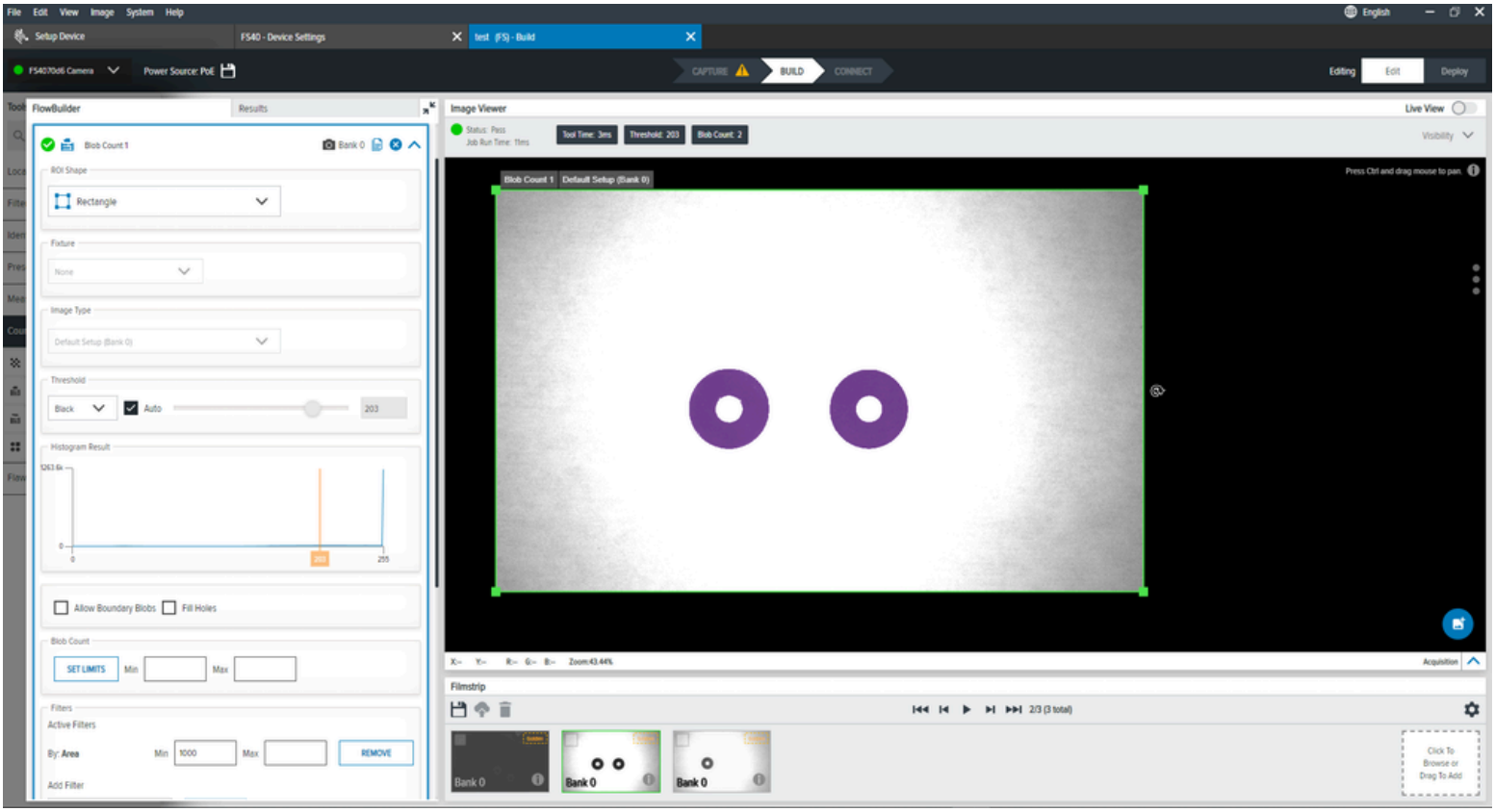


Table 69 Blob Count Settings

Settings	Description
Threshold	Determine the black or white threshold from the drop-down and use the slider to change the threshold value.
Histogram Result	Visual representation of the number of pixels found at each greyscale level.
Blob Count	The minimum and maximum blob count values are considered for passing.
Filters	Apply filters to set the criteria to consider a blob valid for this tool.

Edge Count

The Edge Count tool identifies transitions based on the highest contrast level in a Region of Interest (ROI).

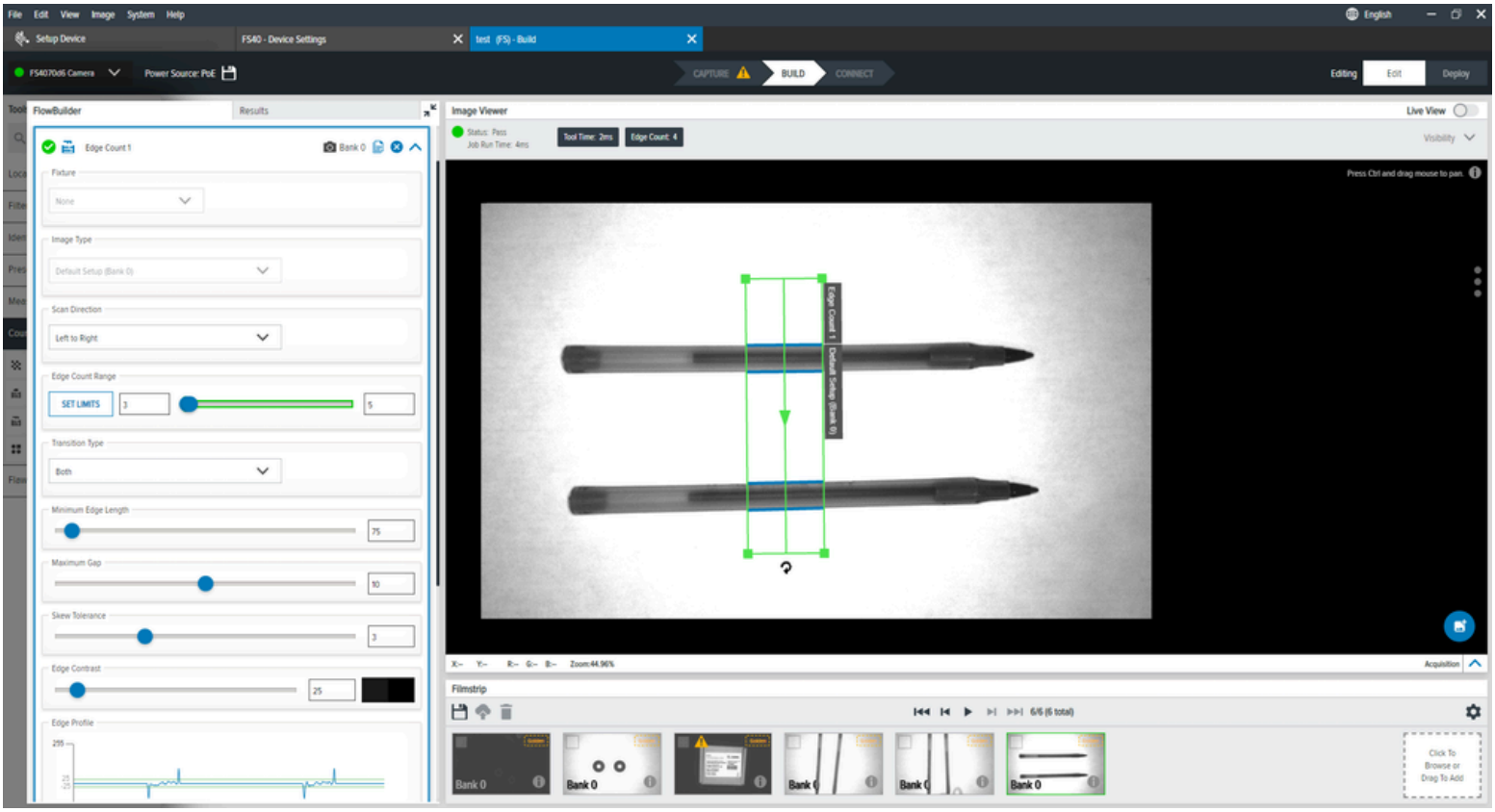


Table 70 Edge Count Settings

Settings	Description
Scan Direction	Set the direction that the tool uses when searching for edges. • Left to Right • Right to Left • Top to Bottom • Bottom to Top
Edge Count Range	
Transition Type	Select the type of edge transition used to find the circle starting from the center outwards.
Minimum Edge Length	The minimum length in pixels of an edge to use.
Maximum Gap	The maximum size in pixels to consider in gaps of the edge.

Table 70 Edge Count Settings (Continued)

Settings	Description
Skew Tolerance	The number of degrees to attempt to match the skew of the line if it is not straight.
Edge Contrast	Threshold acceptance to consider a line.
Edge Profile	Displays the contrast score of the features across the region of interest.

Locate Object Count

Use the Locate Object Count tool to find a pattern in the image and set a match threshold for a candidate to be considered a match.

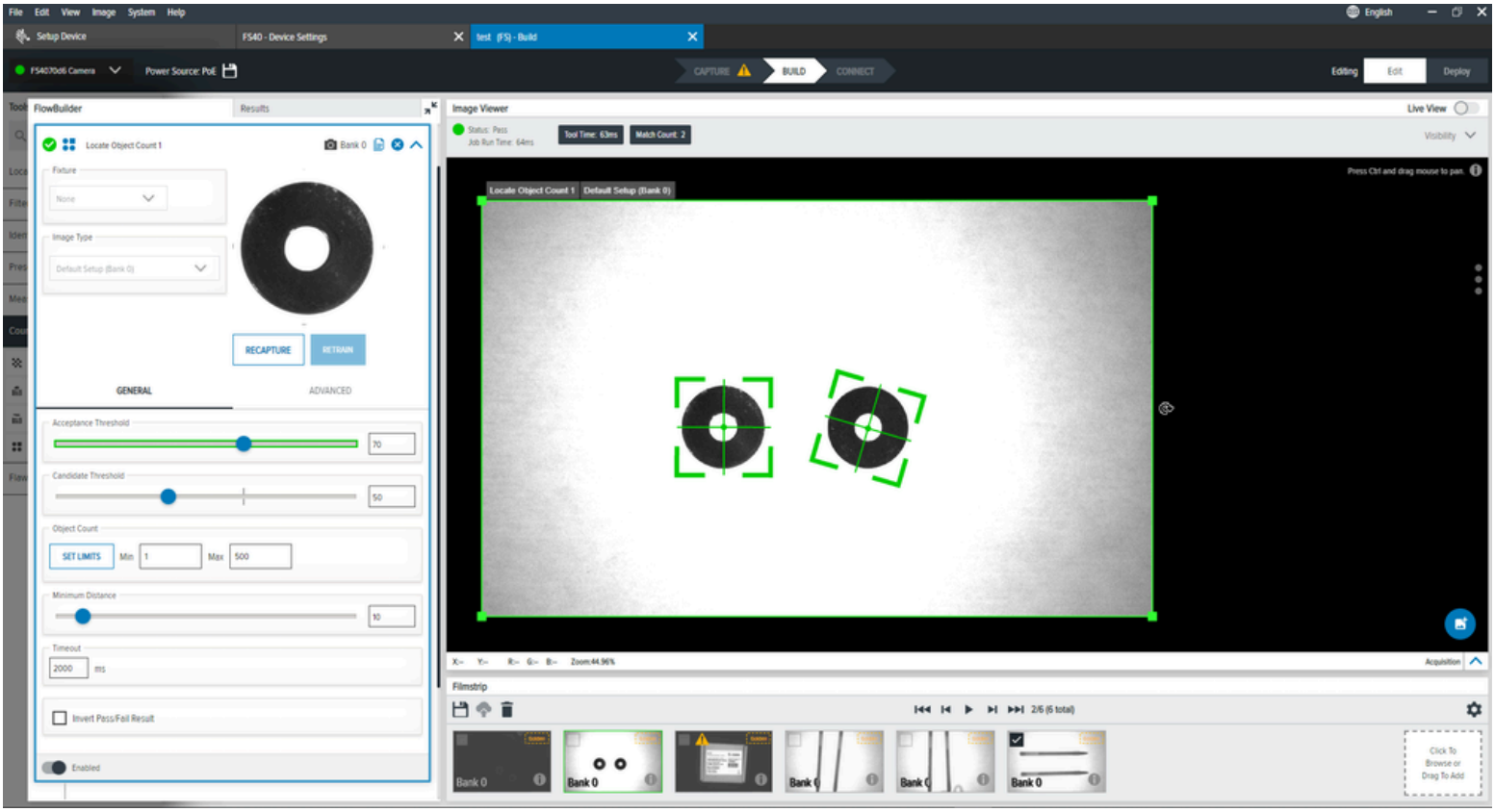


Table 71 General Locate Object Count Settings

Settings	Description
Acceptance Threshold	The minimum match score required to be considered a passing match. If multiple object match scores are above this minimum, the highest matching object score is the final match.

Table 71 General Locate Object Count Settings (Continued)

Settings	Description
Candidate Threshold	The threshold matches the trained pattern to a pattern in the acquired image.
Object Count	The minimum match score required to be a passing match. If multiple object match scores are above this minimum, the highest matching object score becomes the final match.
Minimum Distance	Set the minimum distance required between found objects.

Advanced Locate Object Count Settings

Configure Advanced Locate Object Count settings such as performance control, noise, inverting the pass/fail result. and setting a minimum distance.

Table 72 Advanced Locate Object Count Settings

Settings	Description
Performance Control	Use the selector to choose the best coarseness and search type combinations for a more performant search.
Noise	Removes pixel-level noise but preserves edge data.
Invert Pass/Fail Result	Inverts the result from the tool.
Minimum Distance	Sets the Minimum distance required between identified objects.

Flaw Detection

Flaw Detection tools perform a pixel-to-pixel comparison of two images. This is helpful for cases where the object's surface or shape is complex.

- Edges - comparison based on objects' edges. This method is helpful when defects occur on the object's edge, pixel comparisons fail due to different light reflections, or when checking the object's surface is unnecessary.
- Intensity - comparison based on pixel intensity. Using this method, two images are compared pixel-by-pixel, and the defect is classified based on pixel intensity differences. This technique helps find defects like smudges or scratches.

The model uses a previously prepared image to compare images from the device. This technique provides a quick comparison inspection when specific conditions are met, such as stable lighting conditions, camera position, and precise object positioning.

Edges

The Edge tool compares edges given input image with an image stored in a model created in the Training tab. As a result, the tool creates a region containing only pixels with different edges.

This method helps find defects in the object's shape. Due to its resilience against pixel gray level changes, it is helpful in applications with changing light conditions.

Parameter Maximum Distance defines the maximum distances of two edges that should be treated as the same edge.

The **Edge Threshold** and **Edge Hysteresis** parameters in the **Inspection** tab should have the same value as in the **Training** tab.

NOTE: Increasing the Edge Hysteresis connects more pixels by detecting edges.

If the input image's brightness significantly differs from the brightness of the golden image used in the **Training** tab.

Decrease the **Edge Threshold** slightly to concentrate on missing edges or increase the value to focus on excessive edges.

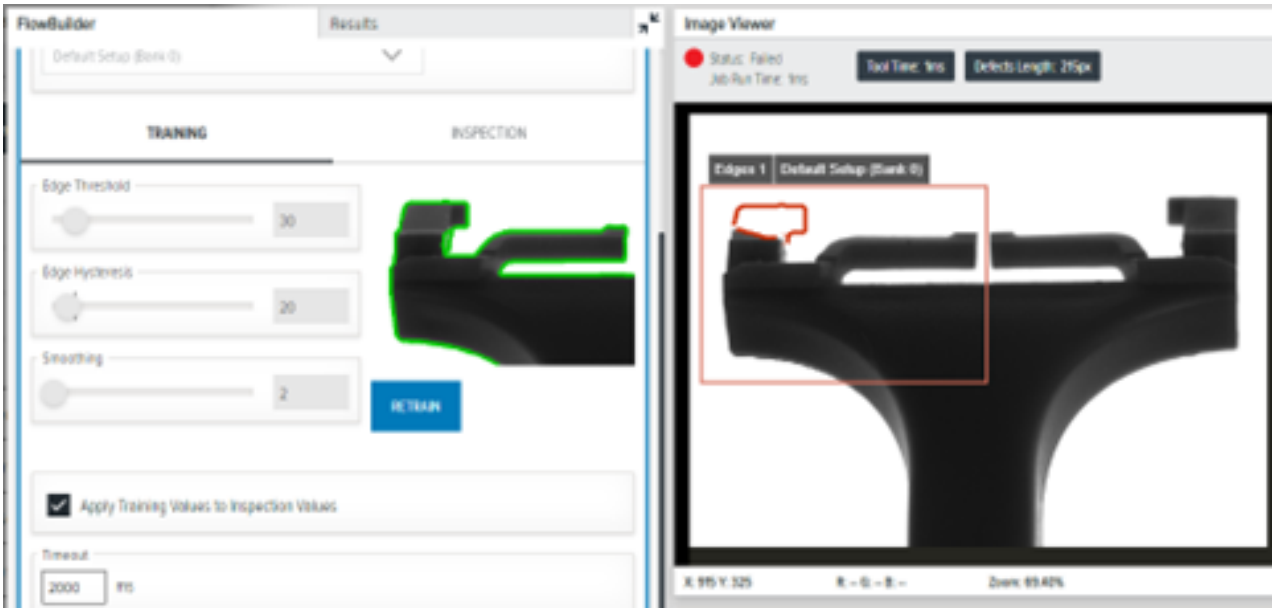


Table 73 Edges Settings

Setting	Description
Edge Threshold	Determine a sufficient edge strength. Edges of at least 30 are detected on the input image.

Table 73 Edges Settings (Continued)

Setting	Description
Edge Hysteresis	Determine the value by which the edge threshold decreases for edge points neighboring sufficiently strong edges. **NOTE:** Use this parameter to control how weak adjacent pixels can be connected to a strong pixel. A higher value connects more pixels
Smoothing	Determine the tool's amount of horizontal and vertical smoothing to improve edge detection.

Intensity

The intensity tool detects edges given input image with an image stored in a model created in the Training tab and then, as a result, creates a region containing only pixels where edges are different.

This method is especially useful for finding defects in the object's shape. Due to its invulnerability to color changes, it may be used in appliances with changing light conditions. Parameter Maximum Distance defines the maximal distances of two edges that should be treated as the same edge.

The **Edge Threshold** and **Edge Hysteresis** parameters in the **Inspection** tab should have the same value as in the **Training** tab. The values should be changed, however, if the input image's brightness is significantly different from the brightness of the golden image used in the **Training** tab. The **Edge Threshold** can also be decreased slightly if one wants to concentrate on missing edges or increased to concentrate on excessive edges.

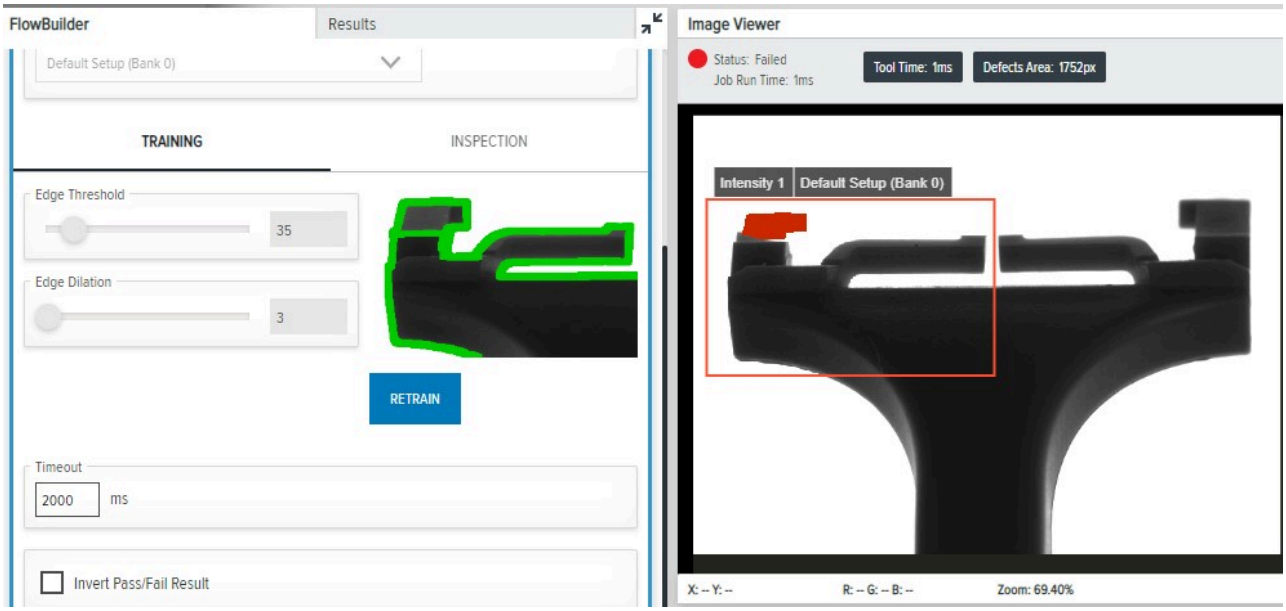


Table 74 Intensity Settings

Setting	Description
Edge Threshold	Determine the minimum strength of the edges on the model where comparison is not performed.
Edge Dilation	Define how far from the detected edges comparison should not be performed.

Device Licensing

This guide provides detailed information on licensing options for Zebra FS and VS devices, including how to obtain a product key ID, activation methods, and features enabled by each license. It also covers the fixed industrial scanning and machine vision toolsets applicable to each device and their associated licenses.

NOTE: Ensure that your date/time is current before applying for a license. If the Date/Time is out of sync, unexpected behavior may occur, such as Device Upgrade Licenses appearing to be successfully applied, yet the tool that should be unlocked does not display in ToolBox when creating a job (for example, the OCR tool). If your clock is out of sync, update it using the Zebra Web HMI and reboot the device afterward. Verify that the time is updated by viewing the top-right bar in the Web HMI (green box) and restart Aurora Focus.

License Types and Toolsets

The following table outlines available licenses and their corresponding part number. Each license is associated with a toolset.

Table 75 License Types

Part Number	Web HMI Name	Description	Toolset	Applicable Devices
LIC-SEN001-0100	Sensor Toolset (Machine Vision)	Upgrade to Sensor Toolset.	MV Sensor	FS10, FS20, FS40, FS42, FS70, FS80
LIC-SEST01-0100	Standard Toolset (Machine Vision)	Fixed Industrial Scanner toolset upgrade from Sensor to MV toolset.	MV Standard	VS20, VS40, VS70
LIC-STD001-0100	Standard Toolset (Machine Vision)	Fixed Industrial Scanner upgrade to the Standard MV toolset (Includes DPM and Fast 2D).	MV Standard	FS10. FS20, FS40, FS70, FS42, FS80

Table 75 License Types (Continued)

Part Number	Web HMI Name	Description	Toolset	Applicable Devices
LIC-DPM001-0200	Fast 1D/2D Decoding with DPM Full	Full DPM (1D/2D/DPM), 60 frames per second. Enables Datacode decoding libraries for DPM reading and enhanced 2D.	Datacode	FS10, FS20, FS40. FS70, FS42, FS80 VS20, VS40, VS70
LIC-2DF001-0200	Fast 1D/2D Decoding	Fast 1D/2D, 60 frames per second (DPM not included).	No toolset associated.	FS10, FS20, FS40, FS70, FS42, FS80 VS20, VS40, VS70
LIC-OCR002-0100	Deep Learning Based OCR	Deep Learning OCR for VS devices.	Deep Learning OCR and MV Essential	VS20, VS40, VS70, NS42
LIC-OCR003-0100	Deep Learning Based OCR	Deep Learning OCR for FS devices.	Deep Learning OCR and MV Essential	FS10, FS20, FS40, FS70, FS42, FS80
LIC-AD002-0100	Deep Learning Based Anomaly Detection	Deep Learning based Anomaly Detection upgrade license for NS42 devices	Deep Learning Anomaly Detection and MV Essential	NS42
LIC-AD003-0100	Deep Learning Based Anomaly Detection	Deep Learning based Anomaly Detection upgrade license for FS42 devices.	Deep Learning Anomaly Detection and MV Essential	FS42
LIC-10LF-0000	Connectivity Gateway Lite	Gateway Connectivity Lite, for FS10 devices.	No toolset associated.	FS10
LIC-20LF-0000	Connectivity Gateway Lite	Gateway Connectivity Lite, for FS20 devices.	No toolset associated.	FS20
LIC-47LF-0000	Connectivity Gateway Lite	Gateway Connectivity Lite, for FS40, FS70, and VS40, VS70 smart cameras.	No toolset associated.	FS40, FS42, FS70 Pre-enabled on FS80 devices, no additional license required.
LIC-EXLF-0000	Connectivity Gateway Full	Gateway Connectivity Full, for FS and VS devices.	No toolset associated.	FS40, FS42, FS70, FS80, VS40, VS70

Toolset List

Table 76 Toolsets

Type	Tool	Sensor	Essential	Standard	DL OCR	DL AD	Datacode	License
Identification	Read Barcode Standard							Available on any FS, VS, or NS device without an additional license.
	Read DPM (using standard decode libraries)						X	LIC-DPM001-0200
	Read DPM and Barcode (using standard decoding libraries)						X	LIC-DPM001-0200
	Datacode (using Datacode decoding libraries)						X	LIC-DPM001-0200
	Deep Learning OCR				X			LIC-OCR002-0100 or LIC-OCR003-0100

Table 76 Toolsets (Continued)

Type	Tool	Sensor	Essential	Standard	DL OCR	DL AD	Datacode	License	
Locate	Locate Object	X	X	X				Available in VS or NS devices without an additional license. or Available on FS devices with LIC-SEN001-0100.	Enabled by any of the following licenses: LIC-OCR002-0100 LIC-OCR003-0100 LIC-AD002-0100 LIC-AD003-0100 LIC-STD001-0100 LIC-SEST01-0100
	Locate Object Plus		X	X					
	Locate Edge	X	X	X				Available in VS or NS devices without an additional license. Available on FS devices with LIC-SEN001-0100.	
	Locate Blob		X	X					
	Locate Circle	X	X	X				Available on VS or NS devices without an additional license. Available on FS devices with LIC-SEN001-0100.	
Filter	Binarize		X	X					
	Dilate		X	X					
	Erode		X	X					
	Open		X	X					
	Close		X	X					
	Gradient Full		X	X					

Table 76 Toolsets (Continued)

Type	Tool	Sensor	Essential	Standard	DL OCR	DL AD	Datacode	License	
	Gradient Horizontal		X	X					
	Gradient Vertical		X	X					
Presence/ Absence	Object P/A	X	X	X				Available on VS or NS devices without an additional license. Available on FS devices with LIC-SEN001-0100.	
	Object Plus P/A		X	X					
	Brightness	X	X	X				Available on VS and NS devices without an additional license. Available on FS devices with LIC-SEN001-0100.	
	Contrast	X	X	X					
	Edge Detect	X	X	X					
	Blob P/A		X	X					
	Anomaly Detection					X			Enabled by any of the following licenses: LIC-AD002-0100 or LIC-AD003-0100

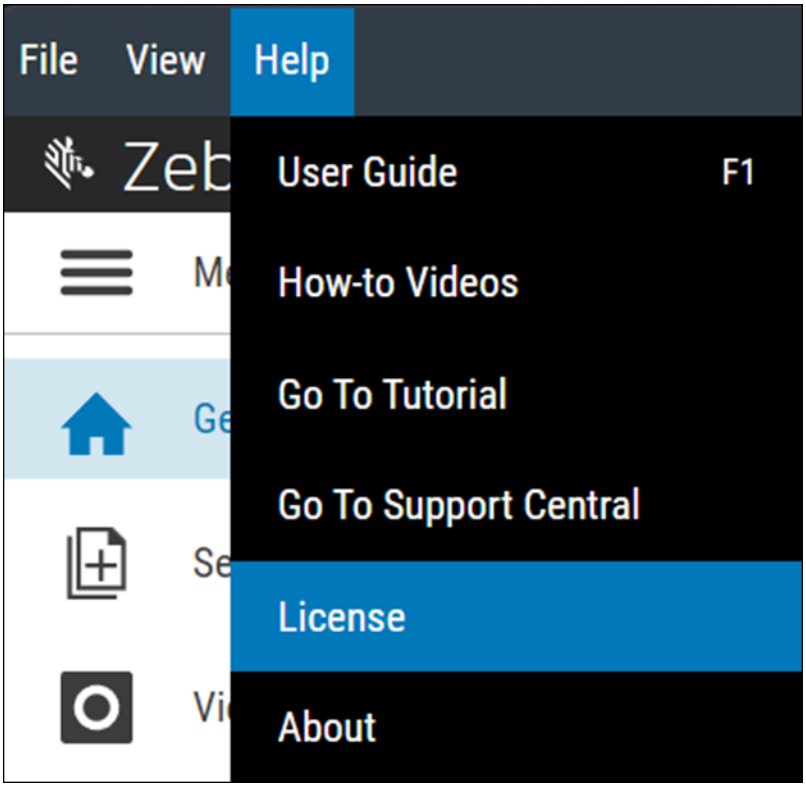
Table 76 Toolsets (Continued)

Type	Tool	Sensor	Essential	Standard	DL OCR	DL AD	Datacode	License	
Measurement	Distance	X	X	X				Available on VS and NS devices without an additional license. Available on FS devices with LIC-SEN001-0100.	Enabled by any of the following licenses: LIC-OCR002-0100 LIC-OCR003-0100 LIC-AD002-0100 LIC-AD003-0100 LIC-STD001-01000
	Circle Diameter	X	X	X					
	Measure Object Width		X	X					
Counting	Pixel Count	X	X	X				Available on VS and NS devices without an additional license. Available on FS devices with LIC-SEN001-0100.	
	Blob Count		X	X					
	Edge Count	X	X	X				Available on VS and NS devices without an additional license. Available on FS devices with LIC-SEN001-0100.	
	Locate Object Count	X	X	X					
Flaw Detection	Edges		X	X					
	Intensity		X	X					

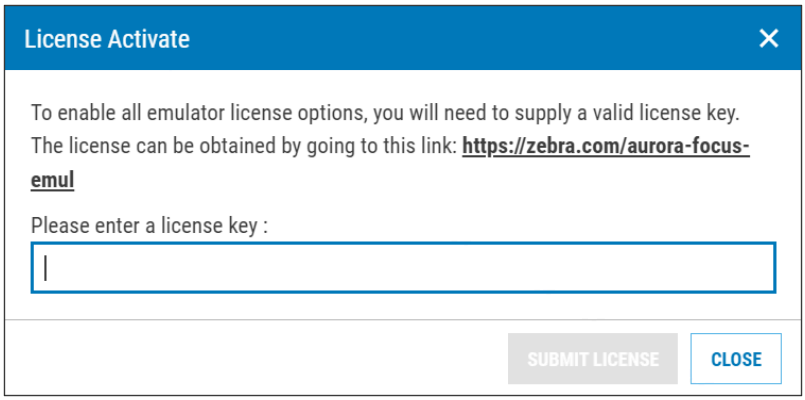
Obtaining an Emulator License Key

Use Aurora Focus to obtain an emulator license key.

1. Click **Help** from the top navigation menu in Aurora Focus.
2. Select **License** from the drop-down menu.

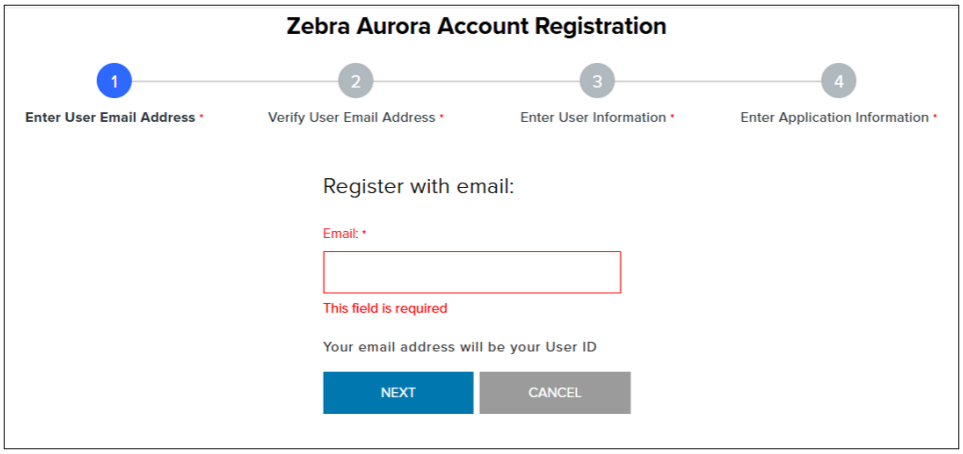


3. Observe the **License Activate** popup and click the link:

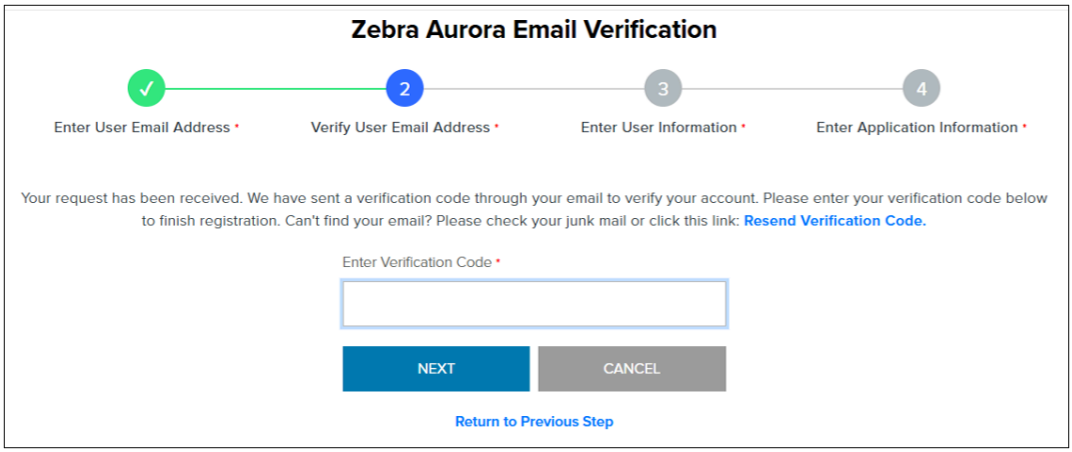


4. Enter your email address.

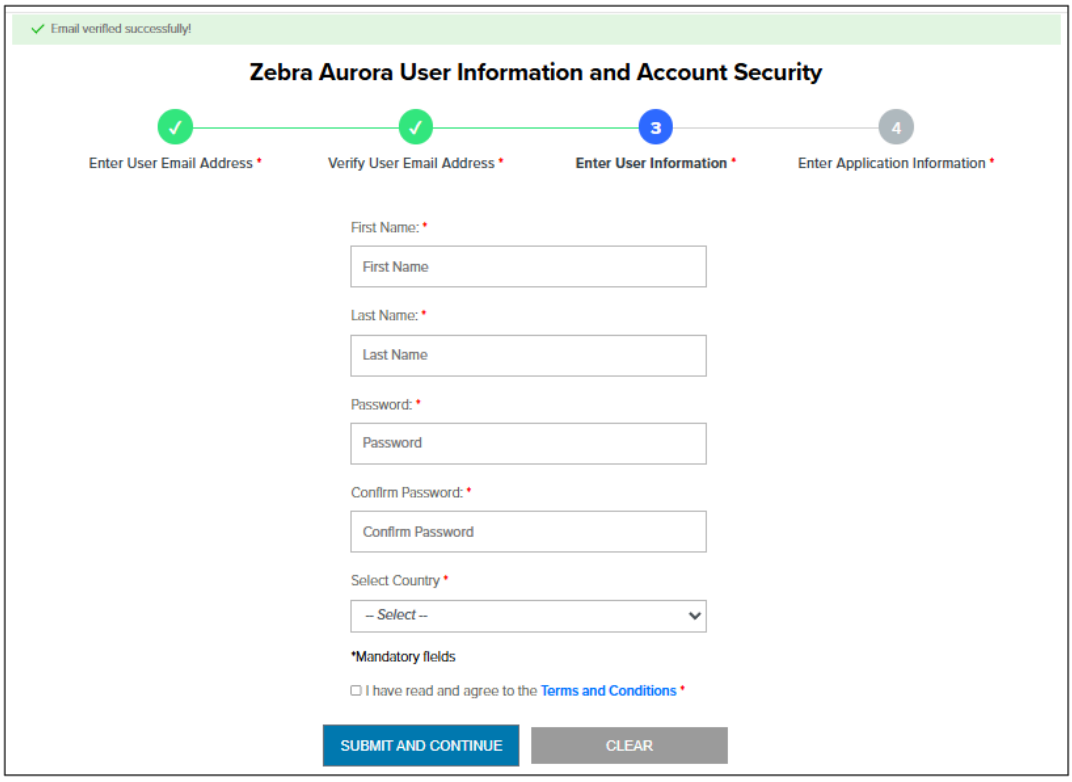
NOTE: This becomes your User ID.



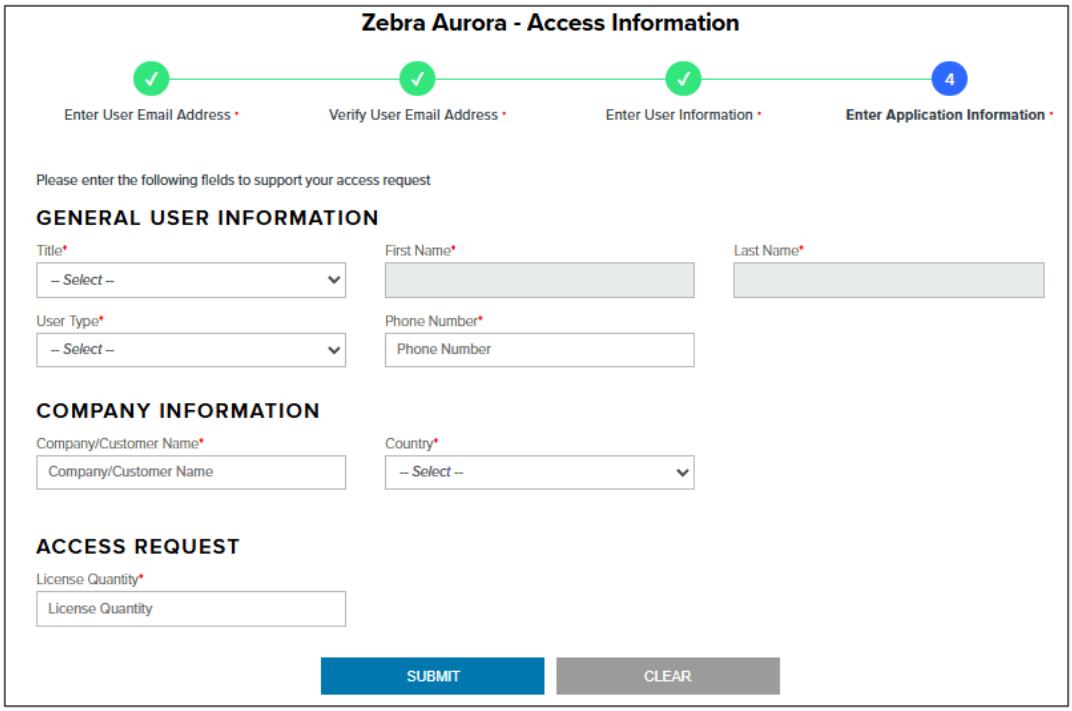
5. Click **Next**.
6. Verify your email address by entering the verification code sent to your inbox.



7. Click **Next**.
8. Enter your name and password, then select your country from the drop-down menu.



9. Click **Submit and Continue**.
10. Enter your user and company information and provide the number of licenses you are requesting.



11. Click **Submit**.

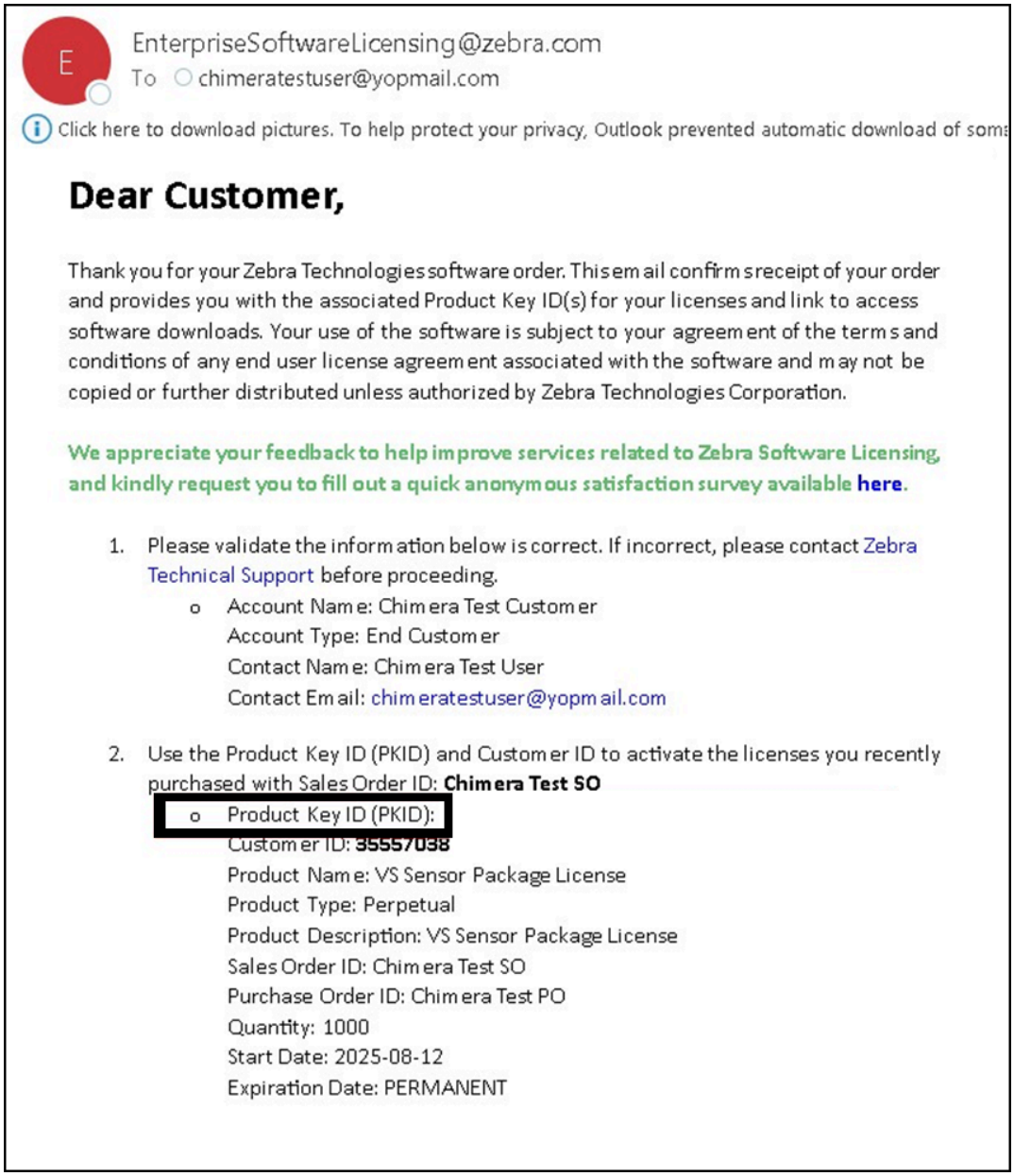
NOTE: Enter the Activation ID into the License Key field after registration to activate the license.

Obtaining the Product Key ID

The Product Key ID (PKID) is the unique 32-bit alphanumeric value provided in the Zebra License Notification email when the license is procured.

Use this key to activate a purchased upgradable feature in Aurora Focus. An example of the PKID is **xssn-ixa3-tdgb-elsi-mxd8-q2kq-cw30-01bp**.

Figure 13 Zebra License Notification Email



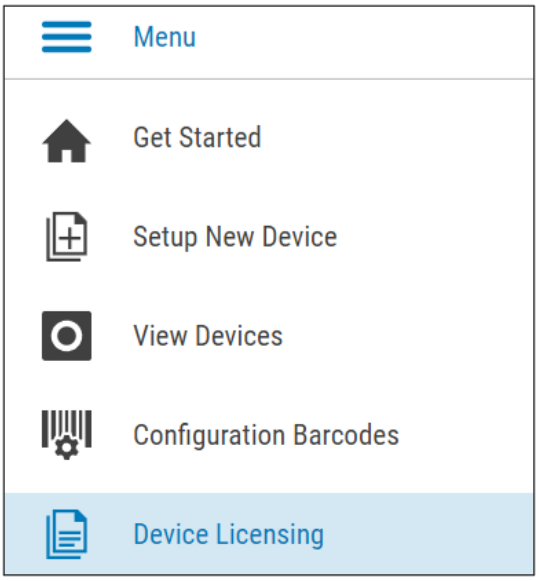
NOTE: Refer to the link shared in the Zebra License Notification email when the license is procured for additional license information.

Activating a License Key in Aurora Focus

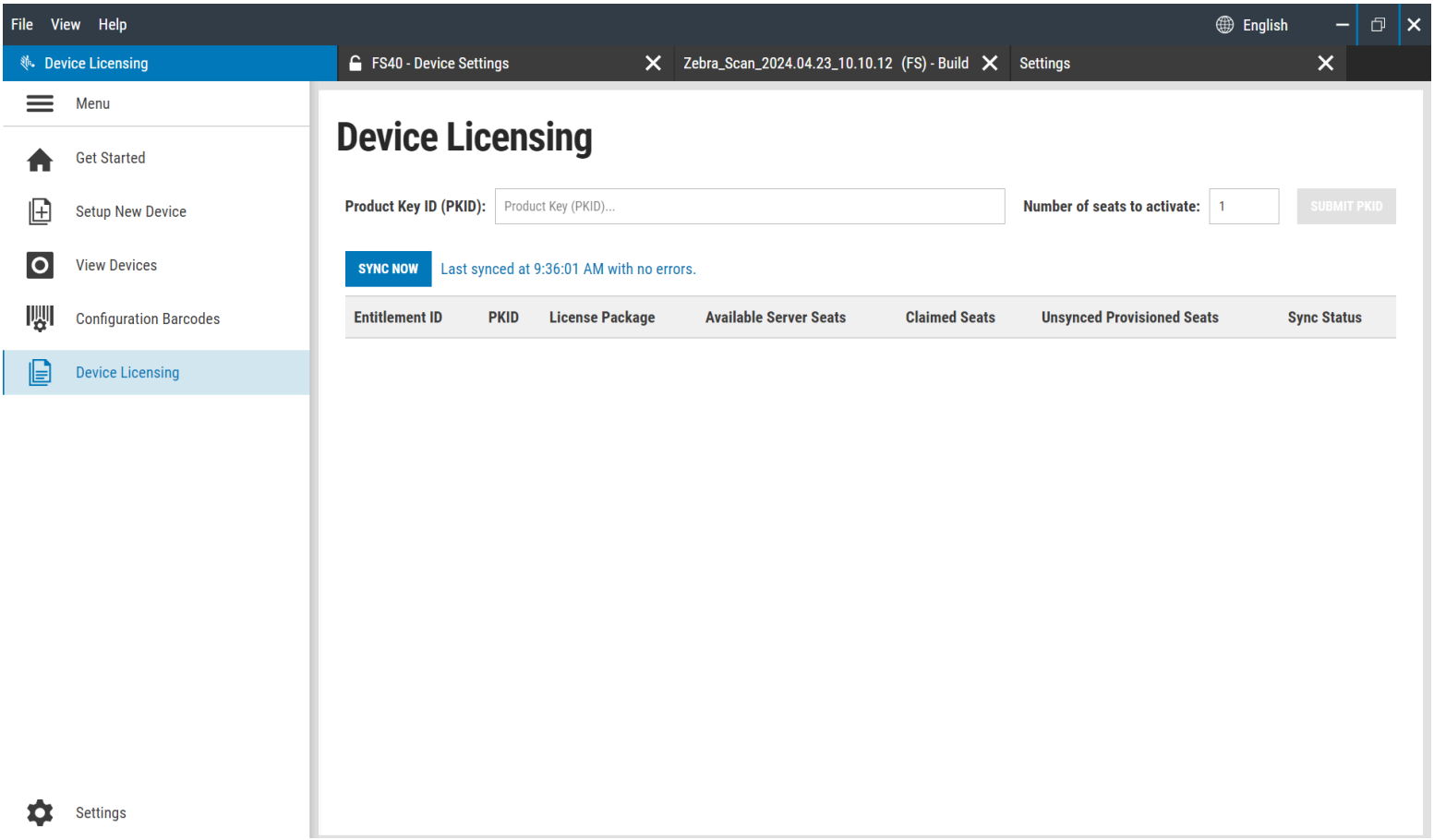
Use the Device Licensing tab in Aurora Focus to manage licenses and remotely provision them to devices.

NOTE: Activating, deactivating, and syncing operations require access to the Aurora Focus Desktop application with a direct internet connection.

1. Select **Device Licensing** from the menu in Aurora Focus.

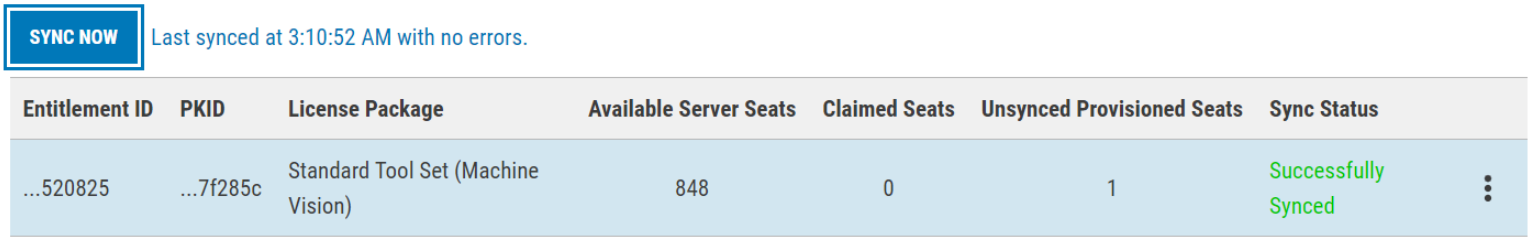


2. Enter the PKID from the Zebra License Notification email and the number of seats to activate.



NOTE: The number of seats to activate must be to greater than the number of available server seats. The maximum is 25 seats per activation.

3. Click **Submit License Key**.
4. Click **Sync Now** to view a list of active entitlement IDs and associated devices.



- The **License Key** is a PKID and a unique identifier for the license.
- **Available Seats** refers to the number of upstream seats on the server that are available for activation.
- **Claimed Seats** refers to the seats claimed by this instance of Aurora Focus software but not yet provisioned to an FS/VS device.
- **Unsynced Provisioned Seats** refers to the number of seats claimed by Aurora Focus and provisioned to a device.

NOTE: Unclaimed seats can be released back to the license server or can be used to provision another device.

Best Practices for Provisioning Devices

Review the following best practices when provisioning a device to a license.

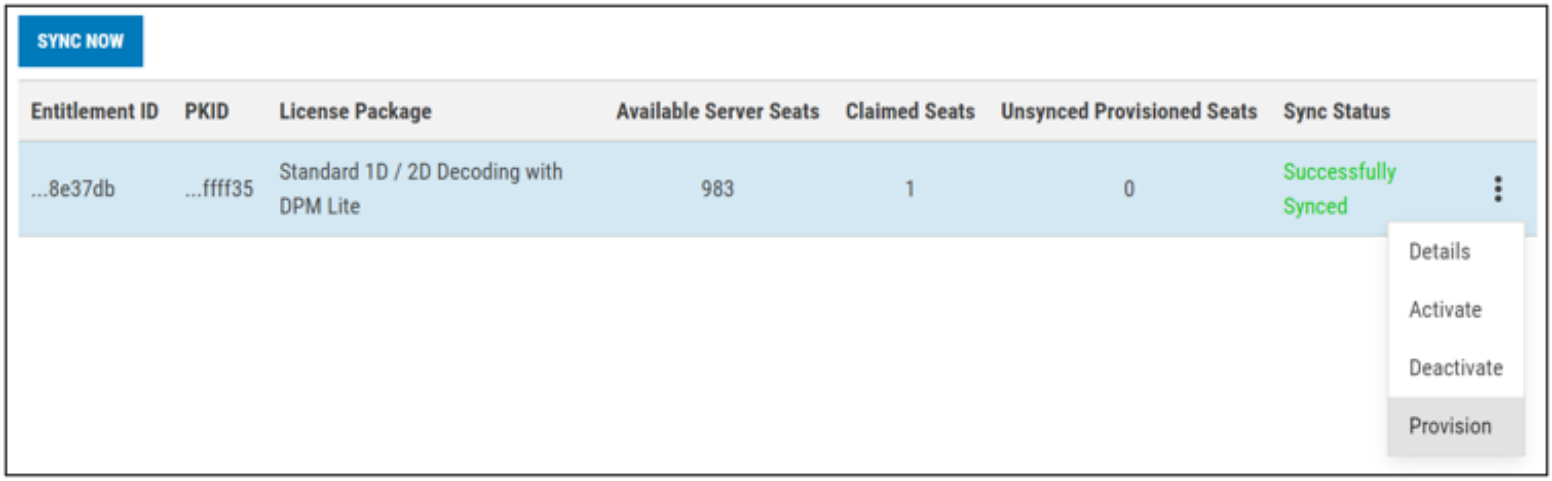
- Activate only the required number of seats that you want to use for provisioning devices.
- Do not hold on to claimed seats in Aurora Focus longer than needed.
  - This enables another user to use them to provision other devices.
- Syncronize provisioned seats to the license server frequently to ensure you are utilizing the active licenses on the device.

Provisioning a License to a Device

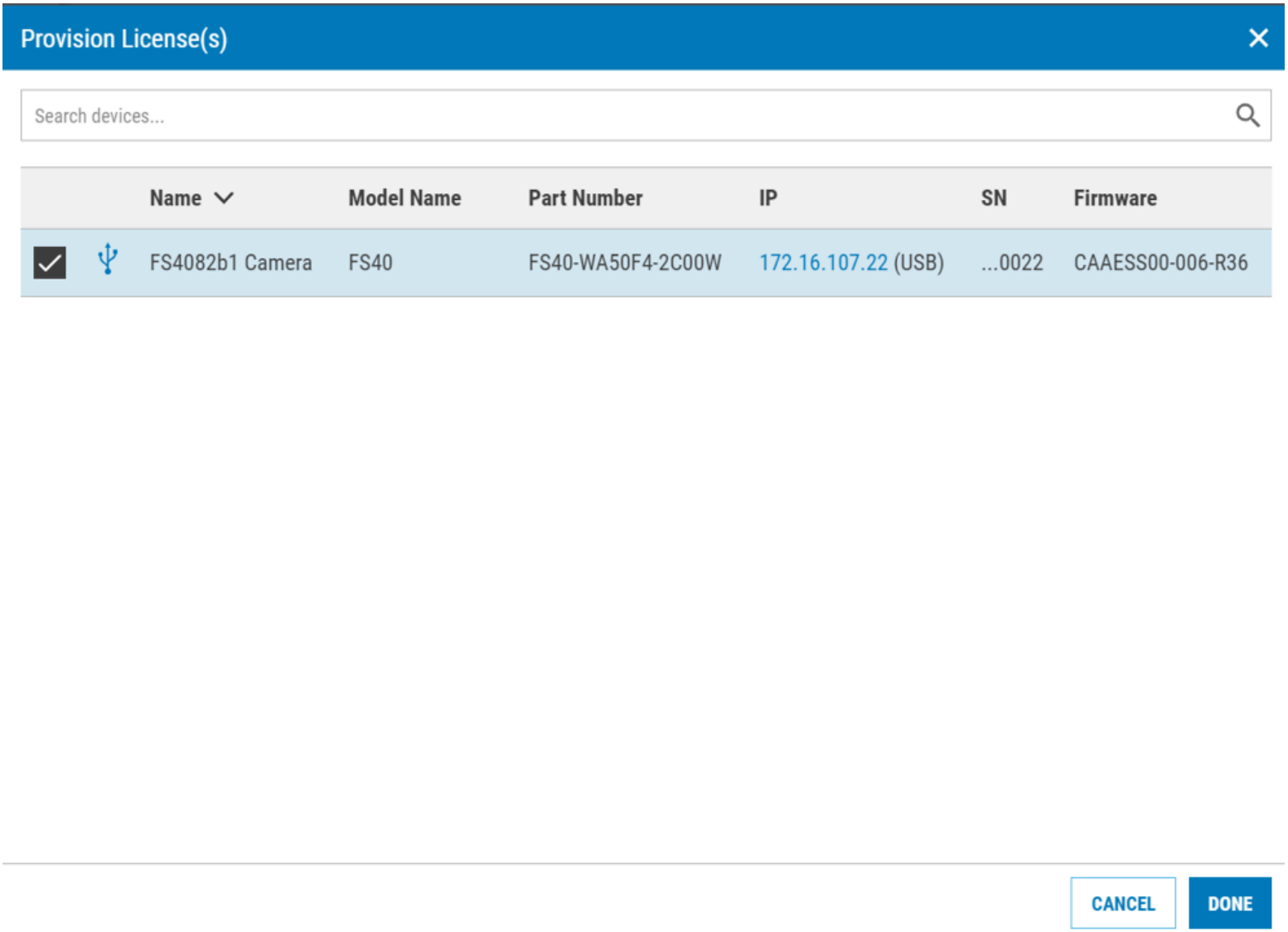
Use the Device Licensing tab in Aurora Focus to provision licenses to devices remotely.

NOTE: Provisioning licenses to devices without internet access while the devices are connected to Aurora Focus.

1. Click ⋮ and select **Provision** from the menu.



2. Select the device to provision the license to and click **Done**.



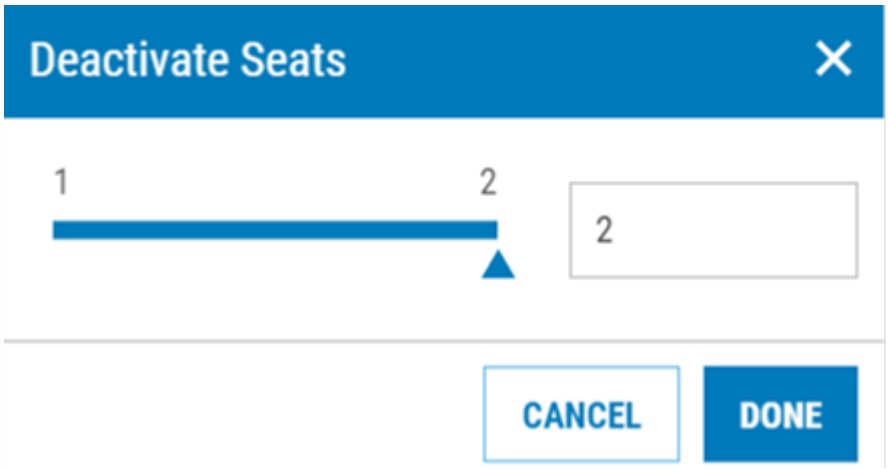
Observe the **Device Successfully Provisioned** notification.

Deactivating a License

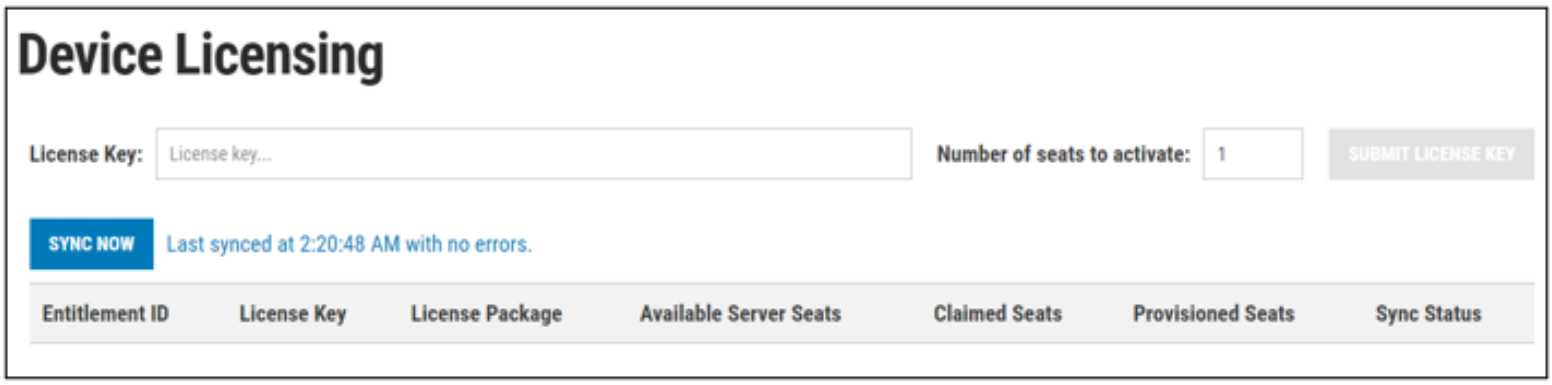
Deactivate a license using Aurora Focus.

1. Click ⋮ and select **Deactivate** from the menu.

2. Select the number of seats to deactivate.



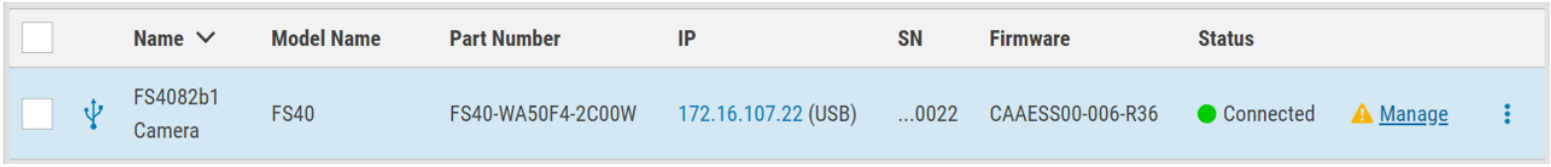
3. Click **Sync Now** to view the updated list of license keys.



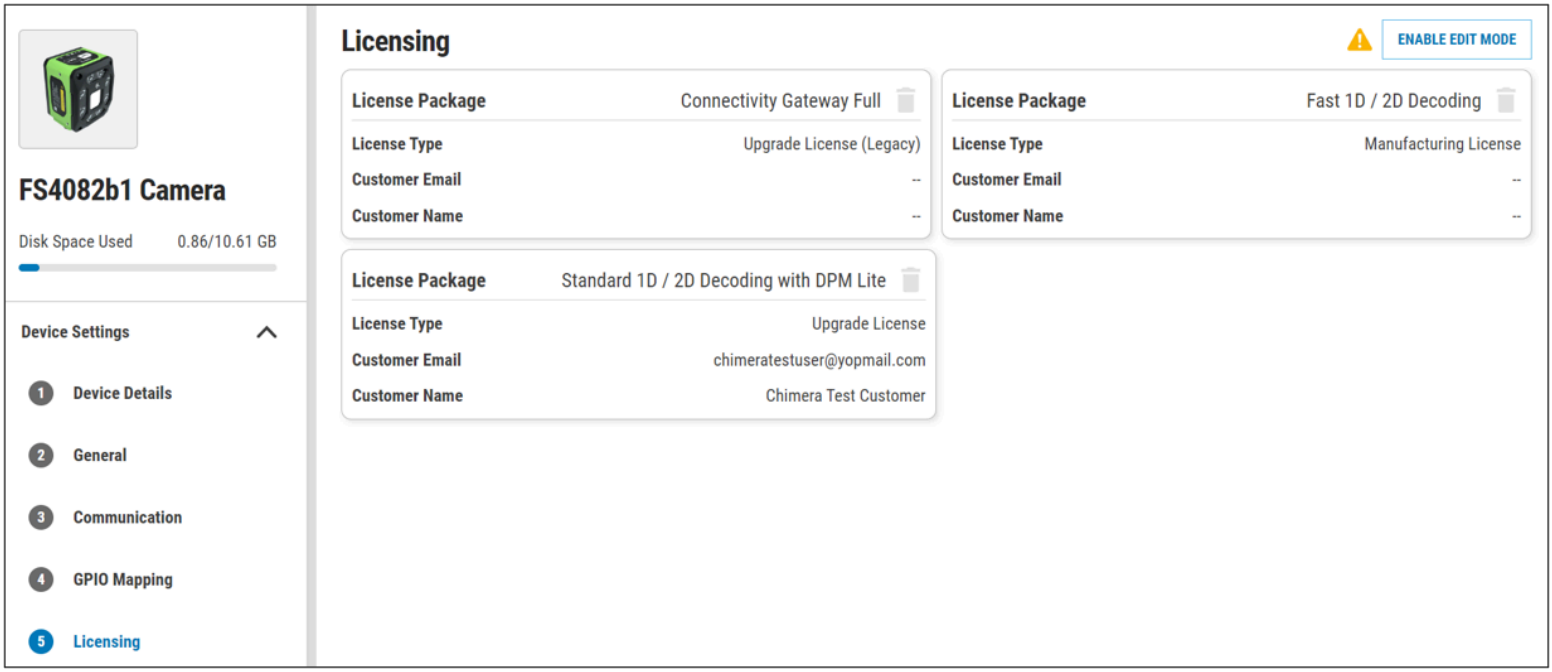
Deprovisioning a License from a Device

Devices can be deprovisioned using Aurora Focus and released back to the License server as needed.

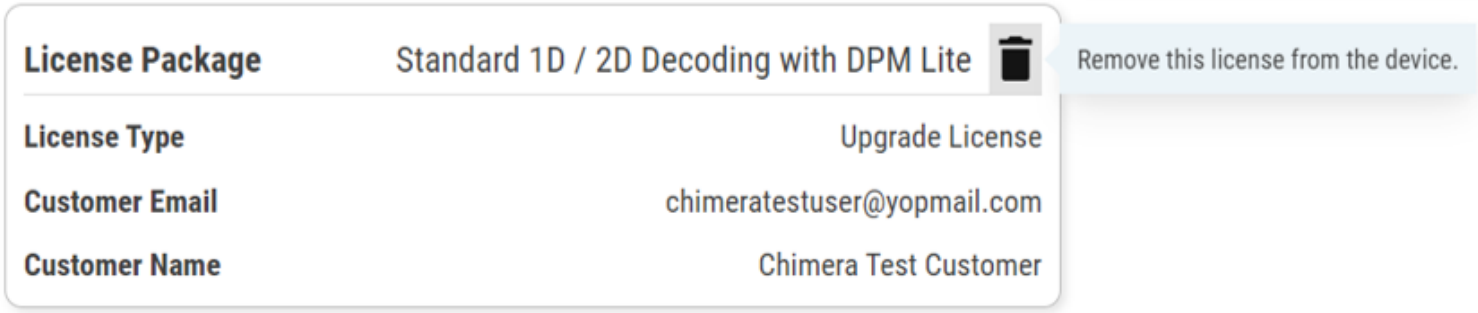
1. Go to **View Devices** and ensure that the device is managed by clicking **Manage**.



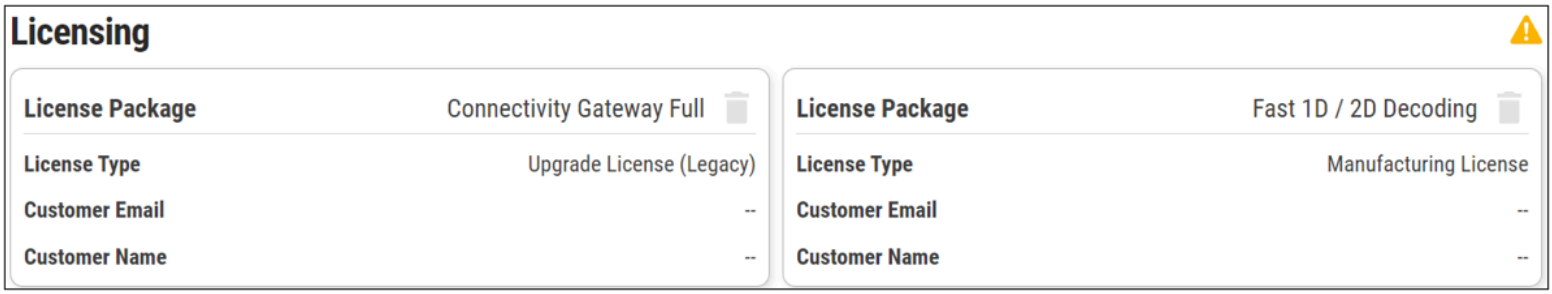
2. Go to the **Licensing** tab in **Device Settings** and click **Enable Edit Mode**.



3. Click to release the license back to the server.



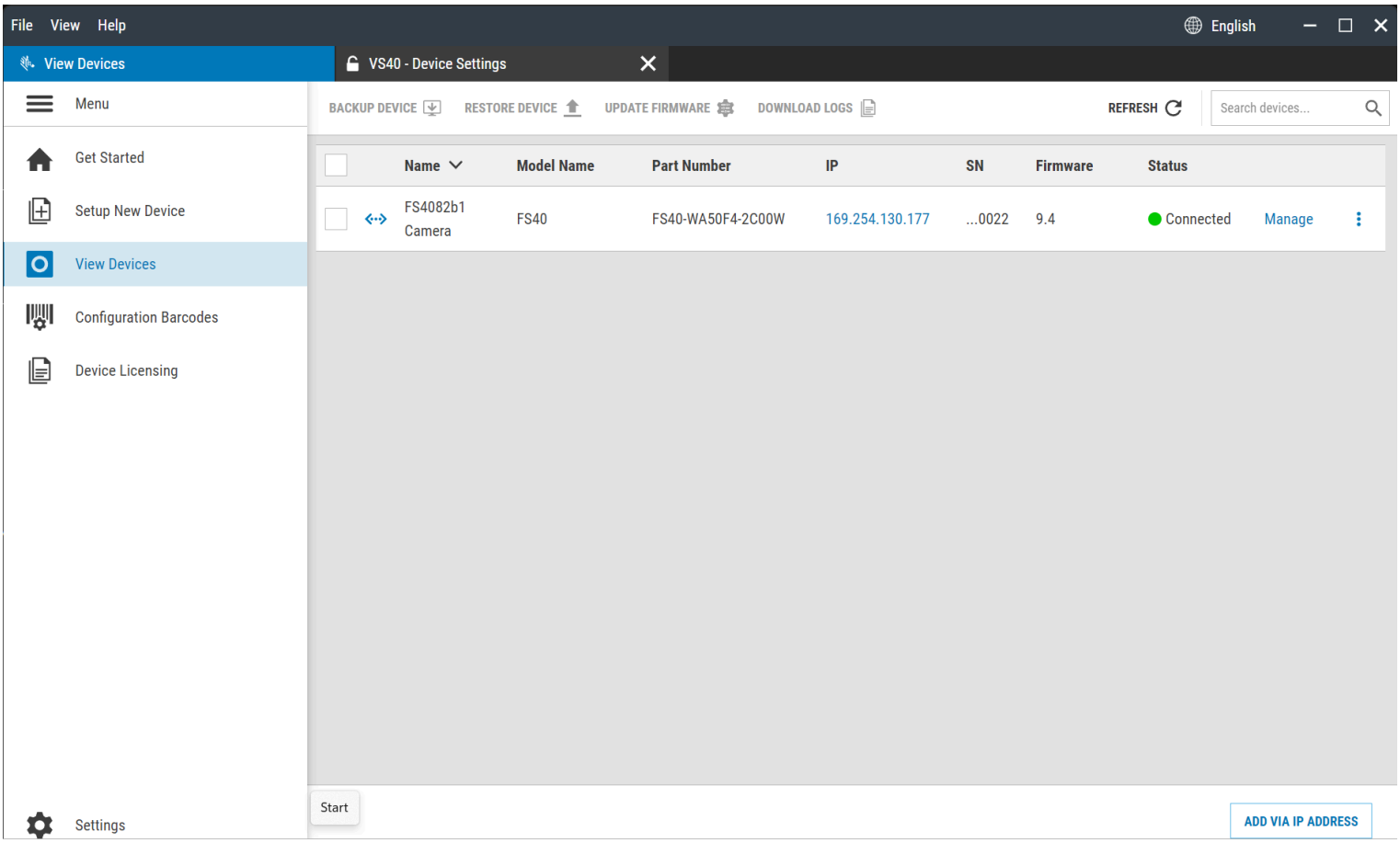
4. Close the **Device Settings** tab and unmanage the device.
5. Return to the **Device Licensing** tab to confirm that the deprovisioned license was released and returned to the pool of claimed seats.



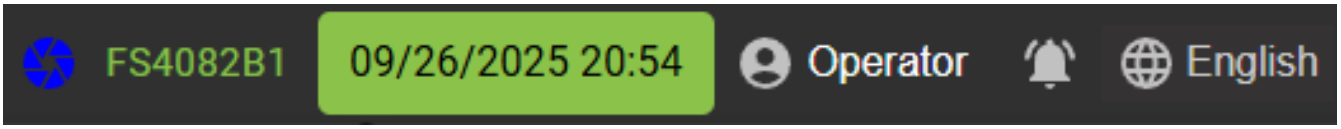
Activating a License Using the Web HMI

Apply a device license, view available licenses, and deactivate licenses online using the Aurora Focus Web HMI.

1. Open the Aurora Focus Web HMI by entering the device's IP address into a web browser or clicking the IP address link under **View Devices** in Aurora Focus.

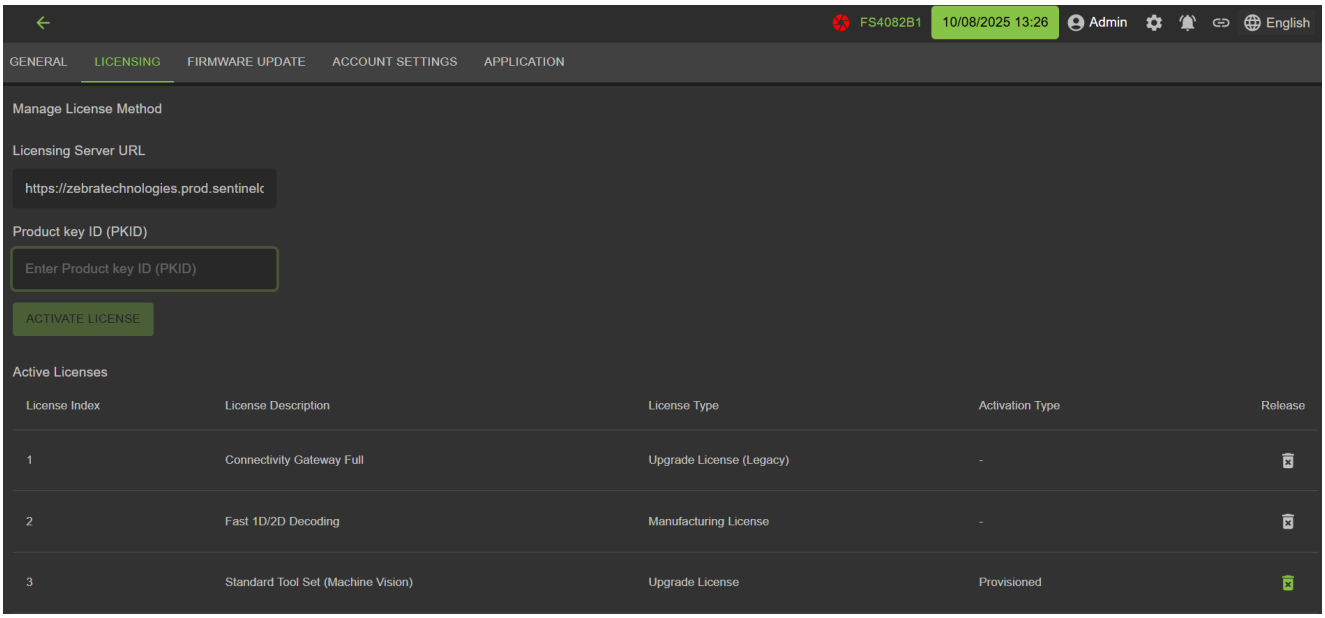


2. If you are logged in as an **Operator**, click to log in to the HMI as an **Admin**.



3. Use the default credentials for an **Admin** account:
   a. Username: admin
   b. Password: admin
4. After you log in as an **Admin**, click ⚙ to access the settings menu.

5. Select the **Licensing** tab.



6. Ensure the following URL is present in the **Licensing Server URL** form field:

   https://zebratechnologies.prod.sentinelcloud.com/ems/api/
7. Enter the **Product Key ID (PKID)** provided in the email when the license is procured.
8. Click **Activate License**.
9. The **Active Licenses** provides the following information:
   a) License Index: displays the license number.
   b) License Description: displays the license's feature or toolset name, for example, Connectivity Gateway Full or FS1D/2D Decoding.
   c) License Type: indicates the type of license and system version on which it was acquired. For example, Fast 1D/2D Decoding with DPM Full.
      - Upgrade License (Legacy) - an upgrade license activated using the legacy system. For example, Advanced Toolset (Machine Vision).

- Upgrade License - an upgrade license using the current system. This license type can be provisioned in Aurora Focus and released in the Web HMI.

d) Activation Type - licenses applied using the current system are provisioned.

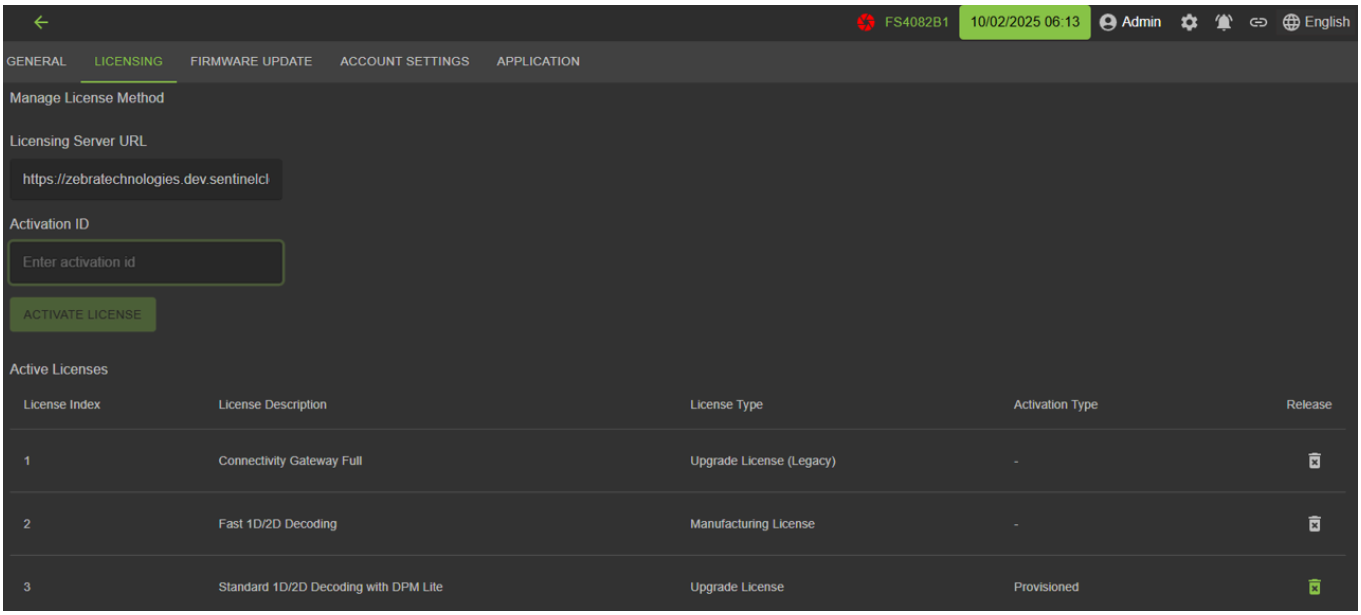
e) Release: Click to release the license back to the device's license server.

Deactivating a License Using the Web HMI

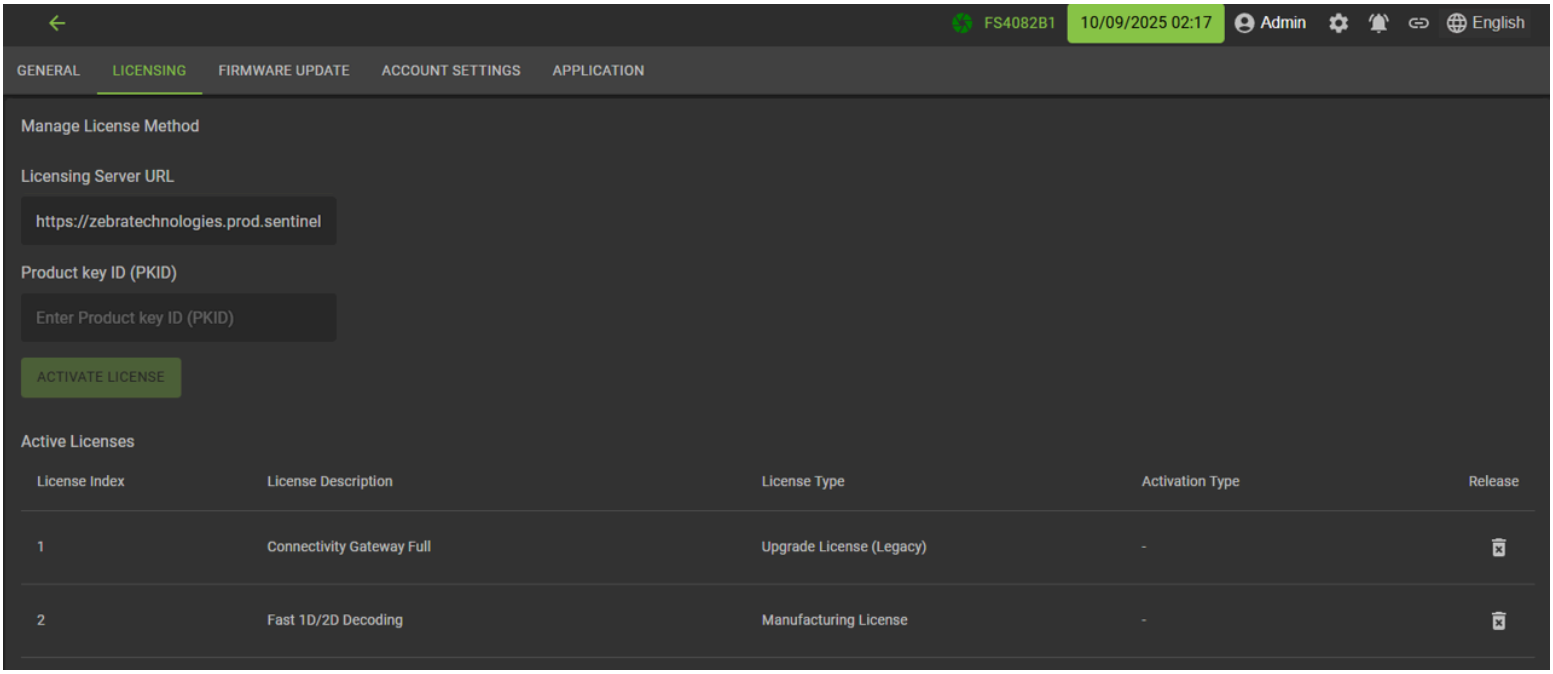
Use the Web HMI to deactivate a license on the device while connected to the internet.

NOTE: Only provisioned licenses can be released.

1. Click to release the license back to the license server.



2. Observe that the license is removed from the list of **Active Licenses**.

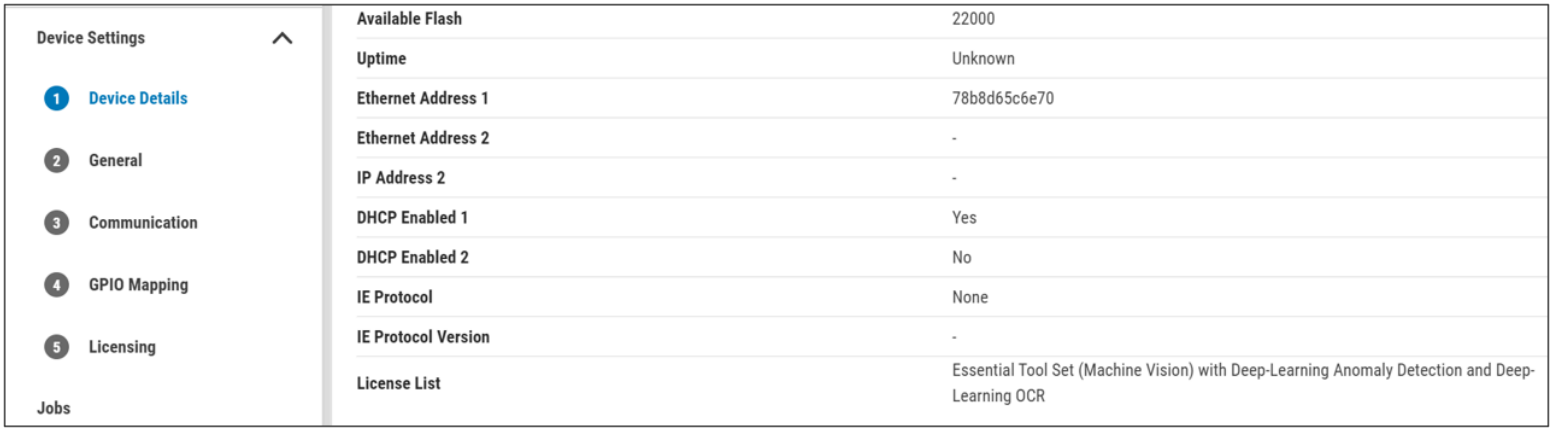


Viewing Active Licenses on the Device

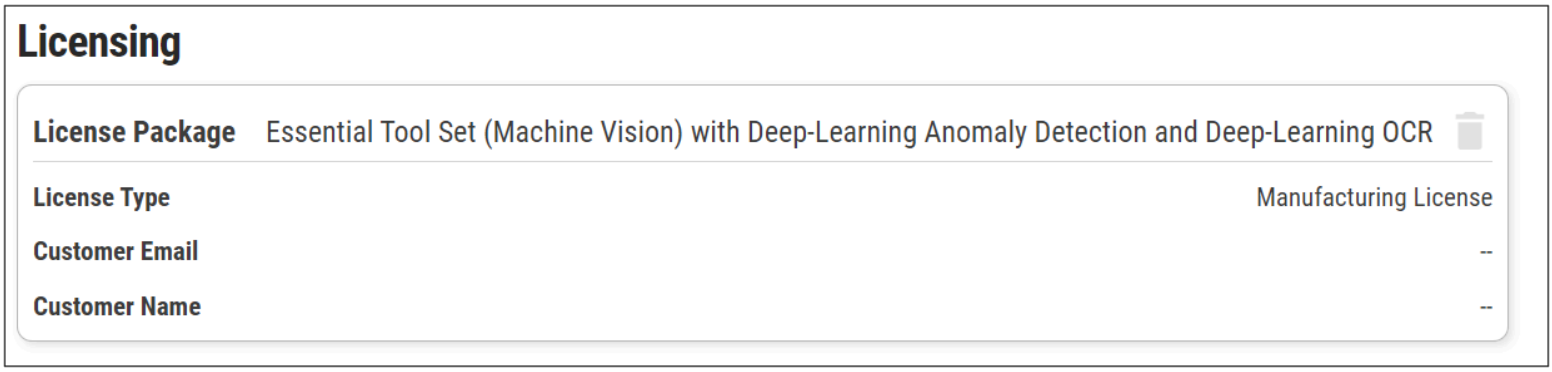
Use Aurora Focus to view active licenses on your device.

View active licenses on the device by navigating to **Device Settings**.

1. Go to **Device Details** and view the **License List** on the bottom of the page.



2. Go to **Licensing** to view the current **License Package** and **License Type** currently enabled on the device.



Bridging an Internet Connection to FS/VS Devices

To add a license to an FS/VS device using an online licensing server, the device must have access to the internet. The device can be connected through a router connected to the internet. But in cases where that is not possible or convenient, the internet connection of a laptop or desktop computer running Windows can be shared with the device. To share internet to a device connected to the computer over USB, these steps must be performed on the device first (does not need to be done for devices connected over Ethernet):

1. Open Aurora Focus.
2. Manage the device to which you want to share Internet.
3. Navigate to the **Communication** tab.
4. Click **Enable Edit Mode** and **OK** if prompted with a warning about stopping jobs.
5. Navigate to the **USB** tab under **Network Settings.**
6. Select **Enable DHCP**, and click **Apply.**

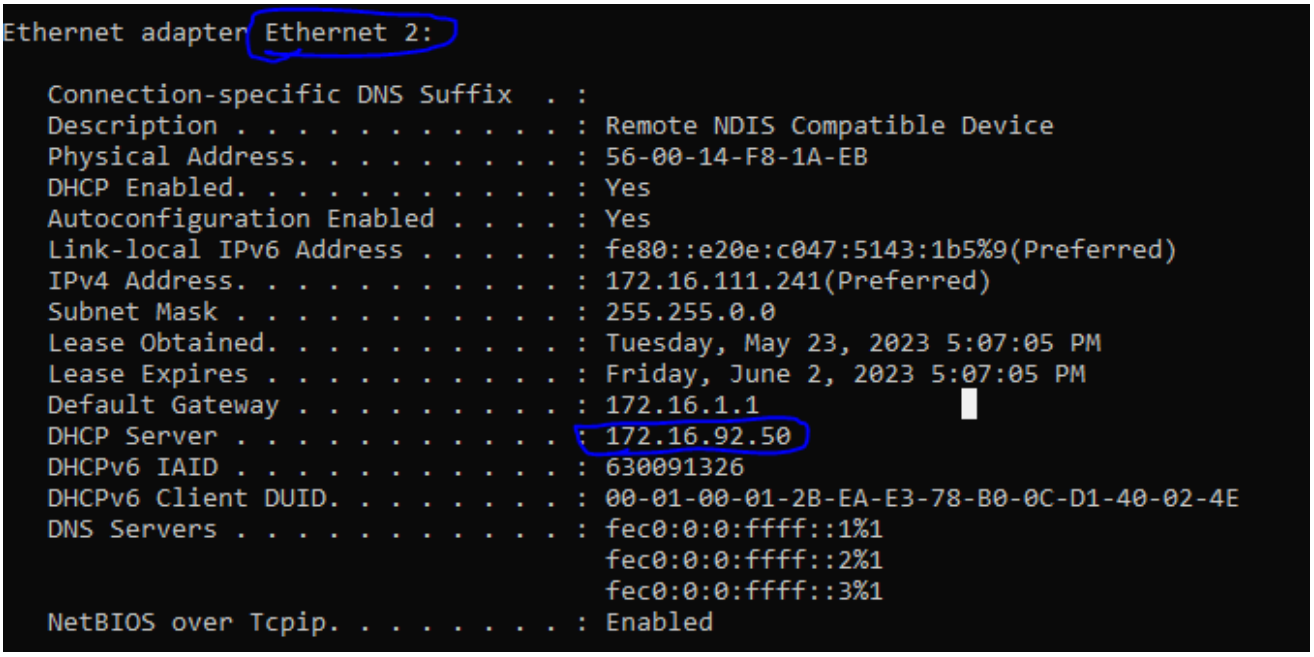
To share the internet connection to a device when connected over USB or when plugged into a PoE switch that is plugged into the computer or dock's Ethernet port:

1. Navigate to **Windows Control Panel > Network and Sharing Center > Change adapter settings**.
2. Right-click on the connection that provides Internet. This is typically Wi-Fi, but can also be Ethernet.
3. Click **Properties**.
4. Click the **Sharing** tab.
5. Select **Allow other network users to connect through this computer's internet connection.**.
6. In the **Home networking connection** drop-down, select the connection over which you want to share internet, if the option appears. This may be something such as Ethernet or Ethernet 5. If there is only one option for the connection, the drop-down does not display.
7. If there are multiple Ethernet options, to find out which adapter is using the USB RNDIS, open a command prompt and type ipconfig /all. Locate your device's USB IP (it typically matches a DHCP Server, as shown in the screenshot below). Locate the Ethernet adapter name for that section. Select that option in the menu.
8. Click **OK**.
9. Reconnect and reboot the device.

To share the internet connection to multiple devices connected over USB and Ethernet simultaneously:

1. Navigate to the **Windows Control Panel > Network and Sharing Center > Change adapter** settings.
2. Ctrl-click the USB or Ethernet connections that the devices are connected to so they are highlighted simultaneously.
3. Right-click any of the highlighted connections and click **Bridge connections**.
4. Right-click the connection that is providing Internet.
5. Click **Properties**.
6. Click the **Sharing** tab.
7. Select **Allow other network users to connect through this computer's internet connection**.
8. On the **Home networking connection** menu, select the bridged connection you just made if the option appears at all. If there is only one option for the connection, the drop-down will not display.
9. Click **OK**.
10. Reconnect and reboot the device(s).

Figure 14 Identifying the DHCP Server



To test that a device has access to the internet, SSH into it and run **ping google.com**. If the connection to the server is successful, the device can connect to a licensing server over the internet.

Frequently Asked Questions

Review the answers to frequently asked questions in this section to understand the licensing procedure.

What happens if I update the firmware of an FS/NS/VS device to a version matching Aurora Focus 9.4 or greater?

- Factory-applied licenses continue to work, while previously applied device upgrade licenses also continue to work unless a device factory reset operation is performed beforehand with the license

storage option selected or the device's license storage is corrupted. If this occurs, re-activate the license using the information in the Zebra License Manager Migration Notification email.

Can I continue to use an FS/NS/VS device with firmware matching Aurora Focus 9.3 or lower after December 31, 2025?

- Factory-applied and device upgrade licenses are perpetual, so they will continue to work after both dates.

How do I recover from a deleted license storage on a device with firmware matching Aurora Focus 9.3 or lower after December 31, 2025?

- If the license was applied using the offline activation method, upgrade the device to firmware matching Aurora Focus 9.4 or greater and re-activate the license using the information provided in the Zebra License Manager Migration Notification email.

How do I handle replacing a device with firmware matching Aurora Focus 9.3 or lower with device upgrade licenses activated after the release of Aurora Focus 9.4 with matching device firmware?

- Obtain a replacement device with the equivalent factory-applied licenses for Aurora Focus 9.3 or lower and Aurora Focus 9.4 or greater with matching firmware. You can also have the replacement device with firmware matching Aurora Focus 9.4 or greater and obtain the required device upgrade license using the standard method.

What happens if I obtain a device upgrade licenses before the release of Aurora Focus 9.4 and matching device firmware but do not activate all of them before December 31, 2025?

- To activate the remaining device upgrade licenses, you must upgrade to Aurora Focus 9.4 or greater, Then, you must match the device firmware and use the information provided in the Zebra License Manager Migration Notification email.

Can I obtain a device emulation license for Aurora Focus 9.3 or lower after the release of Aurora Focus 9.4?

- No

Does a device emulation license obtained before the Aurora Focus 9.4 release and activated before December 31, 2025, continue to work after that date?

- Yes, the device emulation license is valid until its 1-year term expires.

Do the part numbers or SKUs for FS/NS/NS device upgrade licenses remain the same when the new licensing system starts?

- Yes, the part numbers or SKUs remain the same.

Are Aurora Design Assistant, Aurora Imaging Library, and Aurora Vision Studio/Library affected by the Zebra Software Licensing system change?

- They are not affected because they utilize different licensing systems that are not currently changing.

Connectivity Guidelines

Connectivity options are configurable in Device Settings. To access Device Settings, select a device on the View Devices tab and click Manage. On the Device Settings tab, click Communication to configure connectivity settings.

Network Communication

Follow the procedures in this section to set a static IP address or change the hostname.

Setting a Static IP Address

NOTE: DHCP is the default mode for IP addressing.

1. Disable the **Enable DHCP** checkbox.
2. Change the network-related settings based on your network configuration. For example:
    a. Open a command prompt.
    b. Type the command: ipconfig
        a. All TCP/IP network configuration values display on the host PC.
    c. Identify a valid network interface:
    d. In Aurora Focus, enter the following values:
        a. IP Address: 192.168.4.xxx (where xxx is any value from 1-255)
        b. Subnet Mask: 255.255.252.0
        c. Default Gateway: 192.168.4.1
3. Click **Apply** when complete. The device reboots with the new static IP address.

Changing the Hostname

To change the hostname on the device:

1. Locate the hostname field.
2. Enter the desired hostname:

NOTE: The hostname must be alphanumeric and have a maximum length of 26 characters.

3. Click **Apply** when complete.

NOTE: The device, application, and network hardware may need to be power-cycled for the hostname change to occur.

NOTE: The new hostname is visible following the device reboot.

TCP/IP Communication

Configure TCP/IP communication settings to send and accept triggers.

Setting Up TCP/IP Triggering

To set up TCP/IP triggering, follow the steps to enable TCP/IP output in device settings, configure jobs on the device to accept TCP/IP triggers, and send TCP/IP triggers to the device.

Enabling TCP/IP Output in Device Settings

NOTE: TCP/IP output is disabled by default and must be explicitly enabled in the device's settings.

1. Select the representative communication type tab. For example, Ethernet Port 1.
2. Scroll to TCPIP Settings.
3. Check the Enable TCP/IP Control checkbox (disabled by default).

NOTE: Make note of the **Control Part Number** and **Trigger String** fields. This information is required to send a TCP/IP trigger from external software.

4. Click **Apply** in the TCP/IP settings window. Click **OK** in the following prompt.

Configuring Jobs to Accept TCP/IP Triggers

1. Create or open a job on the device.
2. Within the job tab, click the **Connect** tab and navigate to the **Triggers** tab.
3. Ensure the **Trigger Source** is set to **TCP/IP**.
4. Click **Deploy** when the configuration is complete.

Setting up RS-232 Hardware

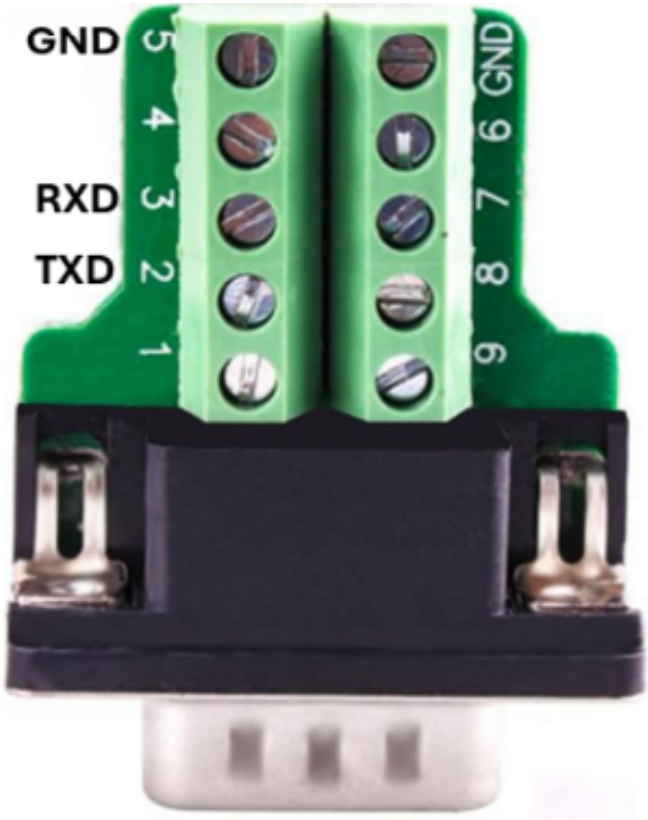
The following accessories are necessary for a serial interface connection:

- Power/GPIO Flying Leads cable
- USB to Serial Adaptor

- DB9 RS-232 D-SUB Serial Adaptor

1. Connect the following wires from the flying leads cables to the DB9 Serial adaptor.

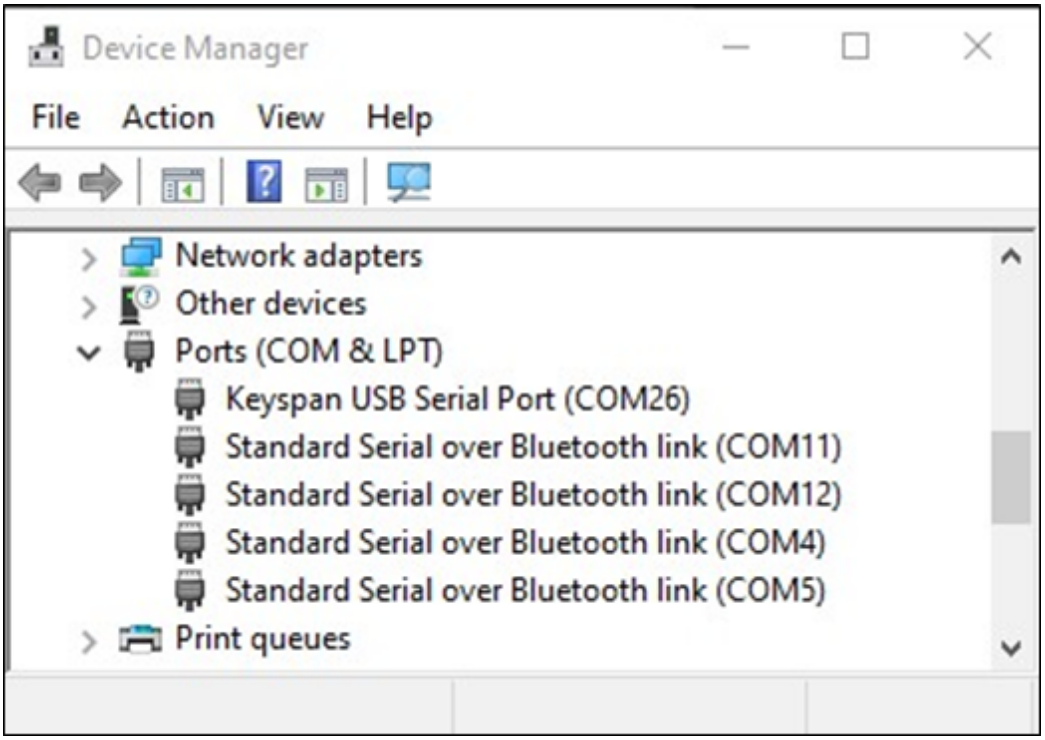
Figure 15 Flying Leads to DB9 Serial Adapter Wire Diagram



Pin	Color	Description
1	Yellow	OUT1
2	White/Yellow	TXD
3	Brown	RXD
4	White/Brown	IN1
5	Violet	RTS
6	White/Violet	COMMON_IN
7	Red	DC_IN
8	Black	GND
9	Green	COMMON_OUT
10	Orange	IN0
11	Blue	OUT0
12	Grey	CTS
SHELL	Bare	SHIELD

2. Connect DB9 Serial Adapter (female) to USB-to-Serial adapter (Male).

3. Connect USB-to Serial adapter USB end into host PC.
4. Identify the COM port associated with the USB-to-Serial adaptor.
5. Identify the COM port associated with the USB-to-Serial adaptor using Device Manager on a Windows PC.
   a) Expand the Ports field and identify the USB-to-Serial adaptor.



Setting up RS-232 Triggering

Follow the steps in this section to enable RS-232 output in device settings or configure a job on the device to accept serial triggers.

Enabling RS-232 Output in Device Settings

NOTE: TCP/IP output is disabled by default and must be explicitly enabled in the device's settings.

1. Select the respective communication type tab. For example, Ethernet Port 1 or USB.
2. Scroll down to RS-232 settings.
3. Check the **Enable RS-232 Control** checkbox (disabled by default).

NOTE: Observe the **Trigger String** and keep this value for reference to send a Serial trigger from external software.

4. Click **Apply** in the RS-232 area and **OK** on the resulting prompt to save the configuration.

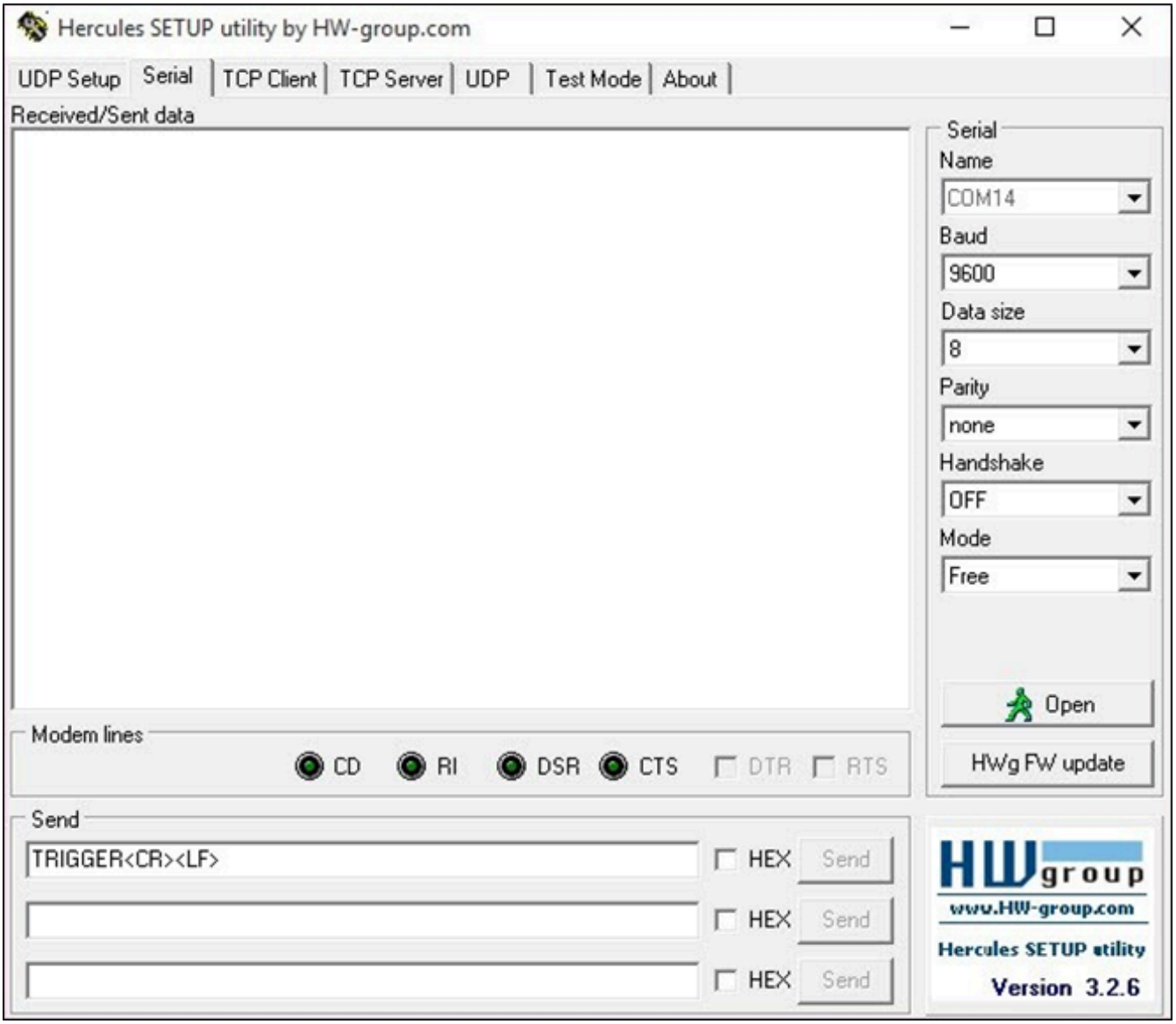
Configuring a Job on the Device to Accept Serial Triggers

1. Create or open a job on the device using Zebra Aurora Focus.
2. On the Job tab, click the **Capture** tab and navigate to the **Triggers** tab to set the **Trigger Source** to **Serial.**
3. Complete the job configuration and click **Deploy** to send the job to the device.

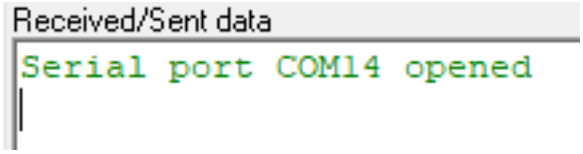
Confirm that the device is able to accept Serial (RS-232) triggers.

Sending a Serial Trigger to the Device

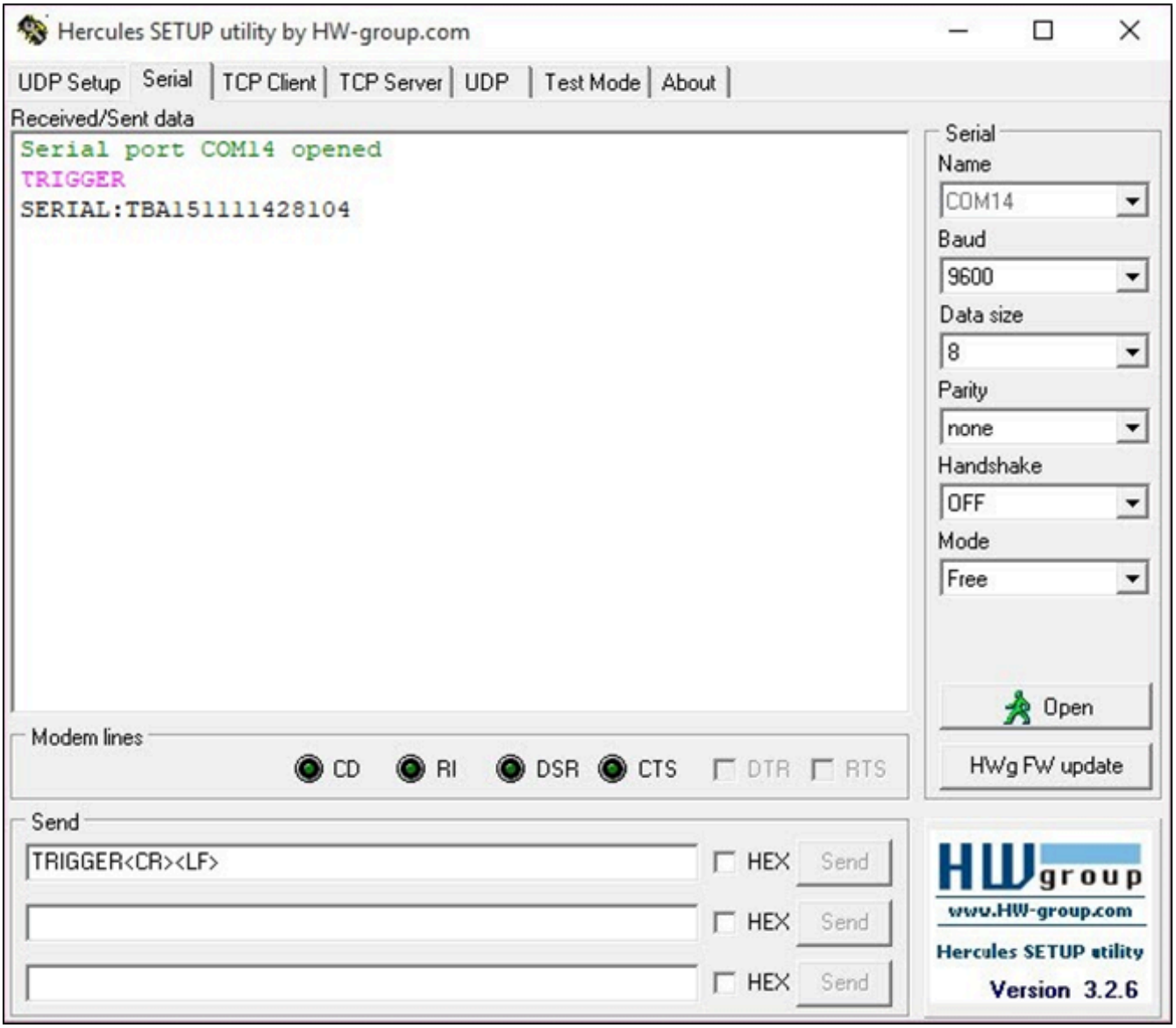
1. Open a Serial client terminal software such as PuTTy.
2. Establish a connection to the device by clicking **Serial** tab.
3. Enter the **COM port** settings:
   - Name: COM port number as defined in Device Manager.
   - Speed, Baud, and Parity as defined in Device Settings using Zebra Aurora Focus.
   - Stop Bits: 1
   - Handshake/Mode: Off
4. Click **Open** to establish a serial connection.



5. If a connection is successfully established, observe the **Serial Port COM opened** message.

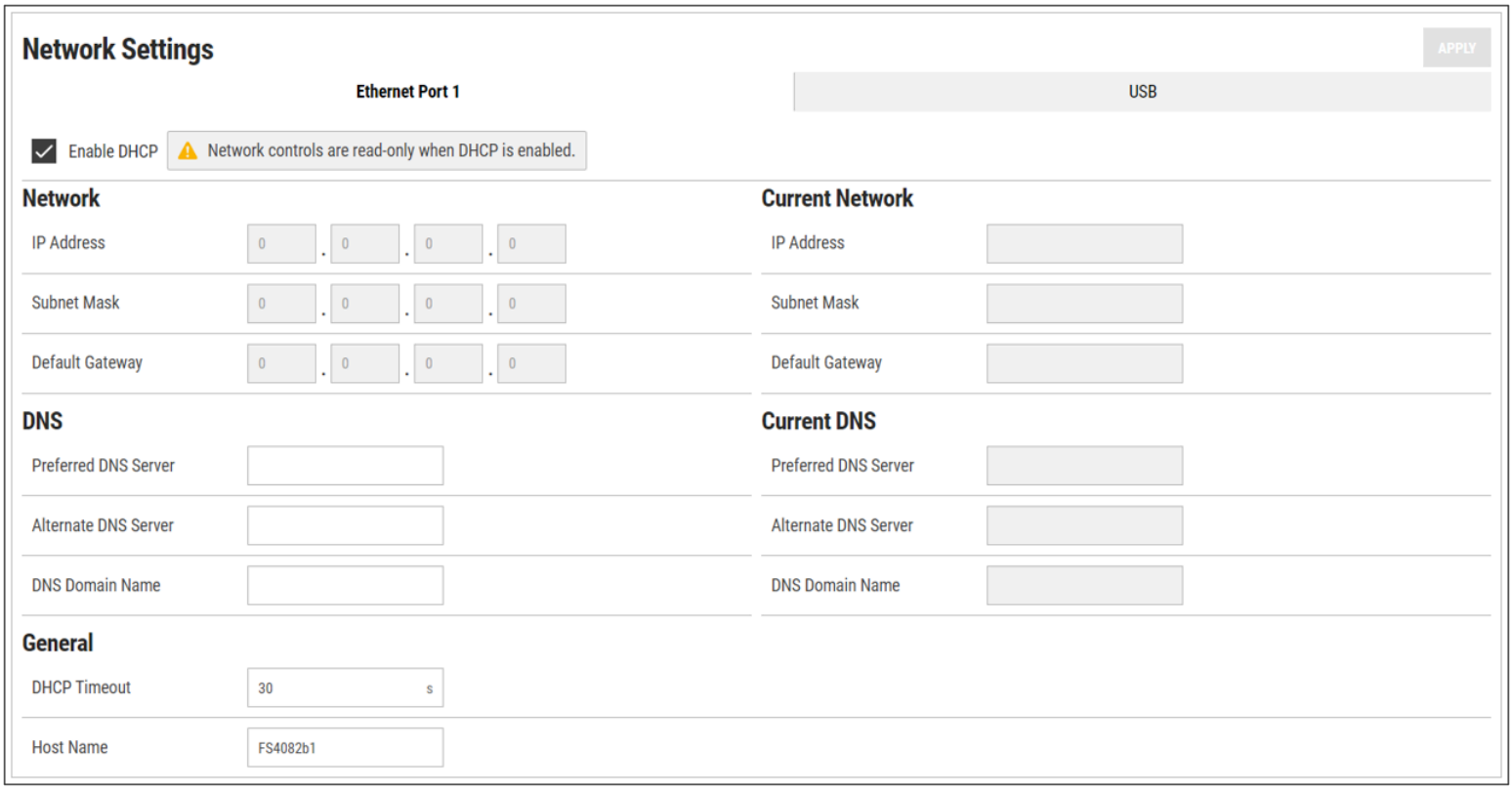


6. Type the **Trigger String** as defined in **Settings** in Zebra Aurora Focus and click **Enter.**
7. Enter **TRIGGER** as the default **Trigger String** and click **Send** to send a trigger using serial.



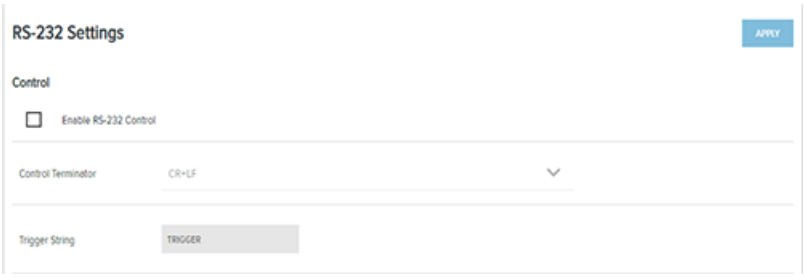
RS-232 Results

NOTE: RS-232 output is disabled by default and must be explicitly enabled in Device Settings.

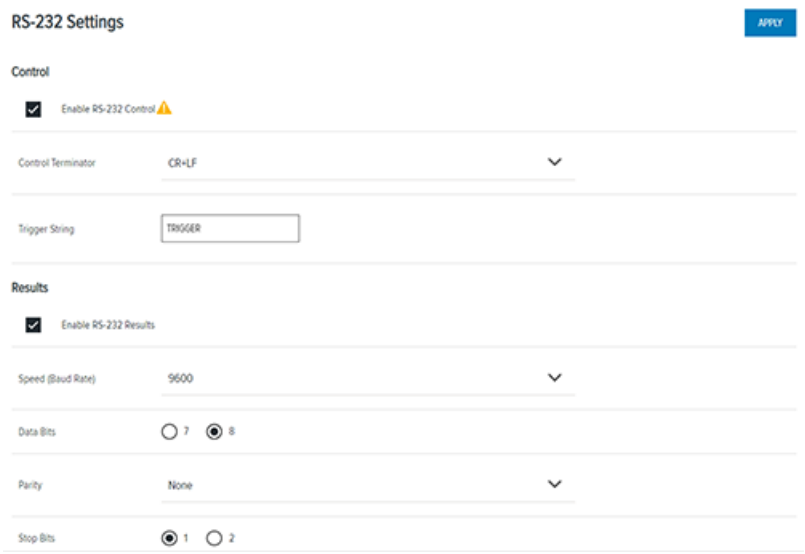


1. Select the associated communication type tab.
2. Scroll down to **RS-232 Settings**.
3. Ensure that the **Enable RS-232 Results** checkbox is checked

NOTE: Record the **Speed, Data Bits, Parity, Stop Bits** you may need these settings later to listen for RS-232/Serial results from external software.

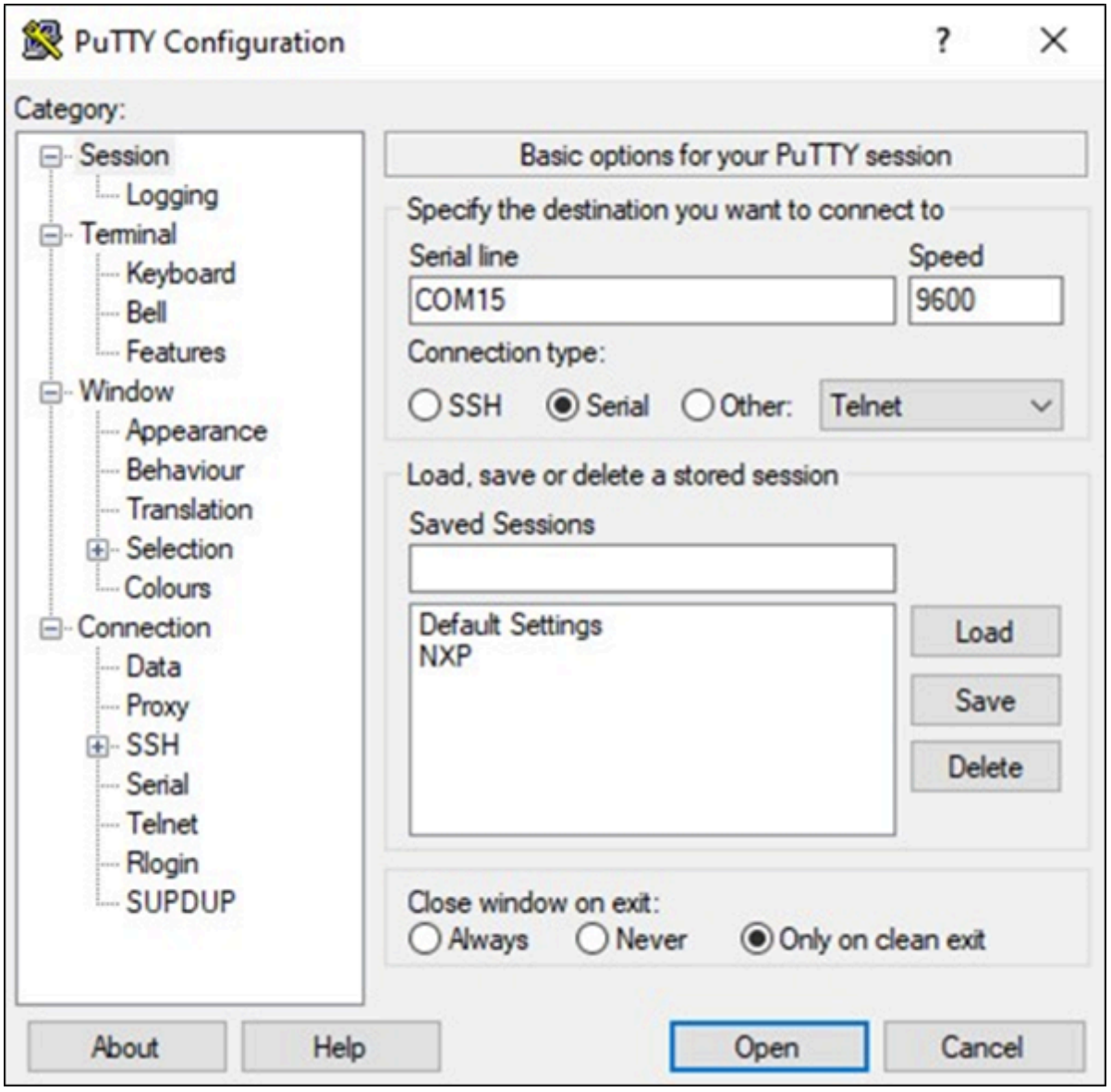


4. Click **Apply** within the **RS-232 Settings** area.



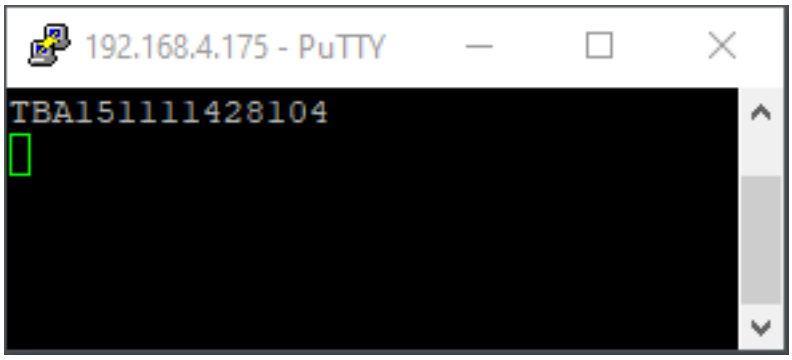
Listening to RS-232 Output Events

1. Open an RS-232 client terminal software such as PuTTy.
2. Establish a connection to the device by clicking **Session**.
3. Set **Serial Line** to the COM Port Number.
4. Set **Speed** to the Serial speed defined in Aurora Focus.
5. Click **Open** to open a serial connection in terminal.



6. If the connection is successful, observe the terminal window with a green cursor.

7. Trigger the job and observe the results in the terminal window to confirm that the device is configured as expected.



Listening for USB-CDC Output Events

To set USB CDC-Serial Results as job output, enable USB CDC-Serial Output in **Device Settings** using Aurora Focus.

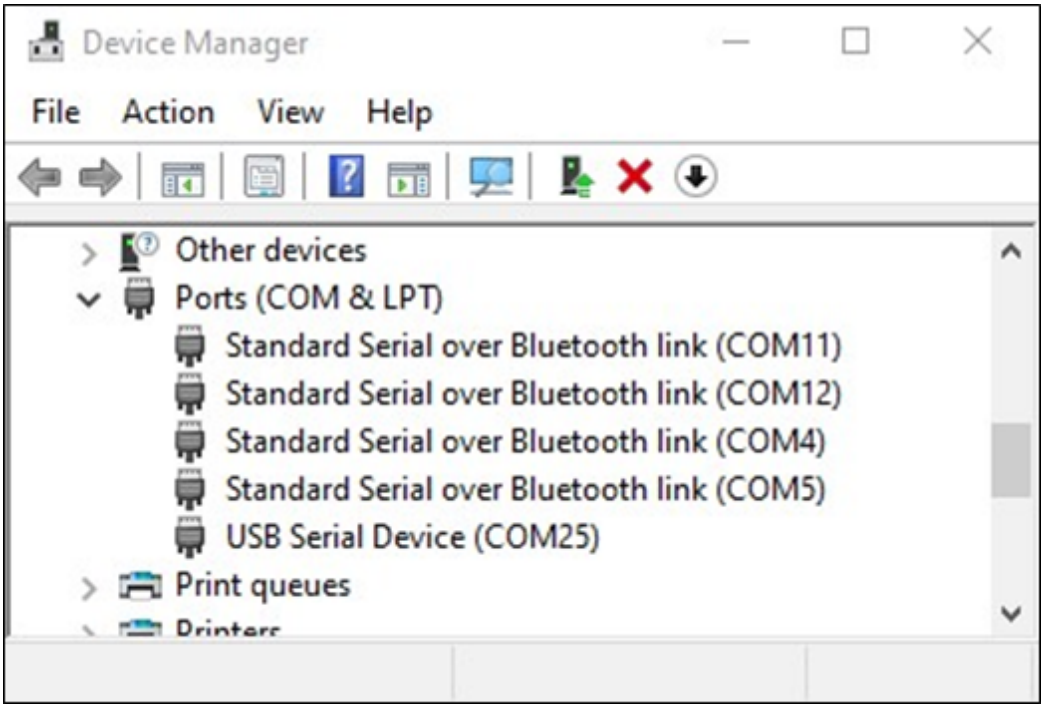
Setting Up CDC-Serial Output in Device Settings

1. Select the associated communication type tab.
2. Scroll to **USB Settings**.
3. Ensure that the **Enable USB CDC-Serial Results** checkbox is enabled.
4. Click **Apply** within the USB Settings.
5. Click **OK**.

Listening Over USB CDC-Serial Connection

The process of listening over USB CDC-Serial connections is similar to the RS-232 setup outlined in the previous section.

1. After enabling USB CDC-Serial Results, navigate to Device Manager to identify the COM port associated with your USB connection.
2. Follow the instructions outlined in the RS-232 sections using the COM port in Device Manager.



USB Settings

Configure USB settings such as enabling a HID Keyboard or adding a Keystring Delay.

Enabling HID Keyboard

Checking this setting will enable USB-HID output. Job result/output will be streamed over a USB connection to the host PC as if the device was a keyboard. The output streams into any field with focus, such as notepad, excel, or other host-based apps.

NOTE: When this setting is applied, the device will automatically reboot. If USB is your sole way of communicating with the device (as opposed to via ethernet cable) and you have a job that automatically starts on startup, it may be difficult to communicate with the device if the job is running and sending output over the USB simultaneously. Ensure that you have de-selected the other options before enabling HID.

Adding a Keystring Delay

Sometimes data is sent over USB faster than the receiving application receives them. This is commonly seen with some remote applications where latency is a factor. Compensate for this by adding a keystroke delay, where a certain pause (typically in ms) is inserted between each character so no characters (data) are missed.

PLC Protocol

For specific details on Industrial Ethernet, EtherNet/IP, PROFINET Interface, and Modbus TCP industrial communication protocols, refer to the FS/VS Industrial Ethernet User Guide.

Communication Port Usage

The following table lists the ports used by Zebra Aurora Focus.

Table 77 Zebra Aurora Focus Communication Port Usage

Port	TCP	UDP
FTP - Data	20	
FTP - Communication	21	
DHCP Server		67
DHCP Client		68
Web HMI	80	443
NTP		123
Modbus	502	502
EtherNet/IP		2222
Discovery Port	3702	3702
TCP Port	4444	
mDNS		5353
mDNS Responder		5354
Listening Port	5555	5555
Communication Port	5556	5556
Output Image	7722	7722
Image Live View	7777	7777
Discovery Port (2)	8889	
Discovery Port (3)	9876	
Raw TCP Data	25250	
Profinet RT Unicast	34962	34962
Profinet RT Multicast	34963	34963
Profinet RT CM	34964	34964
EtherNet/IP	44818	44818
Profinet		49152
Reserved for Gateway Functionality	4012	4012

Table 77 Zebra Aurora Focus Communication Port Usage (Continued)

Port	TCP	UDP
Reserved for Gateway Functionality	4009	4009
Reserved for Gateway Functionality	8080	8080
Listening to Raw TCP Data from Follower Devices	25250	25250
TCP Port (Host to Leader)	25256	25256
MQTT	8883	8883
Reserved for Gateway Functionality	25170	25170
Management Port	5554	
DA Licensing Port	54212	
ZETI	23	
Raw TCP Data	25250	
TCP Control Port (if enabled)	107	

NOTE: TCP Port and Raw TCP Data are user-configurable ports.

Troubleshooting

This section describes potential issues that may occur while using the application and solutions that could correct the problem, such as rebooting the device and reconnecting the USB cable.

Table 78 Troubleshooting

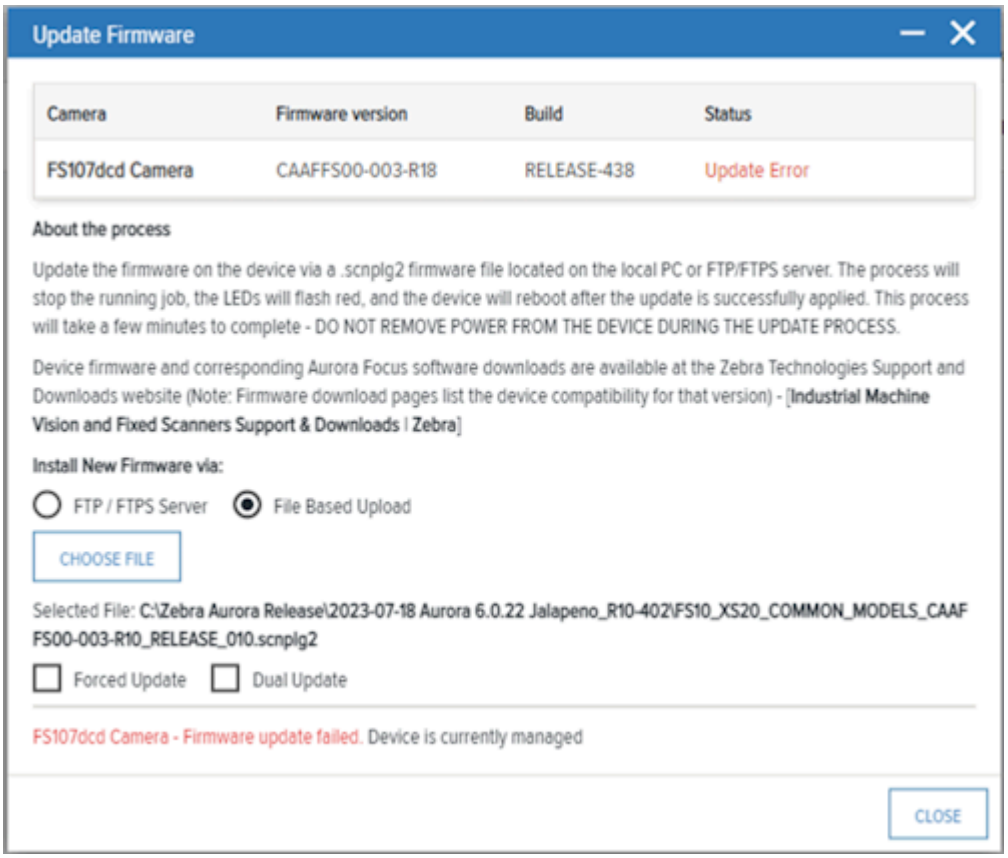
Problem	Solution
Installation	
Run as Administrator	If you receive an error when attempting to install Aurora Focus, try again by right-clicking the Aurora Focus icon on your PC, and selecting Run as Administrator.
Backward Compatibility	Aurora Focus does not connect to a device with older firmware. This may cause a break in functionality if the device is not updated.
Software	
The device becomes idle while configuring an inspection task.	Perform a factory reset. This can resolve issues that may occur after updating the firmware version.
Device	
External illumination does not turn on when running a job created with a previous firmware version.	Enable external illumination.
External illumination does not turn off when in NPN mode.	Switch back to PNP mode before powering off the device.
On FS10 devices, specific hubs cause USB interfaces (RNDIS and HID) to break upon HID keyboard re-enumeration.	Disconnect and reconnect the USB cable.
Firmware updates fail if the memory consumption on the device is currently high.	Reboot the device and upgrade.
Uninstalling the InstallShield does not always remove all of the registry entries.	Type `regedit` to access the Registry Editor on your PC, and manually delete the Zebra Aurora Focus folder under Computer \HKEY_LOCAL_MACHINE\SOFTWARE\Zebra\

Firmware Troubleshooting

Table 79 Firmware Troubleshooting

Problem	Solution
The firmware update procedure fails.	If the device status is set to Managed, close the device and retry.

Figure 16 Managed Device Update Firmware Error

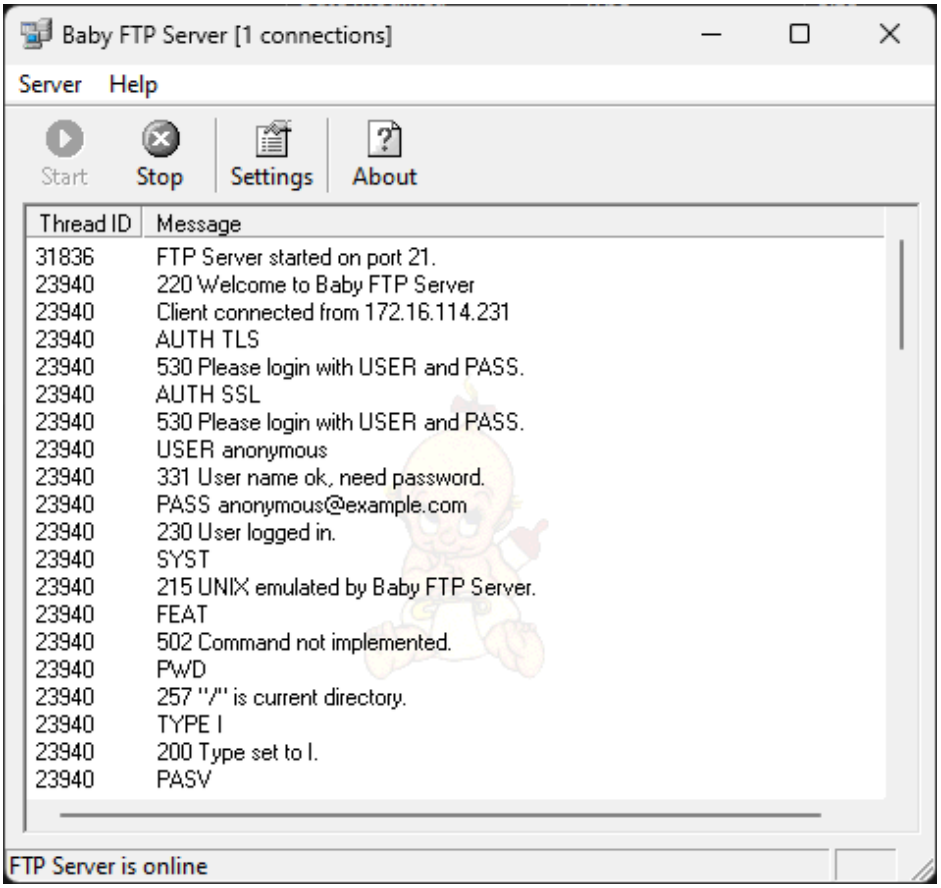


FTP Troubleshooting

If the FTP server is not saving images correctly, it could be a result of an issue with the firewall, folder access or invalid permissions.

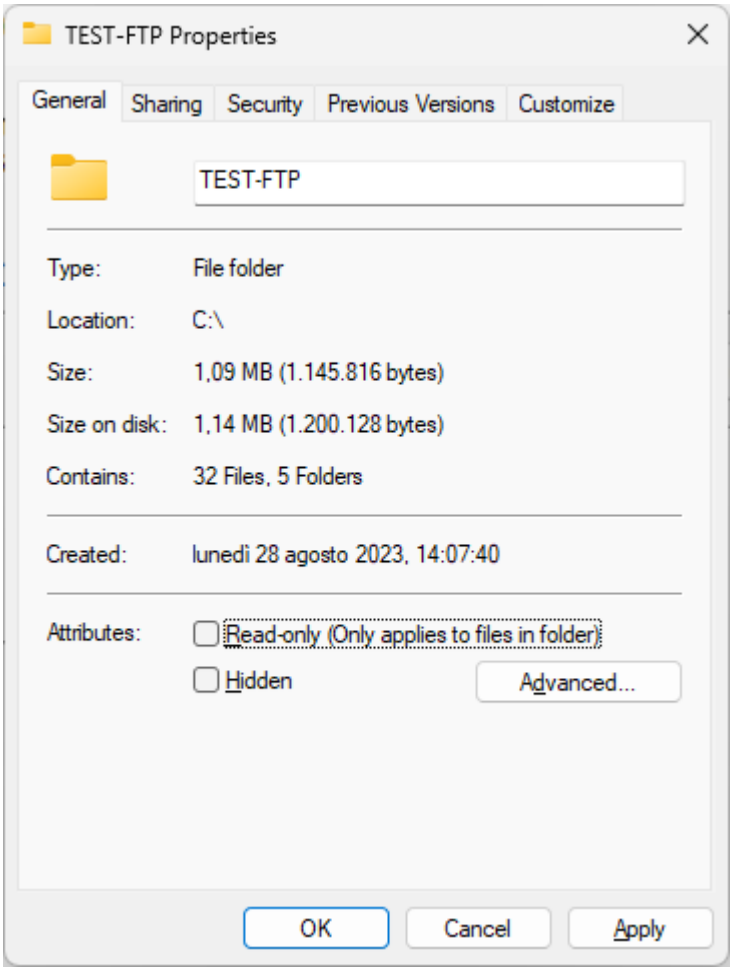
Firewall Settings

If the FTP server is not receiving messages, as in the image below, review the firewall settings.



Invalid Folder Attributes

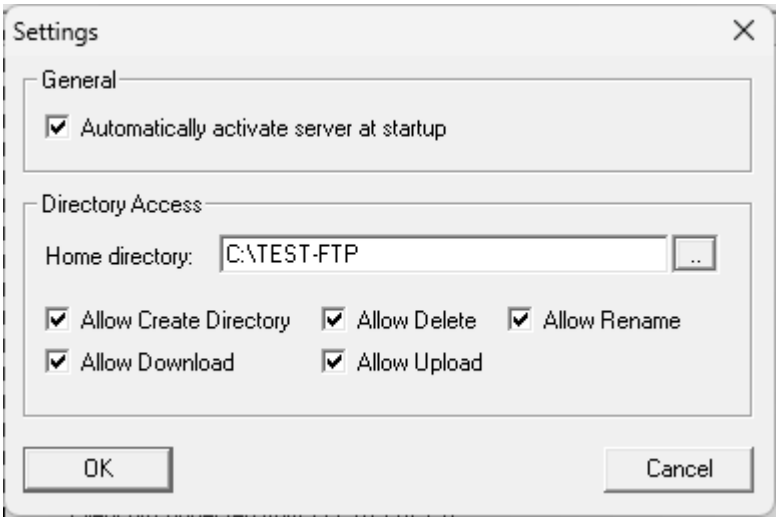
In cases where the FTP server application cannot write on the destination folder. Select the folder and use the **ALT+ENTER** command to open the properties window and verify that the **Read-only** checkbox is not enabled.



Invalid Permissions

The FTP server configuration prevents certain operations. If the log of the FTP server application shows a critical file transfer error.

To create a directory and upload a file, activate all options as shown in the following figure:

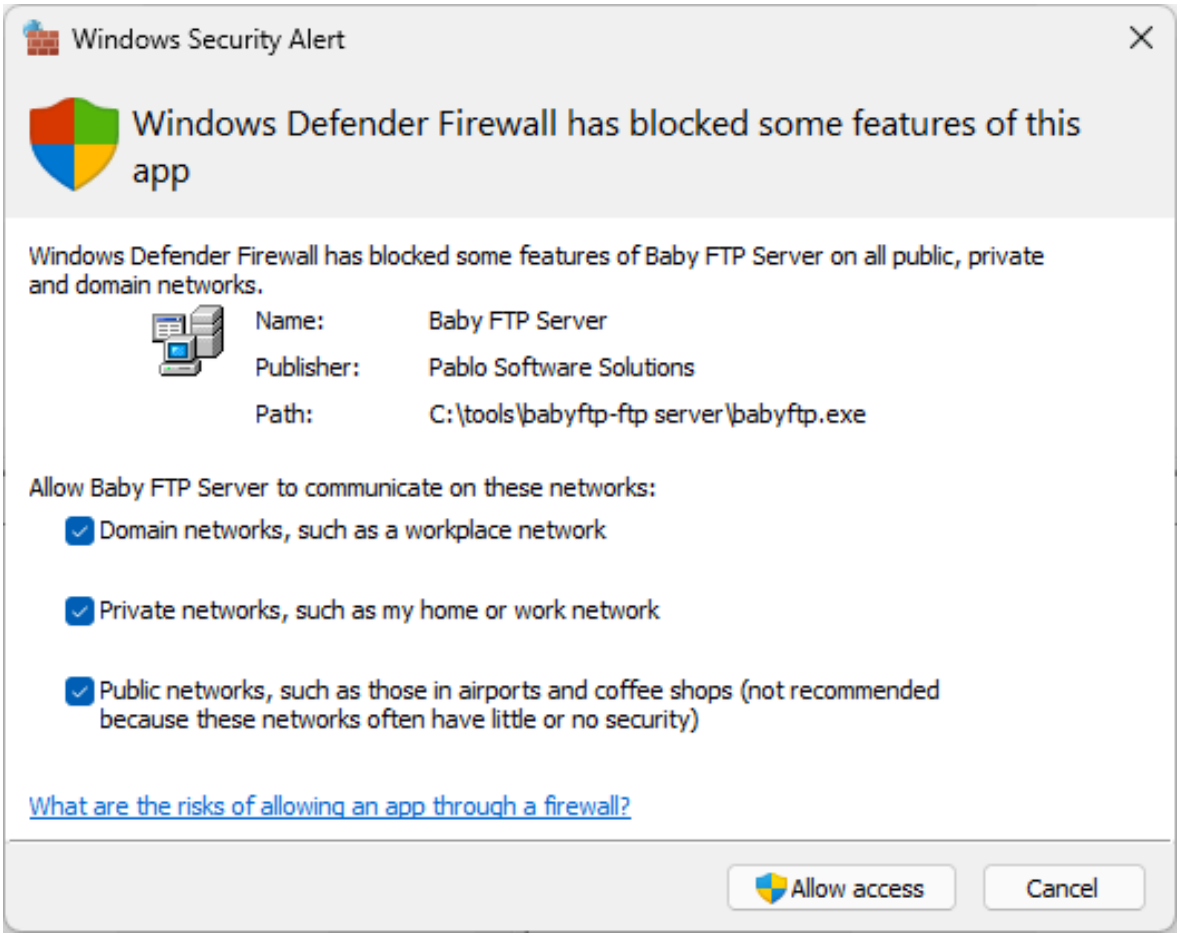


Testing the FTP Connection

Before using the smart camera, test the FTP connection using an FTP client.

This may also trigger the Windows Defender Firewall for the FTP server application. This operation can be done automatically.

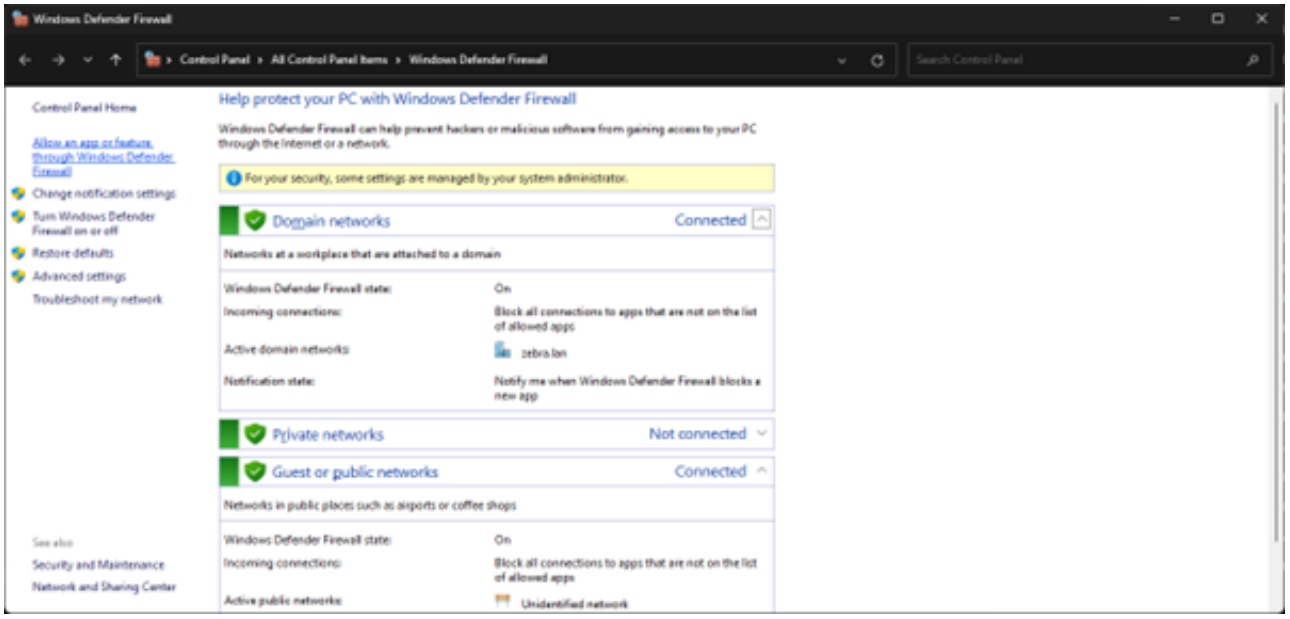
Enable the FTP server for all networks.



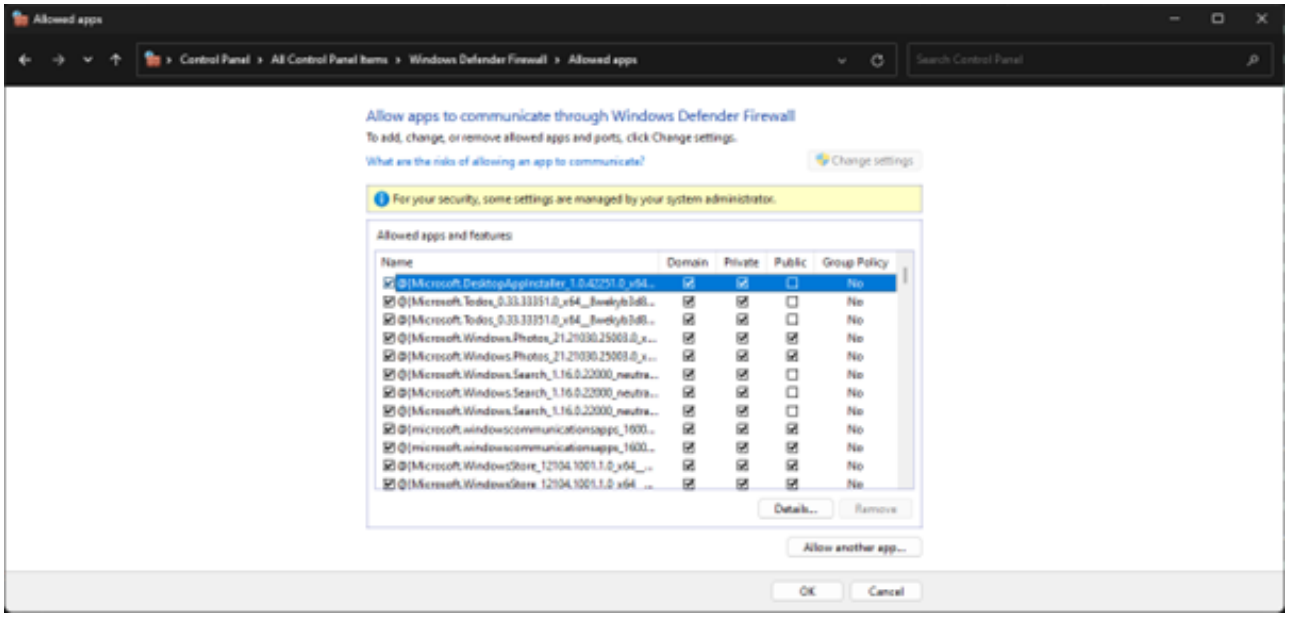
Configuring the Firewall

If there is no communication between the smart camera and the FTP Server application, add the FTP server application to the Windows Firewall Rules using the following procedure.

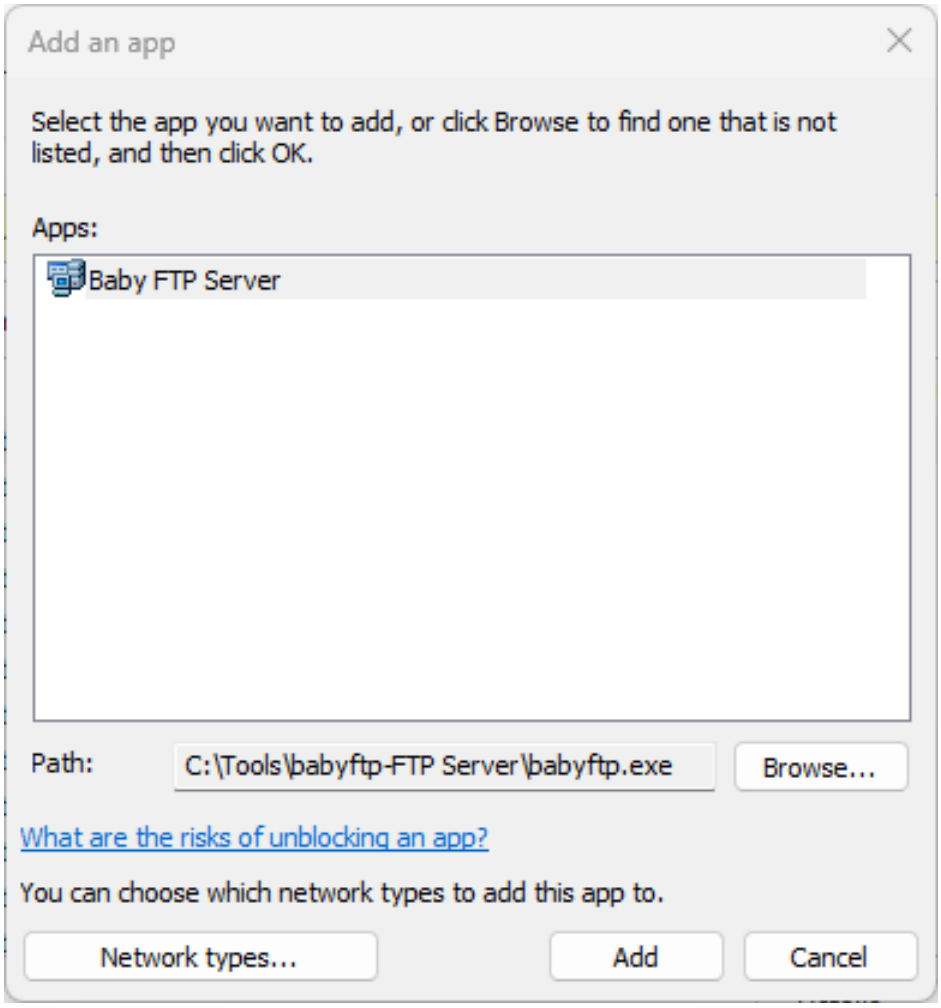
1. Open Windows Defender Firewall.
2. Click **Allow an app or feature ....**



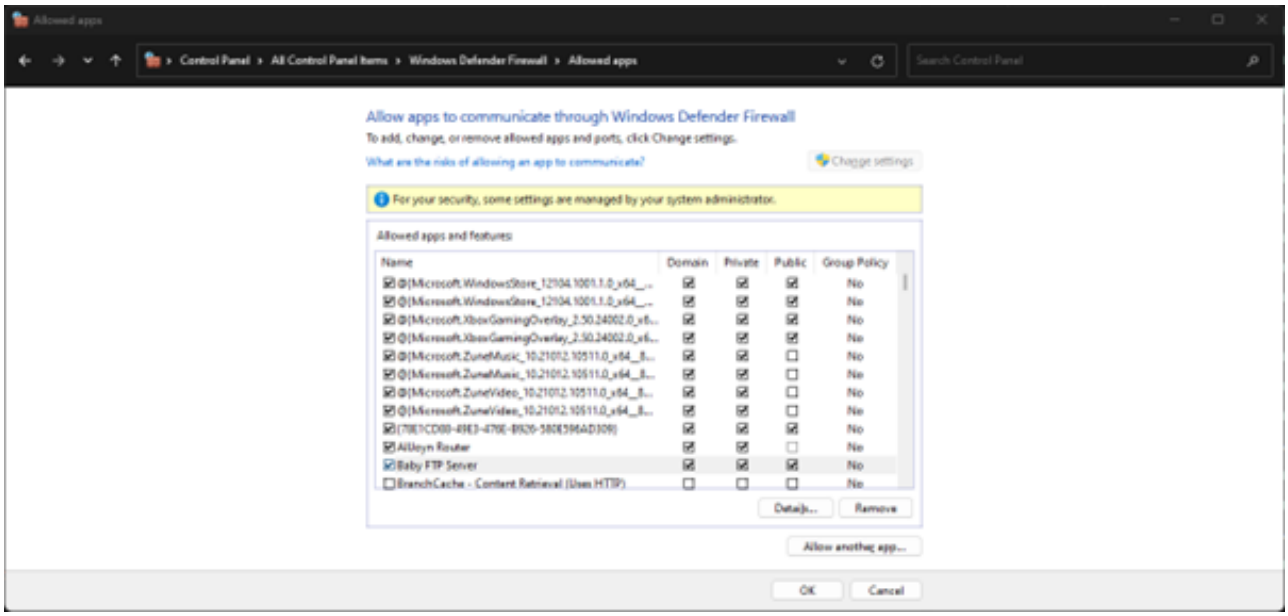
3. Click **Allow another app....**



4. Select the executable of the FTP server.



5. Select all networks.



6. Click **OK** to apply the settings.

RegEx Overview

RegEx is used to filter code to identify specific objects. This section provides details on what to look for in your code to confirm that RegEx is functioning properly, what data is constant, and what data can change. If you intend to change the data, you must provide the data type or reduce the data to a specific number of characters when possible.

For a full list of filtering commands to identify various data types, characters, and fixed positions or serial use of specific data, refer to rexegg.com/regex-quickstart.html

RegEx Examples

RegEx refers to a regular expression sequence of characters that specifies a pattern for the application to identify in the image.

This section provides examples of common RegEx use cases and outlines the procedure to utilize them in Aurora Focus.

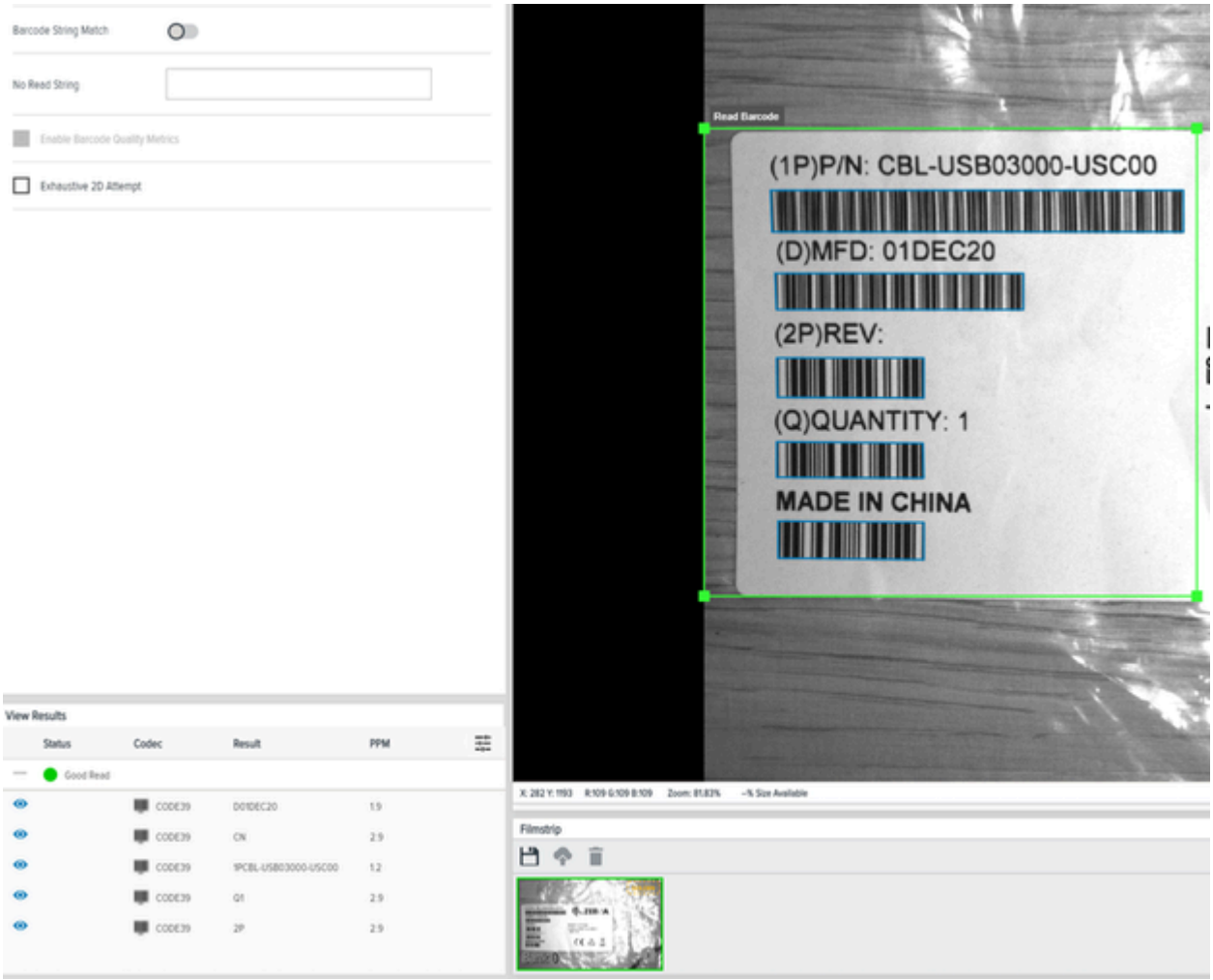
For a full list of filtering commands to identify various data types, characters, and fixed positions or serial use of specific data, refer to rexegg.com/regex-quickstart.html.

NOTE: Ensure that Manycode and Expand String Match to Entire Payload are enabled in settings on the Manycode tab. You have to start every filter with "\" and then add what you are looking for.

No Filter

All five codes were read without a filter:

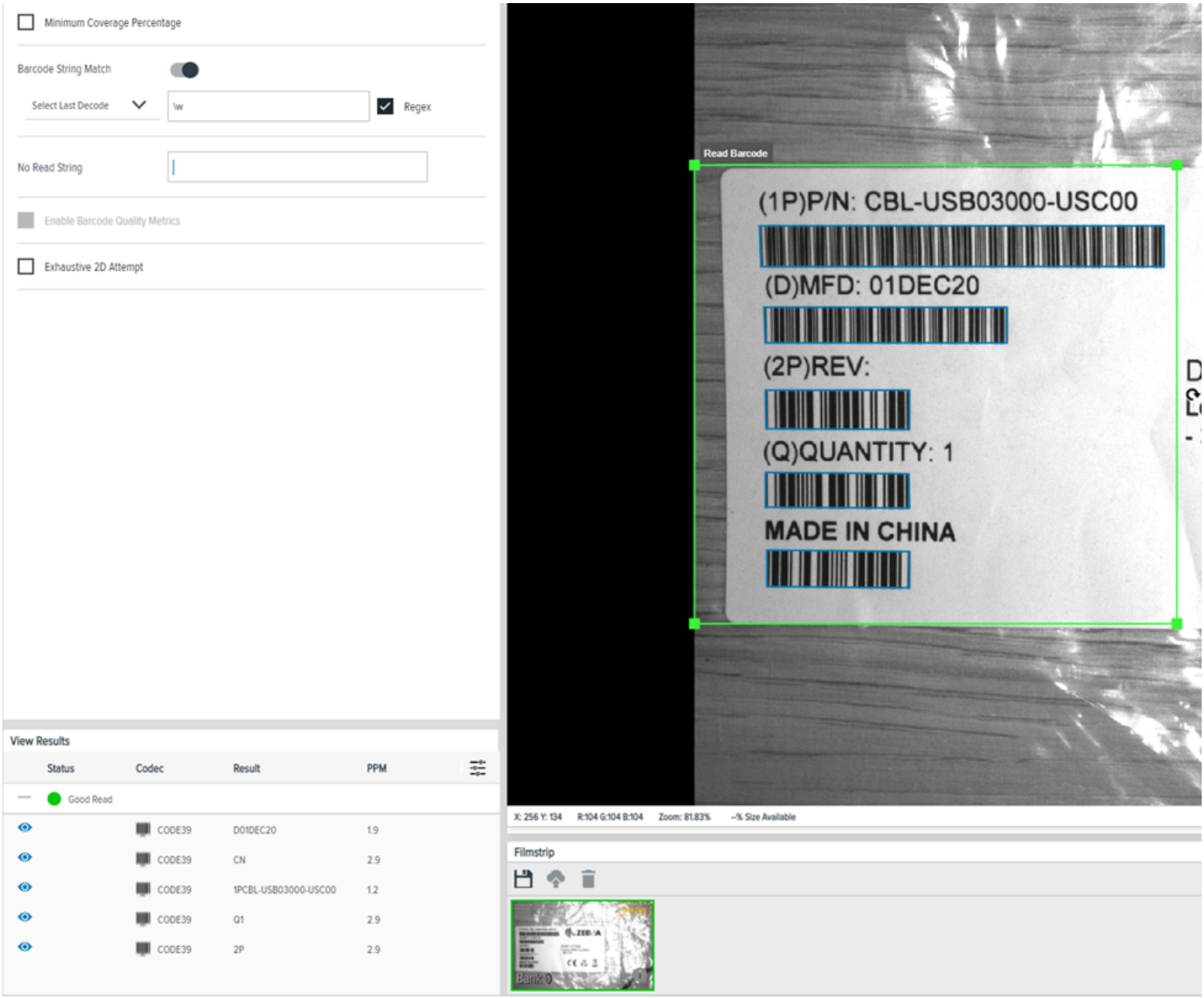
Figure 17 No Filter



Filter (\w) for all strings

Filter (\w) for all strings that include matches any word character (equivalent to [a-zA-Z0-9_]):

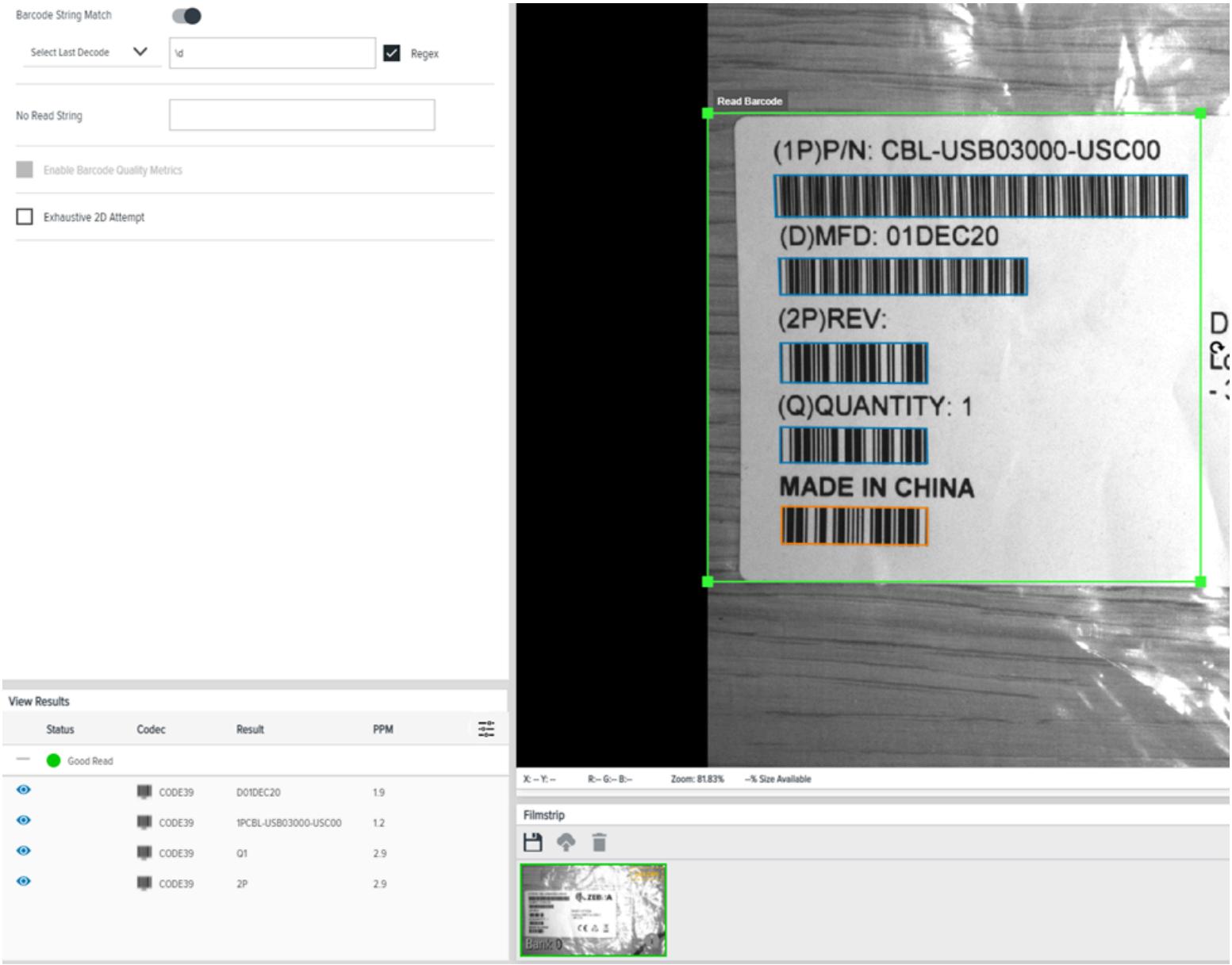
Figure 18 Filter (\w) for all strings that include matches any word character (equivalent to [a-zA-Z0-9_]):



Filter (\d) for all images

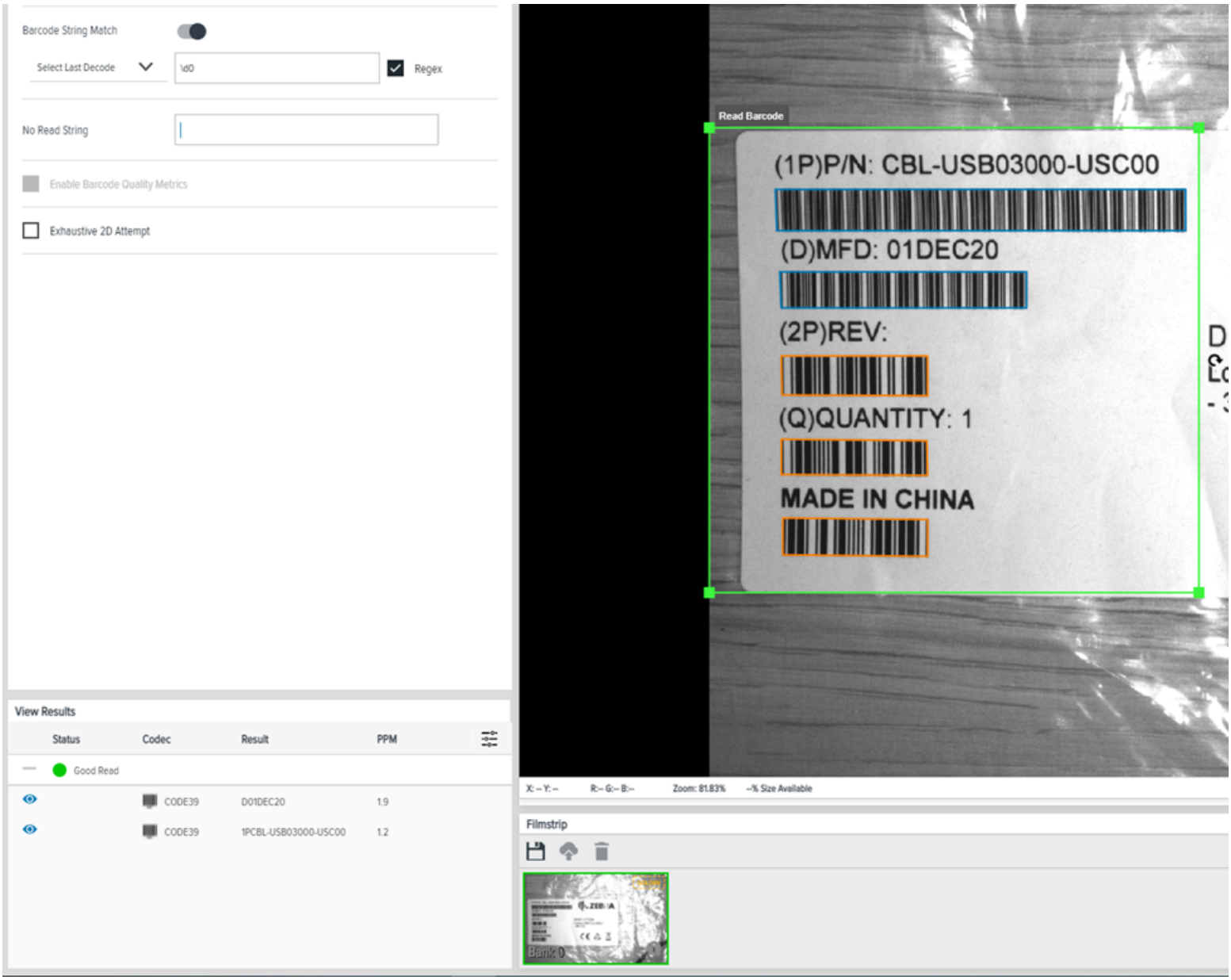
Filter (\d) for all images with at least one number:

Figure 19 Filter (\d) for all images with at least one number:



Filter (\d0) to all numbers

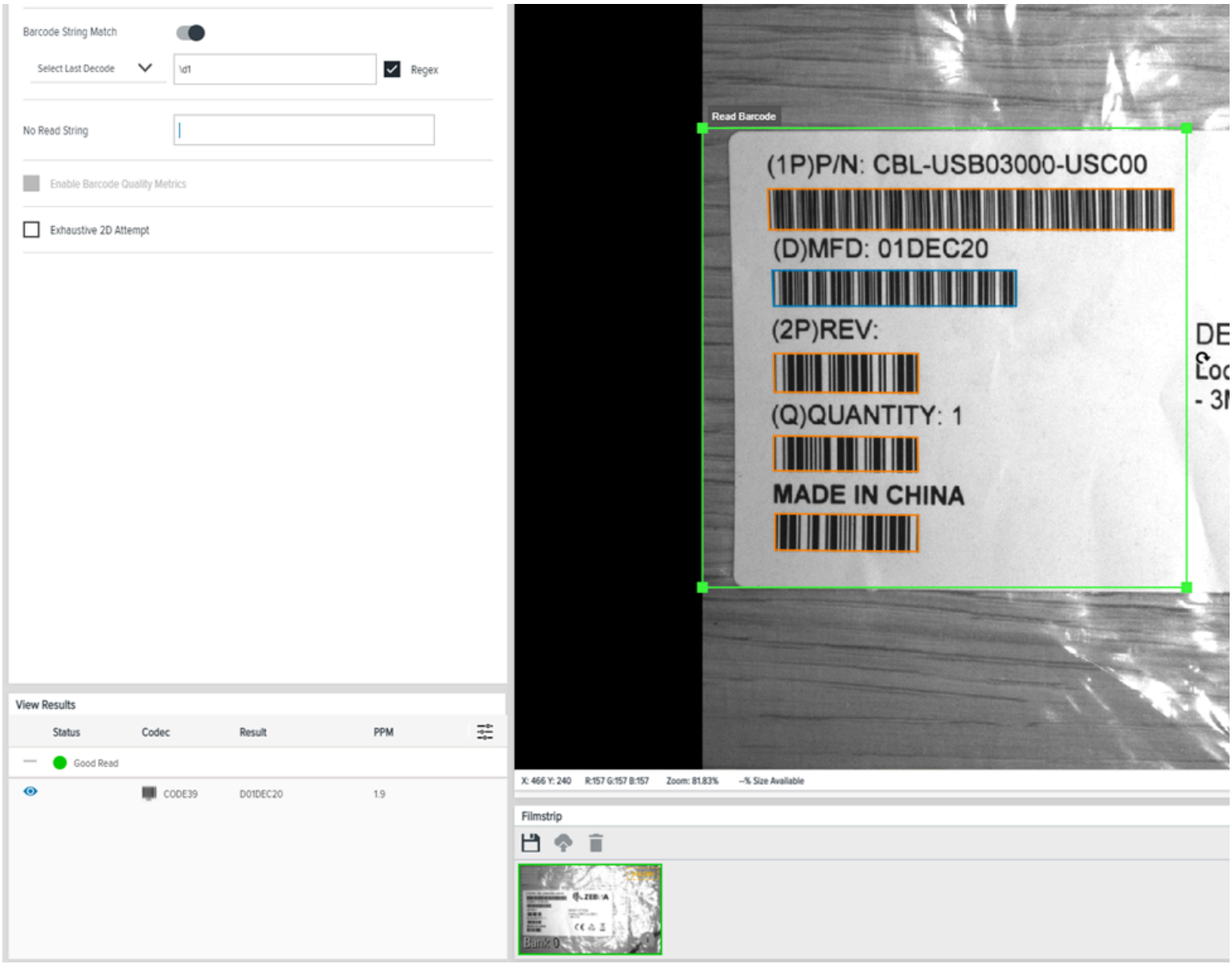
Filter (\d0) to all numbers that are followed by zero:



Filter (\d1) to all numbers

Filter (\d1) to all numbers followed by the number one:

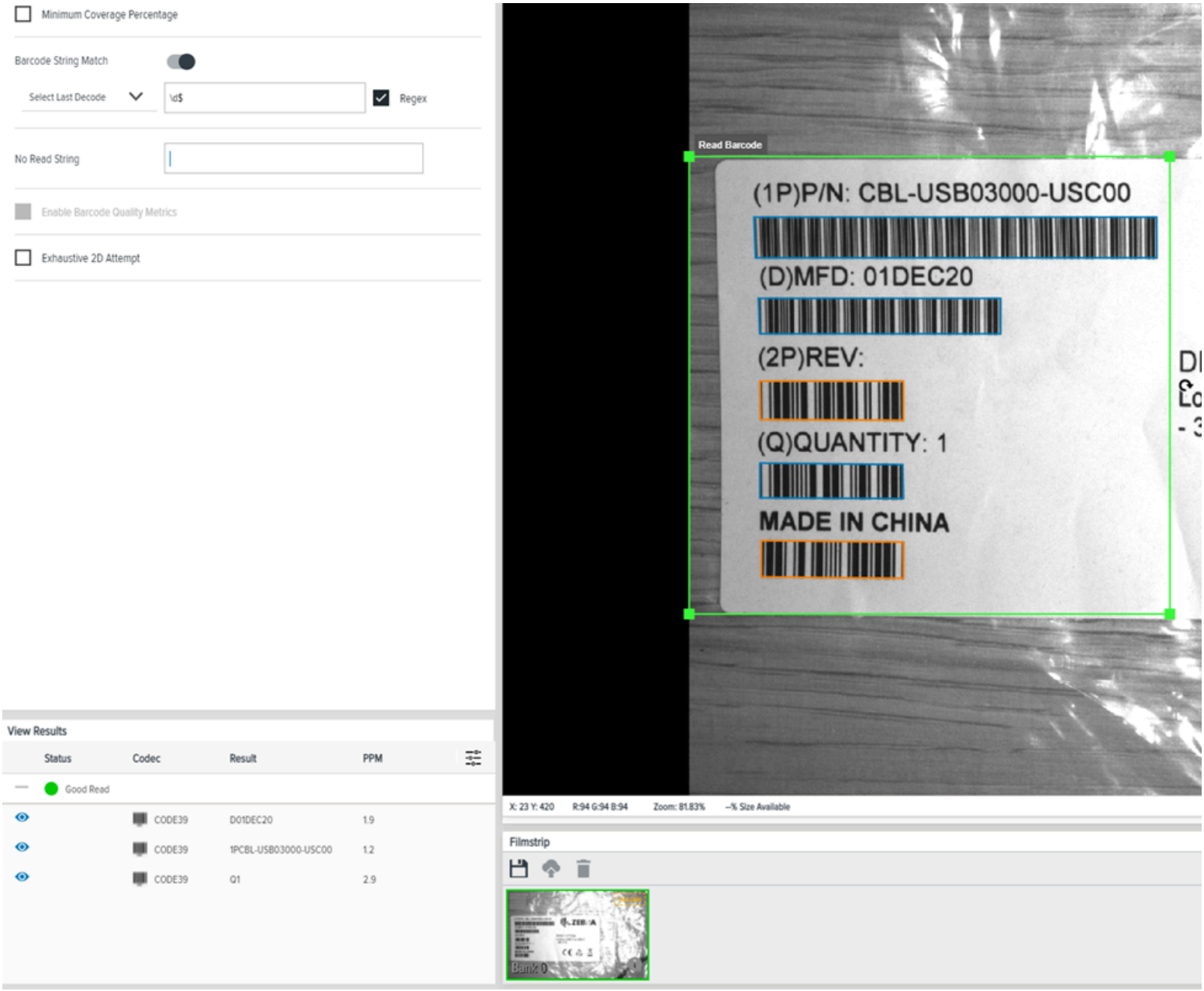
Figure 20 Filter (\d1) to all numbers followed by the number one:



Filter (\d$) to every code

Filter (\d$) to every code with a number as its last digit:

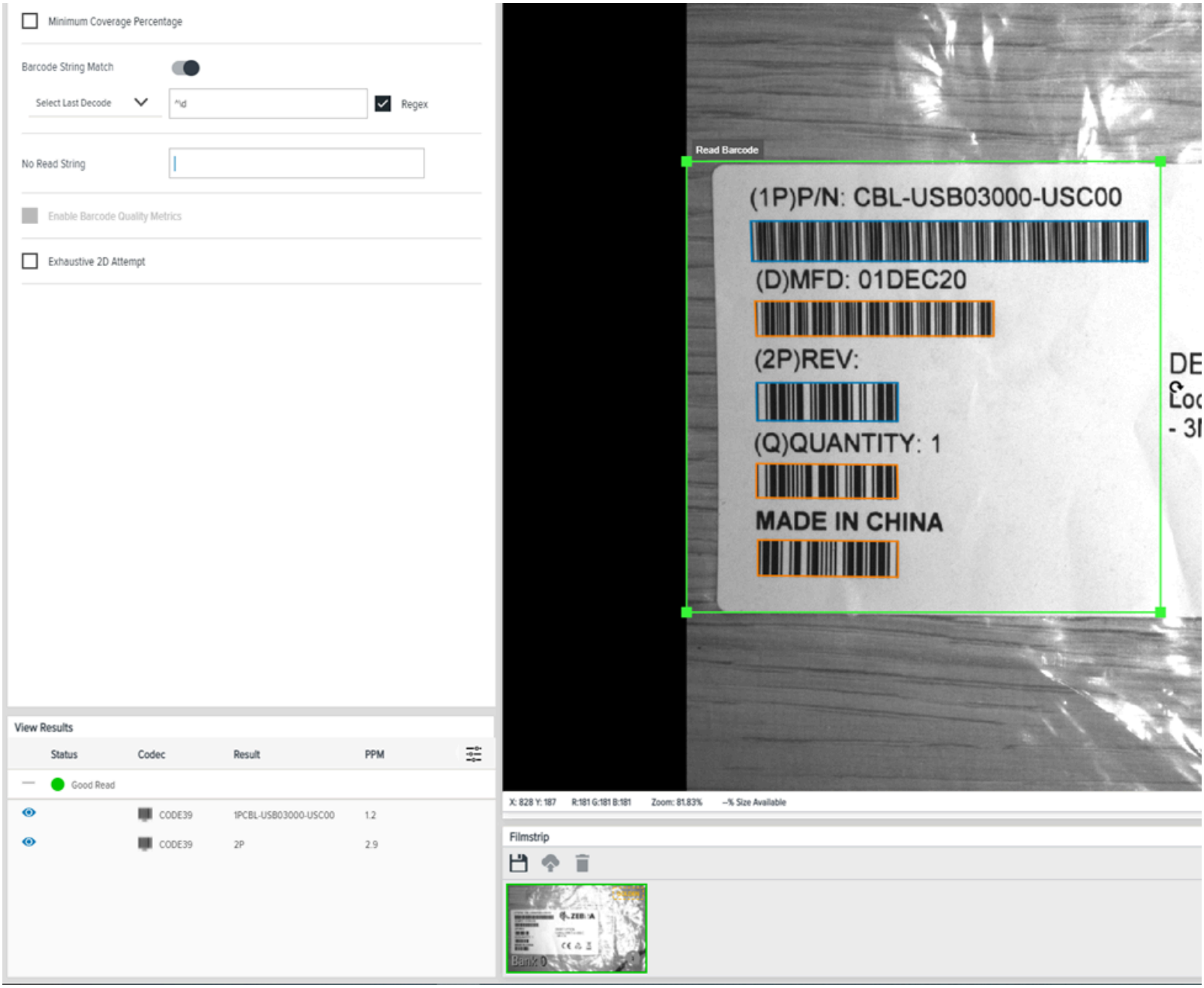
Figure 21 Filter (\d$) to every code with a number as its last digit:



Filter (^\d) to get any code

Filter (^\d) to get any code with a number at the beginning:

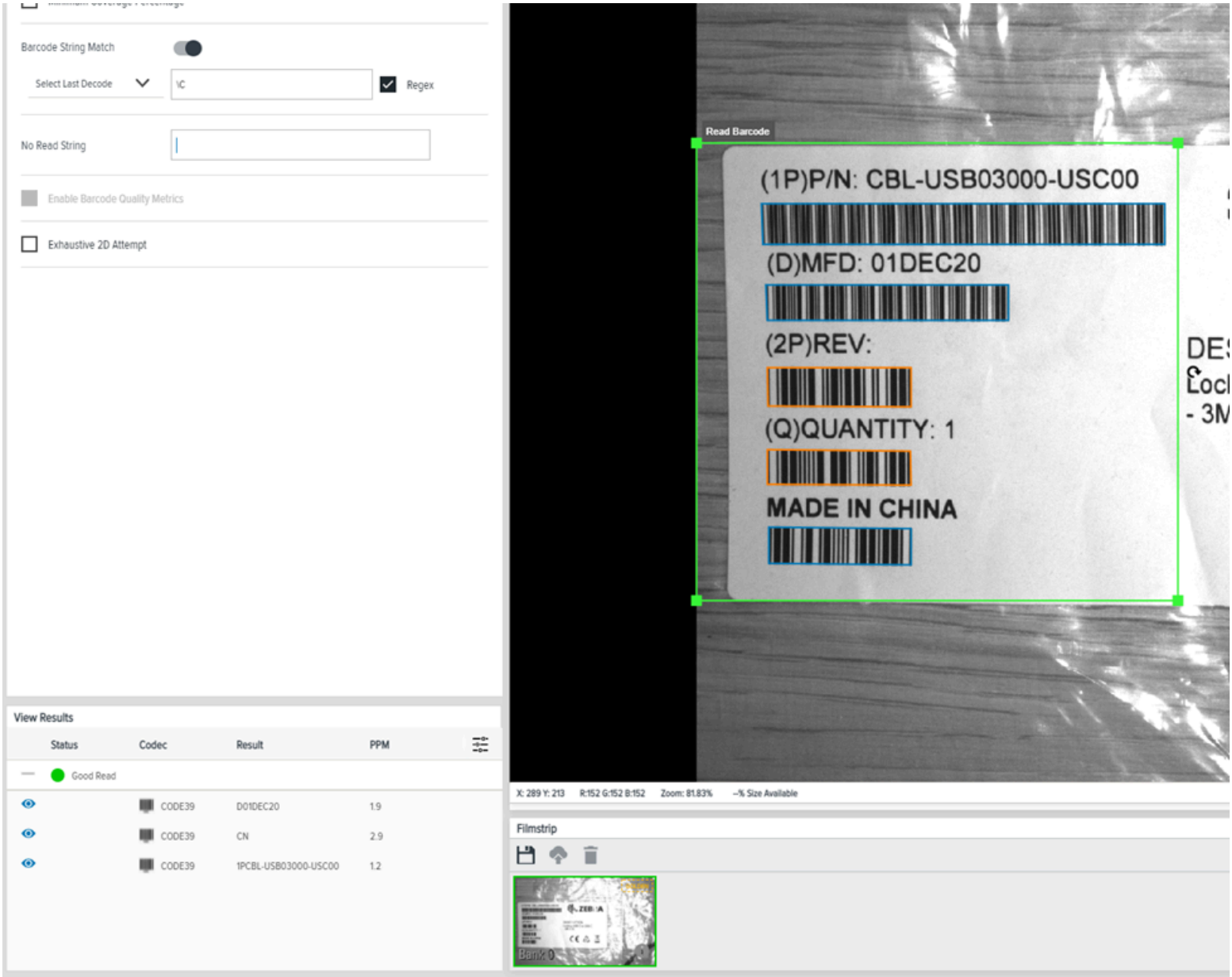
Figure 22 Filter (^\d) to get any code with a number at the beginning:



Filter (\C) for a string

Filter (\C) for a string that has the letter C:

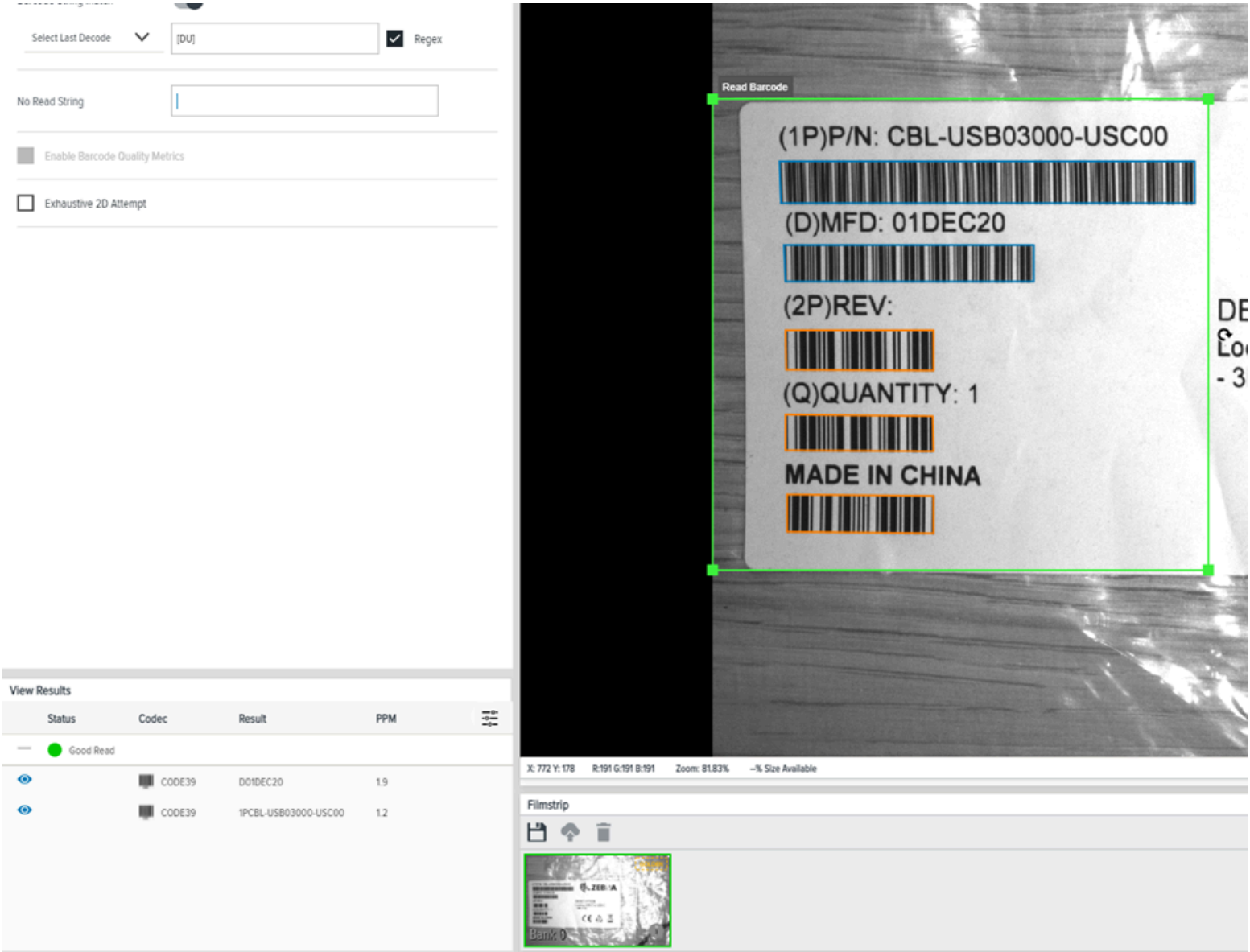
Figure 23 Filter (\C) for a string that has the letter C:



Filter [DU] by a list

Filter [DU] by a list. This example is the letter D or U:

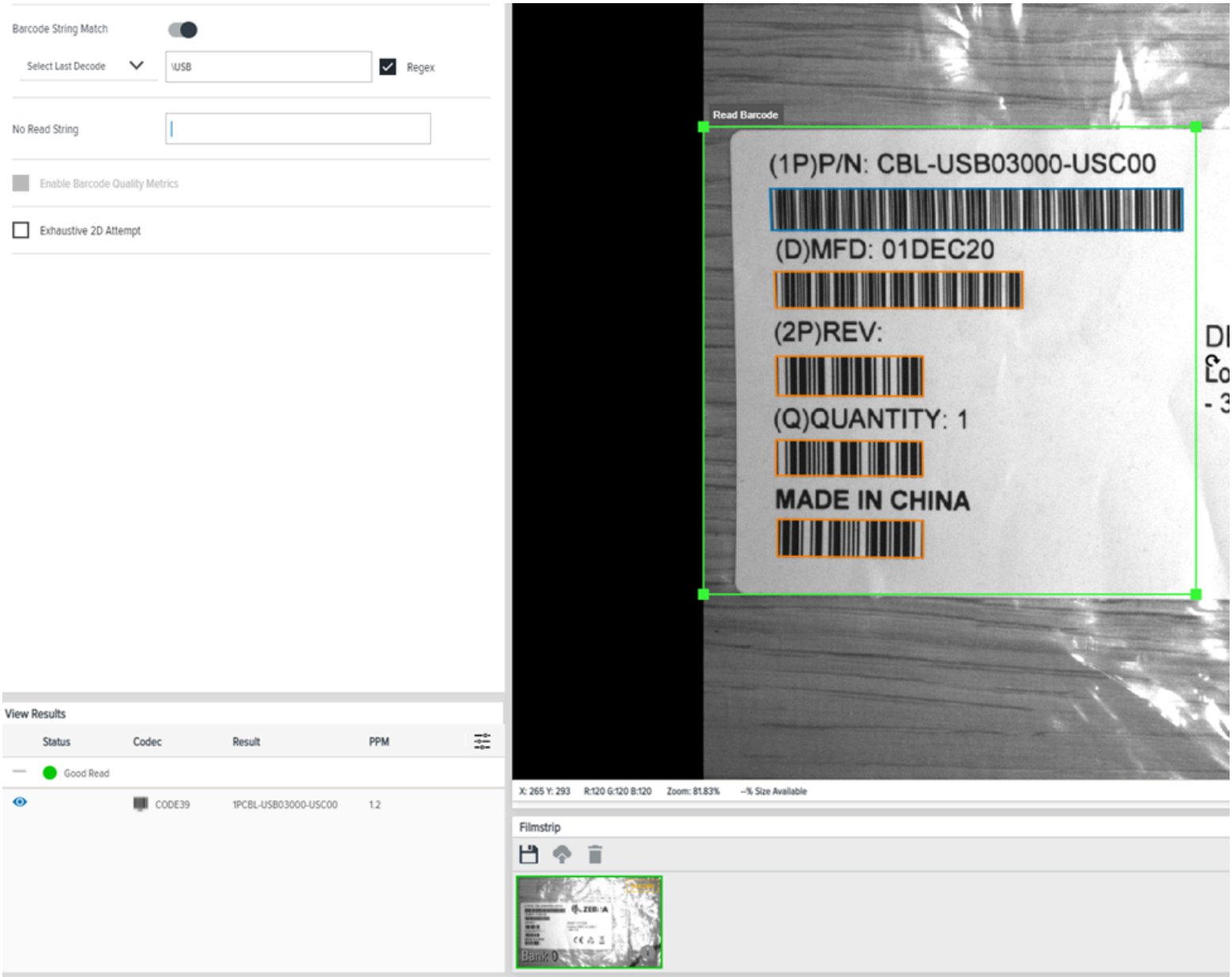
Figure 24 Filter [DU] by a list. This example is the letter D or U:



Look for a specific string in the code

Look for a specific string in the code (\USB):

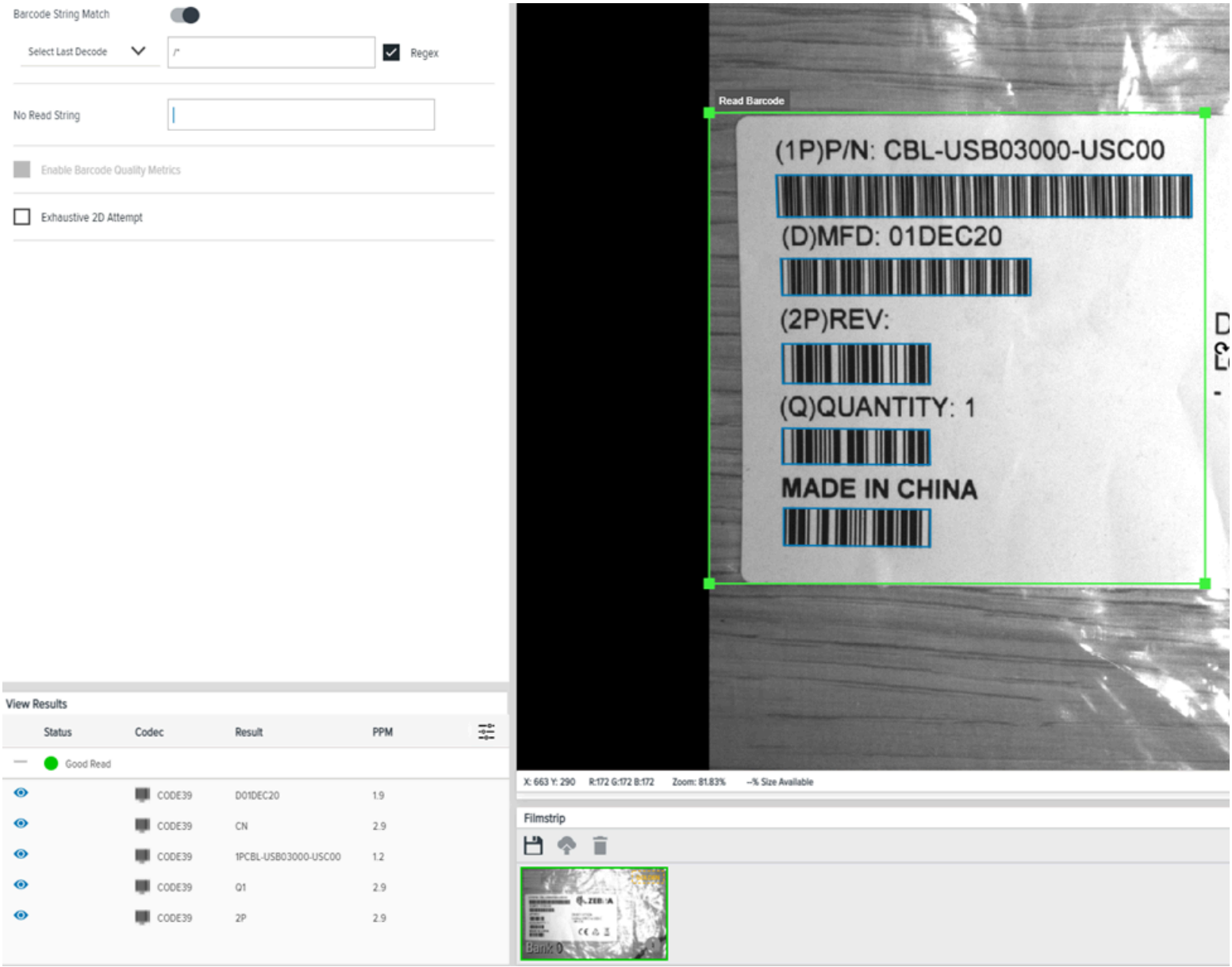
Figure 25 Look for a specific string in the code (\USB):



Look for any data filter

Look for any data filter (*):

Figure 26 Look for any data filter



Look for two specific prefix data filters

Look for two specific prefix data filters (^78|^S2):

- ^ sets the anchor to the first character or string in the code
- | is the logical OR

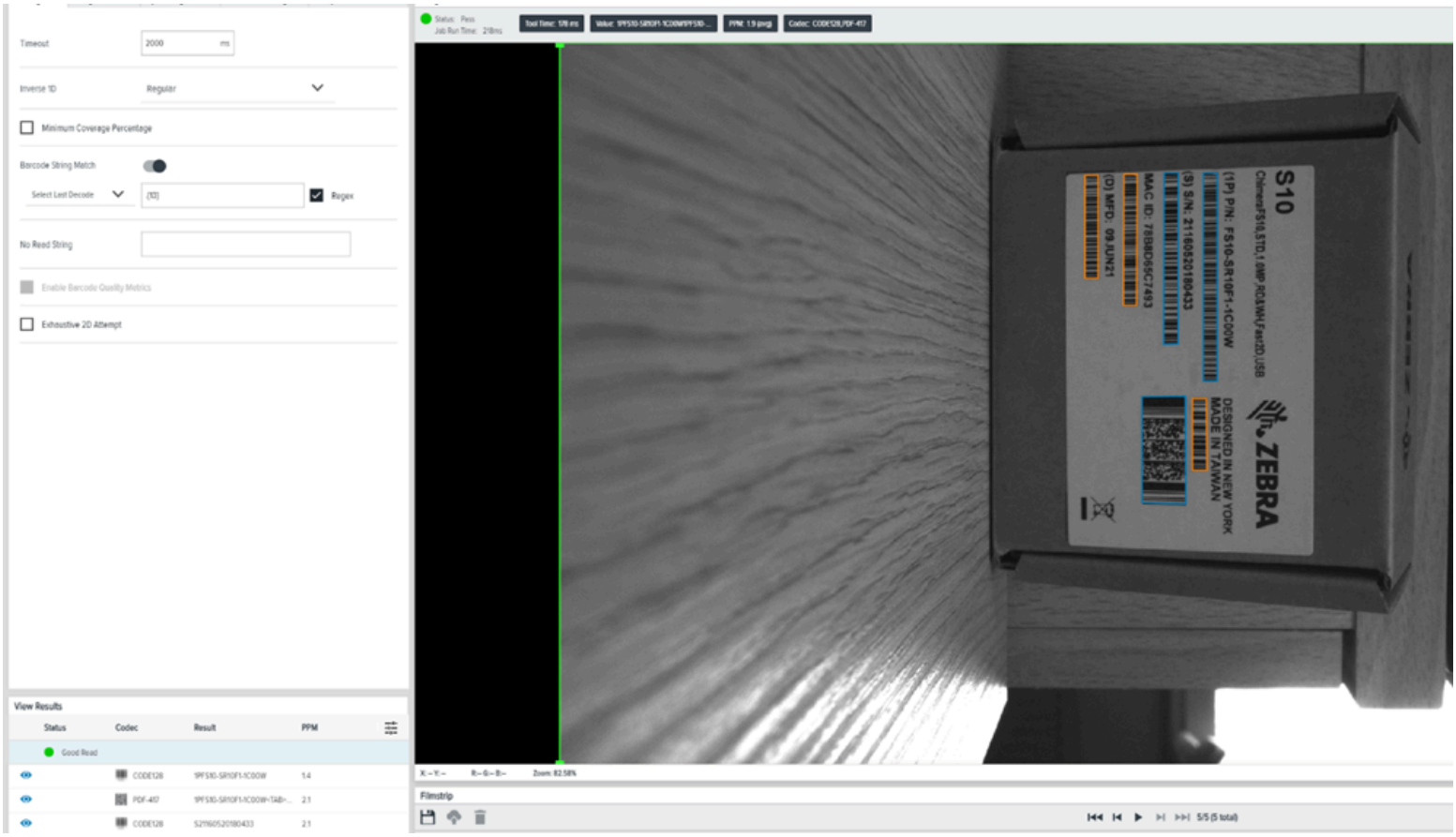
Figure 27 Look for two specific prefix data filters (^78|^S2):

Look for a minimum code length

Look for a minimum code length (.{11}):

- **.** allows any sign
- **{11}** is the number of signs needed for the result to be true. All orange codes are shorter than 11 signs.

Figure 28 Look for a minimum code length (.{11}):



Look for a code length range

Look for a code length range (^.{3,12}$):

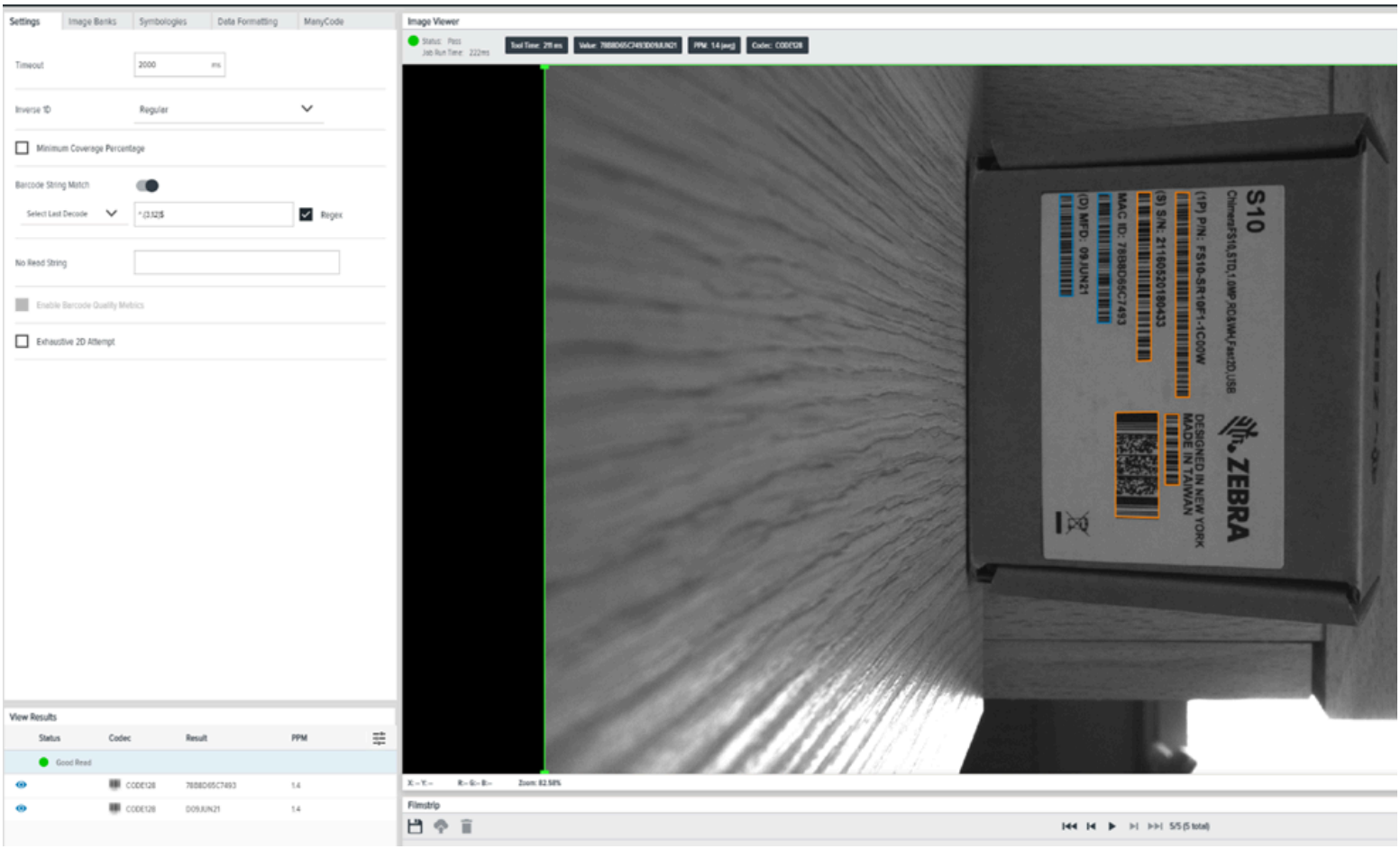
- “^” anchor at the start of the code.
- “$” anchor at the end of the code.

Specify the desired output by providing the information inside the brackets:

- **.** allows any sign.
- **{3,12}** the first number is the minimum number of signs needed to be true, and the second number represents the maximum.

All codes with two or fewer signs are ignored, such as the TW on top of the PDF417 code. Codes with a length of 13 or more signs are also ignored.

Figure 29 Look for a code length range (^.{3,12}$):



Look for a code length range and an identifier

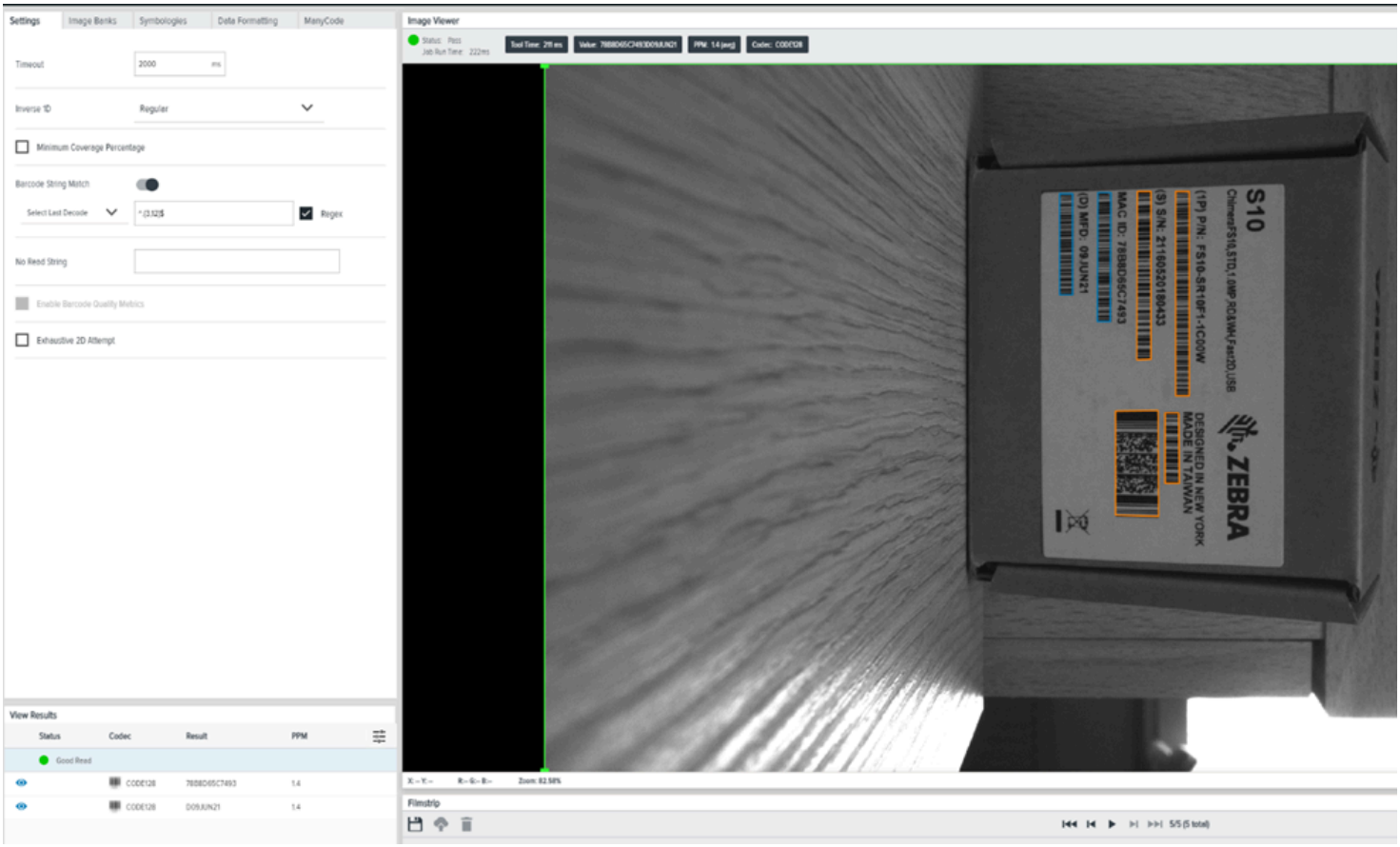
Look for a code length range and an identifier (^78.{3,12}$:

- **^** anchor at the start of the code.
- **$** anchor at the end of the code.

Specify the desired output by providing the information inside the brackets:

- **.** allows any sign.
- **$78** is the identifier needs to be at the beginning of the string.
- **{3,12}** the first number is the minimum number of signs needed to be true, and the second number represents the maximum.

All codes with two or fewer signs are ignored, such as the TW on top of the PDF417 code. Codes with a length of 13 or more signs are ignored as well.



Look for any sign, then look for a code length range and an identifier

Look for anything else, then look for a code length range and an identifier (^(?!^78.{3,12}$).) with inverse logic:

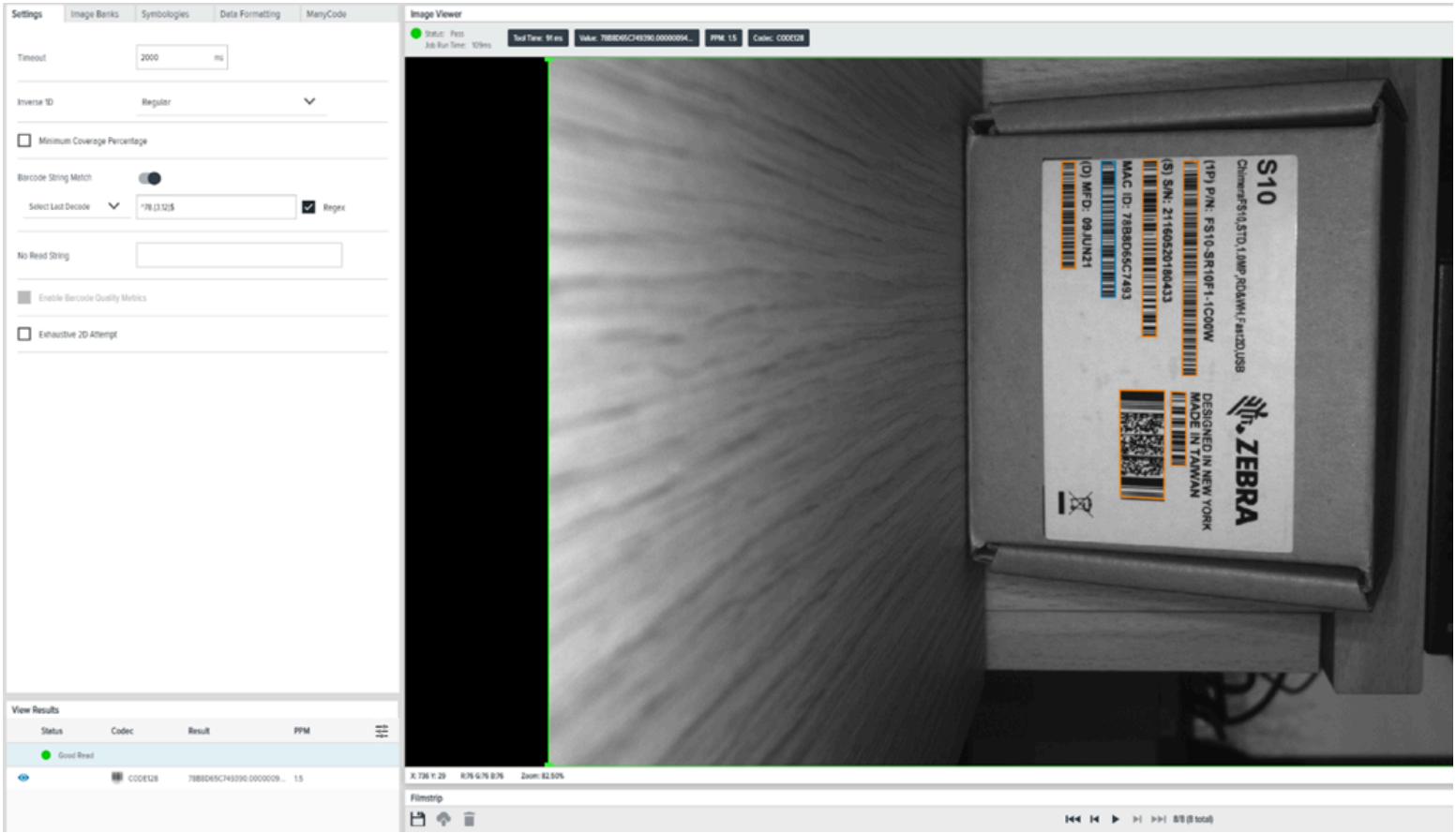
- **^** allows any sign.
- **$** is the identifier needs to be at the beginning of the string.

Specify the desired output by providing the information inside the brackets.

- **.** allows any sign.
- **78** is the identifier needs to be at the beginning of the string.
- **{3,12}** the first number is the minimum number of signs needed to be true, and the second number represents the maximum.

Use the syntax ^(?!pattern). where the pattern is the pattern for negative pattern matching:

Figure 30 Look for any sign then look for a code length range and an identifier with inverse logic

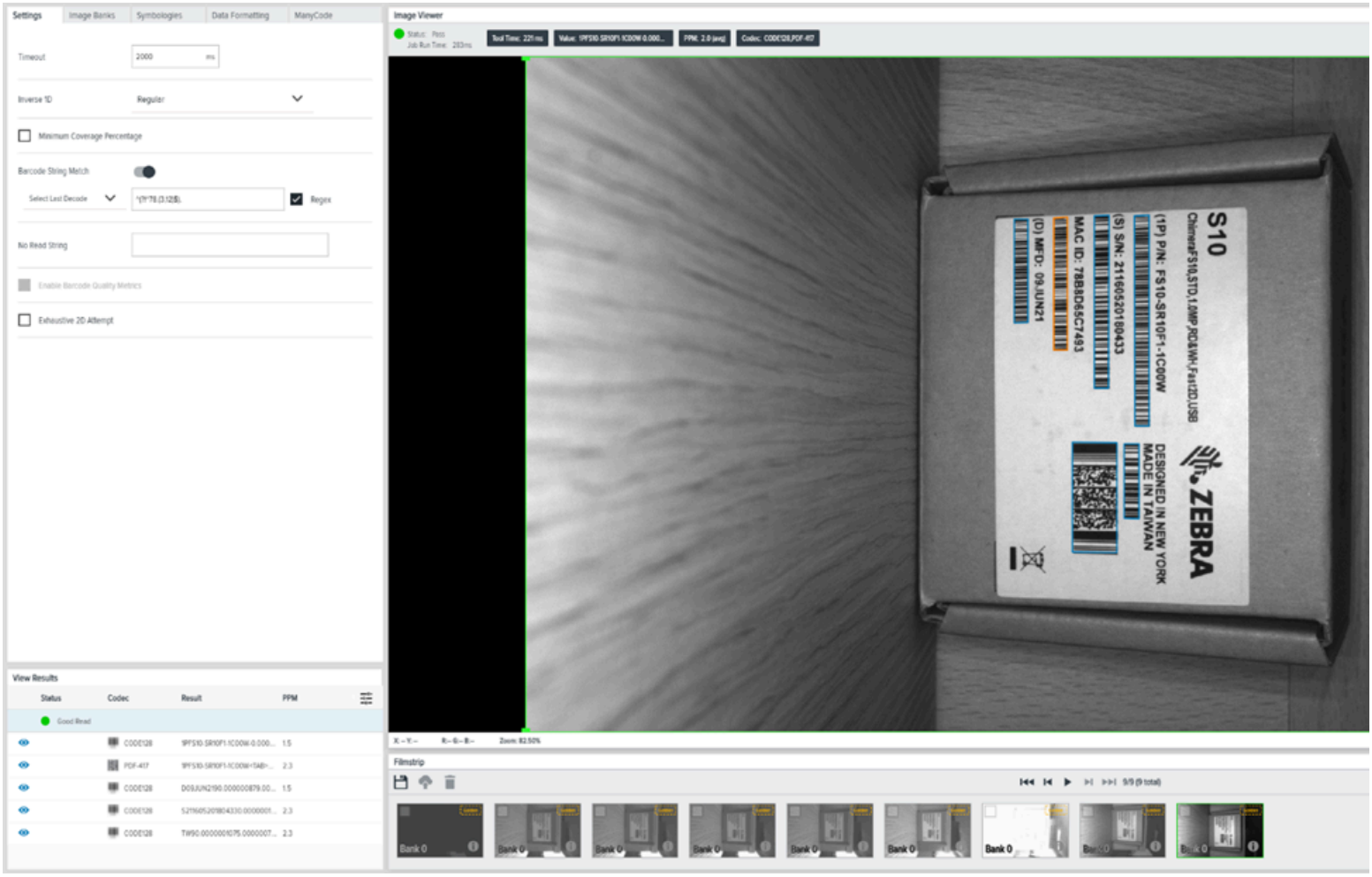


Look for a numeric code with a length of 13 that starts with a four or a numeric code with a length of 20

Look for a numeric code with a length of 13 and starts with a four or a numeric code with a length of 20 that starts with a 0 (^4\d{12}|^0\d{19}):

- ^ anchor at the start of code ^4 means the specific number 4 needs to be the first number in the code.
- **\d** allows numbers only (0-9).
- **{12}** number of signs needed to be true, and the second number represents the maximum. It's one less than the code length because the full string consists of the fixed first number + 12 numbers.
- **|** is the logical OR.

Figure 31 Look for a numeric code with a length of 13 and starts with a four

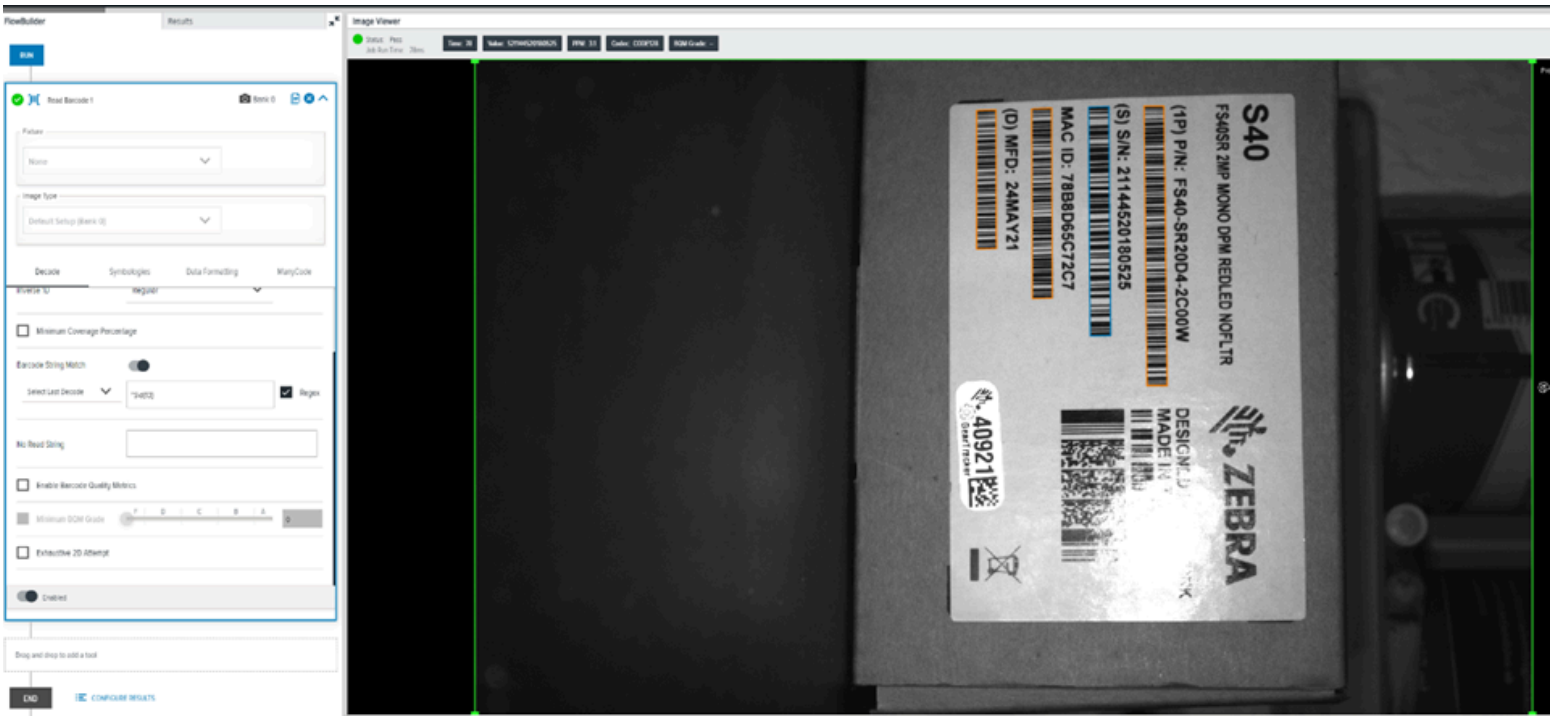


Find the serial number field

Find the serial number field of the FIS/MV Zebra Boxes (^S\d{13}):

- **^**
- **\d** allows numbers only (0-9).
- **{13}**

Figure 32 Find the serial number field of the FIS/MV Zebra Boxes



Using Zebra Easy Text Interface

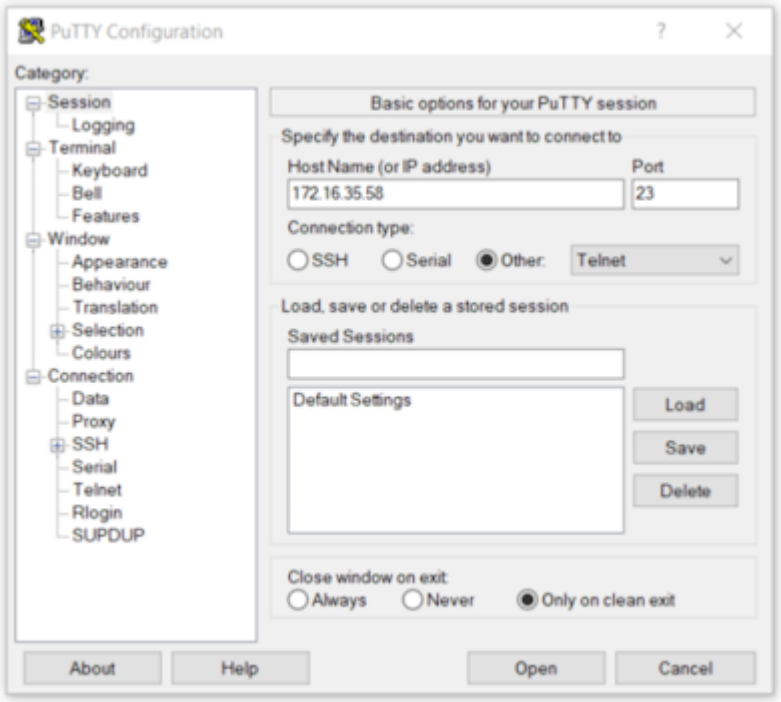
Zebra Easy Text Interface (ZETI) is a set of commands used to retrieve information from the device over telnet Port 23 by default.

Enabling Telnet Connections

Use a Telnet connection with a PC-based terminal over ZETI with a device.

1. Press Win + R to open Run.
2. Search for the Control Panel and click **OK**.
3. Search for the Control Panel and click **Programs and Features**Using the left panel, click **Turn Windows Features On or Off**.
4. Enable telnet client Windows features dialog.
5. In the Windows Features dialog, scroll down and select **Telnet Client**.

You can also use a terminal client such as Teraterm or Putty. The following image displays the procedure using Putty.



After opening the session, type the command name and press **Enter**.

ZETI Commands

Use ZETI commands to make changes on the device and retrieve result data.

autotune

Use the `autotune` command to adjust the focus settings of the device based on the specified parameter.

Table 80 autotune

Command	Short	Parameter	Data	Range	Example

`autotune`	at	`.brightmess`
		`.focus`
		`.method`
			true/false
			true/false
			barcode/ dpm
				-	autotune .brightness true .focus true .method barcode
					command: autotune,ok

`autotune`	at	`.brightmess`
		`.focus`
		`.method`
			true/false
			true/false
			barcode/ dpm
				-	autotune .brightness false .focus true .method barcode
					command: autotune,ok

`autotune`	at	`.brightmess`
		`.focus`
		`.method`
			true/false
			true/false
			barcode/ dpm
				-	autotune .brightness true .focus false .method barcode
					command: autotune,ok

`autotune`	at	`.brightmess`
		`.focus`
		`.method`
			true/false
			true/false
			barcode/ dpm
				-	autotune .brightness false .focus false .method barcode
					command: autotune,ok

`autotune`	at	`.brightmess`
		`.focus`
		`.method`
			true/false
			true/false
			barcode/ dpm
				-	autotune .brightness true .focus true .method dpm
					command: autotune,ok

`autotune`	at	`.brightmess`
		`.focus`
		`.method`
			true/false
			true/false
			barcode/ dpm
				-	autotune .brightness true .focus false .method dpm
					command: autotune,ok

`autotune`	at	`.brightmess`
		`.focus`
		`.method`
			true/false
			true/false
			barcode/ dpm
				-	autotune .brightness false .focus true .method dpm
					command: autotune,ok

Table 80 autotune (Continued)

Command	Short	Parameter	Data	Range	Example
autotune	at	.brightmess .focus .method	true/false true/false barcode/ dpm	-	autotune .brightness false .focus false .method dpm command: autotune,ok

backuprestore

backup

Use the backuprestore command to backup a file by passing a backup parameter as CONFIG and the action as 1.

Table 81 backup

Command	Short	Parameter	Data	Range	Example
backuprestore	br	backup	CONFIG, JOB_DATA, ALL	-	br .backup CONFIG .action 1 command:br,ok
backuprestore	br	action	1		br .backup CONFIG .action 1 command:br,ok

restore

Use the backuprestore command to restore a file to the device.

Table 82 restore

Command	Short	Parameter	Data	Range	Example
backuprestore	br	action. path	0 base64_data	-	br .action 0 .path base64_data command:br,ok

NOTE: Use a Python script to restore the file to the device.

deletejob

Use the deletejob command to delete a job on the device. Provide the job number as an argument.

Table 83 deletejob

Command	Short	Parameter	Data	Range	Example
deletejob	dj	-	Number	-	deletejob 1 command:deletejob,ok

dwload

Use the `dwload` command to upload the dwx configuration file to the host.

NOTE: Use a python script to transfer the .dwx file to the device.

Table 84 dwload

Command	Short	Parameter	Range	Example
`dwload`	dwload	-	-	dwsideload base64_data command: dwsideload,ok

dwsideload

Use the `dwsideload` command to upload the dwx configuration file to the device.

NOTE: Use a python script to transfer the .dwx file to the device.

Table 85 dwsideload

Command	Short	Parameter	Range	Example
`dwsideload`	dws	`file.dwx (base64 format)`	-	python dwsideload file.dwx <CR><LF> command: dwsideload,ok

exposure

Use the `exposure` command to set the exposure value by passing an argument.

Table 86 exposure

Command	Short	Parameter	Range	Example
`exposure`	ex	-	0.05 to 14	exposure 1 command: exposure,ok

factoryreset

Use the `factoryreset` command to return the device to its default state.

Table 87 factoryreset

Command	Short	Parameter	Data	Range	Example
`factoryreset`	fr	`device_settings`	-	-	factoryreset device_settings command: factoryreset,ok

firmwareupdate

Use the `firmwareupdate` command to update the device firmware.

NOTE: The `firmwareupdate` command is not supported on FS10 devices.

Table 88 firmwareupdate

Command	Short	Parameter	Data	Range	Example
`firmwareupdate`	fwu	-	ftp server username ftp server password ftp server url firmware name (scnplg2) force update keepfile	-	fwu user pass 172.16.35.58 abc.scnplg2 1 1 command:fwu,ok

friendly_name

Use the friendly_name command to get or set the friendly name (a customizable label) of a tool in a job. These commands are intended for use with an FS job or a VS job with only one tool. Implementing multiple tools can impact the performance of the first tool. If the job slot is not explicitly stated, the currently deployed job is the default.

Example: getjobdata .job_run_data.graph.nodes.BARCODE.friendly_name .slot 2

The settings related to job_run_data.graph.nodes.BARCODE.friendly_name can be configured using the getjobdata / setjobdata (gjd / sjd) along with the sub parameter as shown in the following table:

Table 89 friendly_name

Command	Parameter	Sub Parameter	Range	Example
`getjobdata` `setjobdata`	msg	`tools`	string	gjd .jrf .slot 3 command: gjd,ok {response} sjd .jrf NewName .slot 3 command: sjd,ok

NOTE: {response} indicates the response for a command.

focus

Use the `focus` command to set the focus value by passing an argument.

Table 90 focus

Command	Short	Parameter	Range	Example
`focus`	fo	-	-6 to 8	focus 7 command: focus,ok

gain

Use the `gain` command to set the gain value by passing an argument.

Table 91 gain

Command	Short	Parameter	Range	Example
`gain`	ga	-	0 to 100	gain 2 command: gain,ok

getcodes

Use the `getcodes` command to retrieve the code and value of the last operation.

Table 92 getcodes

Command	Short	Parameter	Data	Range	Example
`getcodes`	gc	-	-	-	getcodes command: getcodes,ok

getdecodes

Use the `getdecodes` command to retrieve the decoded value of the last operation.

Table 93 getdecodes

Command	Short	Parameter	Data	Range	Example
`getdecodes`	gd	-	-	-	getdecodes command: getdecodes,ok

getimage

Use the `getimage` command to capture a new image that is not counted as a trigger and download the image as a base64 buffer. The buffer contains a BMP file.

Table 94 getimage

Command	Short	Parameter	Range	Example
`getimage`	gi	`Base64`	-	getimage command: getimage,ok 3073440 /9j/4AAQSkZJRgABAQAAAQABAAD

getimagersettings

Use the `getimagersettings` command to retrieve the value of all the parameters mentioned in the following table.

Table 95 getimagersettings

Command	Short	Parameter	Range	Example
`getimagersettings`	gis	`aimer`	-	getimagersettings aimer command: getimagersettings,ok
`getimagersettings`	gis	`external_light`	-	getimagersettings external_light command: getimagersettings,ok
`getimagersettings`	gis	`imager`	-	getimagersettings imager command: getimagersettings,ok
`getimagersettings`	gis	`internal_light`	-	getimagersettings Internal_light command: getimagersettings,ok
`getimagersettings`	gis	`gain`	-	getimagersettings gain command: getimagersettings,ok
`getimagersettings`	gis	`focus`	-	getimagersettings focus command: getimagersettings,ok
`getimagersettings`	gis	`exposure`	-	getimagersettings exposure command: getimagersettings,ok
`getimagersettings`	gis	`autoexposure`	-	getimagersettings autoexposure command: getimagersettings,ok
`getimagersettings`	gis	`autofocus`	-	getimagersettings autofocus command: getimagersettings,ok

getgpiostatus

Use the `getgpiostatus` command to get the value of the requested pin by passing a pin number as the argument.

Table 96 getgpiostatus

Command	Short	Parameter	Data	Range	Example
`getgpiostatus`	gst	`Number`	Pin number	-	getgpiostatus 1 command:getgpiostatus,ok {response}

getgpiosetting

Use the `getgpiosetting` command to update the device firmware.

Table 97 getgpiosetting

Command	Short	Parameter	Range	Example
`getgpiosetting`	ggs	`Pin number, mode`	-	ggs .pin 1 .mode command:ggs,ok {response}
`getgpiosetting`	ggs	`Pin number, strobe`	-	ggs .pin 1 .strobe command:ggs,ok {response}
`getgpiosetting`	ggs	`Pin number, reset`	-	ggs .pin 1 .reset command:ggs,ok {response}
`getgpiosetting`	ggs	`Pin number, edge`	-	ggs .pin 1 .edge command:ggs,ok {response}
`getgpiosetting`	ggs	`Pin number, debounce`	-	ggs .pin 1 .debounce command:ggs,ok {response}
`getgpiosetting`	ggs	`Pin number, delay`	-	ggs .pin 1 .delay command:ggs,ok {response}
`getgpiosetting`	ggs	`Pin number, pulsewidth`	-	ggs .pin 1 .pulsewidth command:ggs,ok {response}

getjoblist

Use the `getjoblist` command to retrieve the list of all jobs loaded on the device.

Table 98 getjoblist

Command	Short	Parameter	Data	Range	Example
`getjoblist`	gjl	-	-	-	getjoblist command: getjoblist,ok

getlogfiles

Use the `getlogfiles` command to retrieve device or perfetto logs.

Device Logs

Generate a log.tar and retrieve the tar file.

Table 99 Device Logs

Command	Short	Parameter	Range	Example
`getlogfiles`	glf	`devicelogs`	-	getlogfiles devicelogs command:getlogfiles,ok

Perfetto Logs

Use a python script to retrieve the latest Perfetto logs.

Table 100 Perfetto Logs

Command	Short	Parameter	Range	Example
`getlogfiles`	glf	`Number`	1-10	getlogfiles 10 command:getlogfiles,ok

getquantity

Use the `getquantity` command to retrieve the number of codes of the last operation.

Table 101 getquantity

Command	Short	Parameter	Data	Range	Example
`getquantity`	gq	-	-	-	getquantity command: getquantity,ok

getresultimage

Use the `getresultimage` command to download the last inspected image as a base64 buffer. The resulting image is a JPG file.

Table 102 getresultimage

Command	Short	Parameter	Data	Range	Example
`getresultimage`	gri	-	Base64	-	getresultimage command: getresultimage,ok 90326 /9j/4AAQSkZJRgABAQAAAQABAAD

help

Use the `help` command to return all supported ZETI commands.

Table 103 help

Command	Short	Parameter	Data	Range	Example
`help`	he	`None`	-	-	help Supported Commands trigger getimage getquantity ... list of all supported commands

internallight

Use the `internallight` command to enable or disable the internal light feature by passing On or Off as an argument.

Table 104 internallight

Command	Short	Parameter	Data	Range	Example
`internallight`	il	-	on/off	-	internallight on command: internallight,ok

match_string_get

Use the match_string_get command to retrieve the match string value of a tool within a specified job.

Table 105 match_string_get

Command	Short	Parameter	Range	Example
match_string_get	msg	tools	string within single quotes	msg .tool 'barcode' command:msg,ok
match_string_get	msg	format	PLAIN BASE64	msg .format PLAIN .tool 'barcode' command:msg,ok
match_string_get	msg		-	example using all parameters: match_string_get .format PLAIN .tool 'barcode command:msg,ok

match_string_update

Use match_string_update to update the match string value of a tool within a job.

NOTE: Value is a required parameter, all other parameters are optional. If the tool value is not specified, the update is applied to all tools.

Table 106 match_string_update

Command	Short	Parameter	Range	Example
match_string_update	.msu	value	string within single quotes	msu.value 'match_value'
match_string_update	.msu	tool	string within single quotes	msu.tool 'barcode' .value 'match_value' command:msu,ok
match_string_update	.msu	match_mode	STRING_EXACT STRING_CONTAINS REGEX DISABLED	msu .match_mode STRING_EXACT .value 'match_value' command:msu,ok
match_string_update	.msu	persistent	true/false	msu .persistent true .value 'match_value' command:msu,ok
match_string_update	.msu	format	PLAIN BASE64	msu.format PLAIN .value 'match_value' command:msu,ok

Table 106 match_string_update (Continued)

Command	Short	Parameter	Range	Example
`match_string_update`	.msu	-	-	example using all parameters: match_string_update .tool 'barcode' .match_mode STRING_EXACT .persistent true .format PLAIN . value 'string match' command:msu,ok

loadjob

Use the `loadjob` command to load the job, passing the slot number as an argument to ensure that the required job is loaded.

Table 107 loadjob

Command	Short	Parameter	Data	Range	Example
`loadjob`	lj	-	Number	-	loadjob command:loadjob,ok

protocolconfig

Use the `protocolconfig` command to set global parameters.

NOTE: Logs are not recorded on Telnet if .echo is off while using protocolconfig.

Table 108 protocolconfig

Command	Short	Parameter	Data	Range	Example
`protocolconfig`	pc	`.echo`	on/off	-	protocolconfig .echo off command:protocolconfig,ok

reboot

Use the `reboot` command to reboot the device.

Table 109 reboot

Command	Short	Parameter	Data	Range	Example
`reboot`	re	`None`	-	-	reboot command:reboot,ok

setgpiosetting

Use the `setgpiosetting` command to retrieve the attribute values of a pin.

Table 110 setgpiosetting

Command	Short	Parameter	Range	Example
`setgpiosetting`	sgs	`Pin number, mode, Number`	-	sgs .pin 1 .mode 0 command:sgs,ok
`setgpiosetting`	sgs	`Pin number, reset, Number`	-	sgs .pin 1 .strobe 0 command:sgs,ok
`setgpiosetting`	sgs	`Pin number, strobe, Number`	-	sgs .pin 1 .reset 0 command:sgs,ok
`setgpiosetting`	sgs	`Pin number, edge, Number`	-	sgs .pin 1 .edge 0 command:sgs,ok
`setgpiosetting`	sgs	`Pin number, debounce, Number`	-	sgs .pin 1 .debounce 0 command:sgs,ok
`setgpiosetting`	sgs	`Pin number, delay, Number`	-	sgs .pin 1 .delay 0 command:sgs,ok
`setgpiosetting`	sgs	`Pin number, pulsewidth, Number`	-	sgs .pin 1 .mode pulsewidth 0 command:sgs,ok

setgpiostatus

Use the `setgpiostatus` command to set the value for the required pin by passing a pin number as the first argument and the value as the second argument.

Table 111 setgpiostatus

Command	Short	Parameter	Data	Range	Example
`setgpiostatus`	sst	`Number, Number`	Pin number, value	-	setgpiostatus 1 0 command:setgpiostatus,ok

trigger

Use the trigger command to trigger a job on a specified slot by passing the required job number as an argument. The `withresult` parameter gives the resulting JSON of the last result as a response after the trigger.

Table 112 trigger

Command	Short	Parameter	Data	Range	Example
`trigger`	tr	`withresult`	Number	-	trigger command: trigger,ok trigger withresult command: trigger,ok

- getjobdata(gjd) – Get the data from specified job.
- setjobdata(sjd) – Set the data from specified job.

getjobdata .jobData.trigger.mode .slot 3

NOTE: Values are case sensitive and must be capitalized as indicated in the following tables.

Settings related to jobData.trigger are configured using get/setjobdata with the sub parameters displayed in the following table:

Table 113 trigger

Command	Parameter	Sub Parameter	Range	Example

`getjobdata`
`setjobdata`
	`.jobData.`
	`trigger`
	`.jt`
		`mode`	NONE
			SINGLE_SHOT
			AGGREGATE
			LEVEL_CONTINUOUS
			BURST
			PERIODIC_SINGLE_SHOT
			CONTINUOUS
			PRESENTATION
				gjd .jt.mode .slot 3
				command: gjd,ok
				{response}
				sjd .jt.mode LEVEL_CONTINUOUS
				.slot 3
				command: sjd,ok

`getjobdata`
`setjobdata`
	`.jobData.`
	`trigger`
	`.jt`
		`start_criteria`	GPIO
			DEVICE
			SERIAL
			PLC
			TCP_IP
			AUTO
			TEST_TRIGGER
				gjd .jt.start_criteria .slot 3
				command: gjd,ok
				{response}
				sjd .jt.start_criteria AUTO
				.slot 3
				command: sjd,ok

Table 113 trigger (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. trigger .jt	end_criteria	GPIO DEVICE SERIAL PLC TCP_IP AUTO TEST_TRIGGER	gjd .jt.end_criteria .slot 3 command: gjd,ok {response} sjd .jt.end_criteria AUTO .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger .jt	show_image	true/false	gjd .jt.show_image .slot 3 command: gjd,ok {response} sjd .jt.show_image true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger .jt	no_read _same_ barcode	Int	gjd .jt.no_read_same_barcode .slot 3 command: gjd,ok {response} sjd .jt.no_read_same_barcode 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger .jt	reset_dnr _on_trigger	true/false	gjd .jt.reset_dnr_on_trigger .slot 3 command: gjd,ok {response} sjd .jt.reset_dnr_on_trigger true .slot 3 command: sjd,ok

getjobdata .jobData.trigger.trigger_settings.same_barcode_timeout .slot 3

Settings related to jobData.trigger.trigger_settings are configured using get/setjobdata with the sub-parameters displayed in the following table:

Table 114 trigger_settings

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	same_barcode _timeout	Number (0-500 ms)	gjd .jts.same_barcode_timeout .slot 3 command: gjd,ok {response} sjd .jts.same_barcode_timeout 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	job_interval	Number (0-60000 ms)	gjd .jts.job_interval .slot 3 command: gjd,ok {response} sjd .jts.job_interval 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	burst_count	Number (1-100)	gjd .jts.burst_mode .slot 3 command: gjd,ok {response} sjd .jts.burst_count 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	presentation_ sensitivity	Number	gjd .jts.presentation_sensitivity .slot 3 command: gjd,ok {response} sjd .jts.presentation_sensitivity 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	decode_ failure_ timeout	Number (0-60000 ms)	gjd .jts.decode_failure_timeout .slot 3 command: gjd,ok {response} sjd .jts.decode_failure_timeout 1 .slot 3 command: sjd,ok

Table 114 trigger_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	active_level	LOW HIGH	gjd .jts.active_level .slot 3 command: gjd,ok {response} sjd .jts.active_level 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	fast_hw _trigger_ enabled	true/false	gjd .jts.fast_hw_trigger_enabled .slot 3 command: gjd,ok {response} sjd .jts.fast_hw_trigger_enabled true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	read_multiple _barcode	true/false	gjd .jts.read_multiple_barcodes .slot 3 command: gjd,ok {response} sjd .jts.read_multiple_barcodes true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	send _aggregate _level_assert_ data	true/false	gjd .jts.send_aggregate_level_assert_data .slot 3 command: gjd,ok {response} sjd .jtssend_aggregate_level_assert_data true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	disable_active _job_timeout	true/false	gjd .jts.disable_active_job_timeout .slot 3 command: gjd,ok {response} sjd .jts.disable_active_job_timeout true .slot 3 command: sjd,ok

Table 114 trigger_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. trigger trigger_settings .jts	same_data _timeout _must_leave _fov	true/false	gjd .jts. same_data_timeout_must_leave_fov .slot 3 command: gjd,ok {response} sjd .jts. same_data_timeout_must_leave_fov true .slot 3 command: sjd,ok

meta

getjobdata .jobData.meta.description .slot 3

Settings related to jobData.meta are configured using the get/setjobdatawith sub parameters displayed in this table:

Table 115 meta

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. meta .jm	description	String	gjd .jm.description .slot 3 command: gjd,ok {response} sjd .jm.description String .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. meta .jm	device	String	gjd .jm.device .slot 3 command: gjd,ok {response} sjd .jm.device String .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. meta .jm	filePath	Null|String	gjd .jm.filePath .slot 3 command: gjd,ok {response} sjd .jm.filePath String .slot 3 command: sjd,ok

Table 115 meta (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. meta .jm	title	String	gjd .jm.title .slot 3 command: gjd,ok {response} sjd .jm.title String .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. meta .jm	zoom_level	Number	gjd .jm.zoom_level .slot 3 command: gjd,ok {response} sjd .jm.zoom_level String .slot 3 command: sjd,ok

imager setups

getjobdata .jobData.imager_setups.aimer

Settings related to jobData.image_setups are configured using the getjobdata/setjobdata with the sub-parameters displayed in the following table:

Table 116 imager_setups

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. imager_setups .ji	aimer	true/false	gjd .ji.0.aimer .slot 3 command: gjd,ok {response} sjd .ji.0.aimer.true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	imager .auto_exposure	true/false	gjd .ji.0.imager.auto_exposure .slot 3 command: gjd,ok {response} sjd .ji.0.imager.auto_exposure .slot 3 command: sjd,ok

Table 116 imager_setups (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. imager_setups .ji	bank	Number	gjd .ji.0.bank .slot 3 command: gjd,ok {response} sjd .ji.0.bank true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	external_light .enabled	true/false	gjd .ji.0.external_light.enabled .slot 3 command: gjd,ok {response} sjd .ji.0.external_light.enabled true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	external_light .intensity	Number	gjd .ji.0.external_light.intensity .slot 3 command: gjd,ok {response} sjd .ji.0.external_light.intensity 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	internal_light .color	WHITE BLUE RED INFRARED GREEN YELLOW DPM	gjd .ji.0.internal_light.color .slot 3 command: gjd,ok {response} sjd .ji.0.internal_light.color true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	imager auto_focus	true/false	gjd .ji.0. imager.auto_focus .slot 3 command: gjd,ok {response} sjd .ji.0.imager.auto_focus true .slot 3 command: sjd,ok

Table 116 imager_setups (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. imager_setups .ji	imager.exposure	Number (0.5 to 14)	gjd .ji.0.imager.exposure .slot 3 command: gjd,ok {response} sjd .ji.0.imager.exposure 0.6 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	imager .gain	Number (0 to 100)	gjd .ji.0.imager.gain .slot 3 command: gjd,ok {response} sjd.ji.0.imager.gain 2 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	imager .focus	Number (-7 to +7)	gjd .ji.0.imager.focus .slot 3 command: gjd,ok {response} sjd .ji.0.imager.focus 3 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	imager .long_exposure	Number (1 to 10)	gjd .ji.0.imager.long_exposure .slot 3 command: gjd,ok {response} sjd .ji.0.imager.long_exposure 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	internal_light .enabled	true/false	gjd .ji.0.internal_light.enabled .slot 3 command: gjd,ok {response} sjd .ji.0.internal_light.enabled true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	internal_light .intensity	Number	gjd .ji.0.internal_light.intensity .slot 3 command: gjd,ok {response} sjd .ji.0.internal_light.intensity 1 .slot 3 command: sjd,ok

Table 116 imager_setups (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. imager_setups .ji	internal_light .max_duration	Number	gjd .ji.0.internal_light.max_duration .slot 3 command: gjd,ok {response} sjd .ji.0.internal_light.max_duration 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	internal_light .regions.bottom	true/false	gjd .ji.0.internal_light.regions.bottom .slot 3 command: gjd,ok {response} sjd .ji.0.internal_light.regions.bottom true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	internal_light .regions.top	true/false	gjd .ji.0.internal_light.regions.top .slot 3 command: gjd,ok {response} sjd .ji.0.internal_light.regions.top true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	internal_light .regions.right	true/false	gjd .ji.0.internal_light.regions.right .slot 3 command: gjd,ok {response} sjd .ji.0.internal_light.regions.right true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	internal_light .regions.left	true/false	gjd .ji.0.internal_light.regions.left .slot 3 command: gjd,ok {response} sjd .ji.0.internal_light.regions.left true .slot 3 command: sjd,ok

Table 116 imager_setups (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. imager_setups .ji	internal_light .torch_mode	true/false	gjd .ji.0.internal_light.torch_mode .slot 3 command: gjd,ok {response} sjd .ji.0.internal_light.torch_mode true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	name	String	gjd .ji.0.name .slot 3 command: gjd,ok {response} sjd .ji.0.name STRING .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	auto_tune .brightness	true/false	gjd .ji.0.auto_tune.brightness .slot 3 command: gjd,ok {response} sjd .ji.0.auto_tune.brightness true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	auto_tune .max_exposure	Number	gjd .ji.0.auto_tune.max_exposure .slot 3 command: gjd,ok {response} sjd .ji.0.auto_tune.max_exposure 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	auto_tune .tune_focus	true/false	gjd .ji.0.auto_tune.tune_focus .slot 3 command: gjd,ok {response} sjd .ji.0.auto_tune.tune_focus true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. imager_setups .ji	auto_tune .method	String	gjd .ji.0.auto_tune.method .slot 3 command: gjd,ok {response} sjd .ji.0.auto_tune.method STRING .slot 3 command: sjd,ok

industrial ethernet

getjobdata .jobData.industrial_ethernet.input.mode .slot 3

Settings related to jobData.industrial_ethernet are configured using the getjobdata/setjobdata with the sub-parameters in the following table:

Table 117 industrial_ethernet

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	input .mode	ENTRY RAW	gjd .jie.input.mode .slot 3 command: gjd,ok {response} sjd .jt.input.mode RAW . slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	input .switch_bits	true/false	gjd .jie.input.switch_bits .slot command: gjd,ok {response} sjd .jie.input.switch_bits true . slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	input.config .data_type	String	gjd .jie.input.config.0.data_type .slot 3 command: gjd,ok {response} sjd .jie.input.config.0.data_type STRING .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	input.config .field	String	gjd .jie.input.config.0.field .slot 3 command: gjd,ok {response} sjd .jie.input.config.0.field STRING .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	input.config .field_size	Number	gjd .jie.input.config.0.field_size .slot 3 command: gjd,ok {response} sjd .jie.input.config.0.field_size 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	input.config .isBase64	true/false	gjd .jie.input.config.0..isBase64 .slot 3 command: gjd,ok {response} sjd .jie.input.config.0.isBase64 true .slot 3 command: sjd,ok

Table 117 industrial_ethernet (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	input.config .order	Number	gjd .jie.input.config.0.order .slot 3 command: gjd,ok {response} sjd .jie.input.config.0.order 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	input.config .type	String	gjd .jie.input.config.0.type .slot 3 command: gjd,ok {response} sjd .jie.input.config.0.type String .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	input.config .type_id	String	gjd .jie.input.config.0.type_id .slot 3 command: gjd,ok {response} sjd .jie.input.config.0.type_id String .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output .mode	ENTRY RAW	gjd .jie.output.mode .slot 3 command: gjd,ok {response} sjd .jie output.mode RAW .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output .switch_bits	true/false	gjd .jie.output.switch_bits .slot 3 command: gjd,ok {response} sjd .jie.output.switch_bits true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .data_type	String	gjd.jie.output.config.0.data_type .slot 3 command: gjd,ok {response} sjd .jie.output.config.0.data_type String .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .field	String	gjd .jie.output.config.0.field .slot 3 command: gjd,ok {response} sjd .jie.output.config.0.field String .slot 3 command: sjd,ok

Table 117 industrial_ethernet (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .field_size	Number	gjd .jie.output.config.0.field_size .slot 3 command: gjd,ok {response} sjd .jie.output.config..0.field_size 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .isBase64	true/false	gjd .jie. output.config.0.isBase64 .slot 3 command: gjd,ok {response} sjd .jie. output.config.0.isBase64 true .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .order	Number	gjd .jie.output.config.0.order .slot 3 command: gjd,ok {response} sjd .jie. output.config.0.order 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .type	String	gjd .jie. output.config.0.type .slot 3 command: gjd,ok {response} sjd .jie.output.config.0.type String .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .type_id	String	gjd .jie.output.config.0.type_id .slot 3 command: gjd,ok {response} sjd .jie.output.config.0.type_id String .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .default.codec		gjd .jie.output.config.0.default.codec .slot 3 command: gjd,ok {response} sjd .jie.output.config.0.default.codec 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .default.ppm		gjd .jie.output.config.0.default.ppm .slot 3 command: gjd,ok {response} sjd .jie.output.0.config.default.ppm 1 .slot 3 command: sjd,ok

Table 117 industrial_ethernet (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. industrial_ ethernet .jie	output.config .default.quality_ score		gjd .jie. output.config.0.default.quality_score .slot 3 command: gjd,ok {response} sjd .jie. output.config.0.default.quality_score 1 .slot 3 command: sjd,ok

save options

getjobdata .jobData.save_options.conditions .slot 3

Settings related to jobData.save_options are configured using getjobdata/setjobdata with the sub parameters displayed in the following table:

Table 118 save options

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. save_options .jt	conditions	NO_READ GOOD_READ	gjd .js.0.conditions .slot 3 command: gjd,ok {response} sjd .js.0.conditions NO_READ. slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. save_options .jt	destination	ON_DEVICE FTP SFTP	gjd .js.0.destination .slot 3 command: gjd,ok {response} sjd .js.0.destination ON_DEVICE .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. save_options .jt	enabled	true/false	gjd .js.0.enabled .slot 3 command: gjd,ok {response} sjd .js.0.enabled true .slot 3 command: sjd,ok

Table 118 save options (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. save_options .jt	file_prefix	String	gjd .js.0.file_prefix .slot 3 command: gjd,ok {response} sjd .js.0.file_prefix ZEB .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. save_options .jt	file_suffix	DATETIME INDEX	gjd .js.0.file_suffix .slot 3 command: gjd,ok {response} sjd .js.0.file_suffix INDEX .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. save_options .jt	format	BMP JPG	gjd .js.0.format .slot 3 command: gjd,ok {response} sjd .js.0.format JPG .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. save_options .jt	id	Number	gjd .js.0.id .slot 3 command: gjd,ok {response} sjd .js.0.id 2 .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. save_options .jt	remote_settings	ftp/sftp server settings	gjd .js.0.remote_settings .slot 3 command: gjd,ok {response} sjd .js.0.remote_settings {ftp server settings} .slot 3 command: sjd,ok
getjobdata setjobdata	.jobData. save_options .jt	size	FULL QUARTER ONE_SIXTEENTH ONE_SIXTY_FOURTH	gjd .js.0.size .slot 3 command: gjd,ok {response} sjd .js.0.size FULL .slot 3 command: sjd,ok

Table 118 save options (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.jobData. save_options .jt	trigger	String	gjd .js.0.trigger .slot 3 command: gjd,ok {response} sjd .js.0.trigger IMAGES_PER_RESULT .slot 3 command: sjd,ok

barcode configuration

getjobdata .job_run_data.graph.nodes.BARCODE.configuration.type .slot 2

Settings related to job_run_data.graph.nodes.BARCODE.configuration are configured using getjobdata/setjobdata with the sub-parameters in the following table:

Table 119 BARCODE.configuration

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	roi.data .height	Number	gjd .jrc.roi.data.height .slot 3 command: gjd,ok {response} sjd .jrc.roi.data.height 3 . slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	roi.data .rotation	Number	gjd .jrc.roi.data.rotation .slot 3 command: gjd,ok {response} sjd .jrc.roi.data.rotation 1 . slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	roi.data .width	Number	gjd .jrc.roi.data.width .slot 3 command: gjd,ok {response} sjd .jrc.roi.data.width 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	roi.data.x	Number	gjd .jrc.roi.data.x .slot 3 command: gjd,ok {response} sjd .jrc.roi.data.x 1 .slot 3 command: sjd,ok

Table 119 BARCODE.configuration (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	roi.data.y	Number	gjd .jrc.roi.data.y .slot 3 command: gjd,ok {response} sjd .jrc.roi.data.y 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	roi.type	RECTANGLE POLYGON CIRCLE ANNULUS	gjd .jrc.roi.type .slot 3 command: gjd,ok {response} sjd .jrc.roi.type ANNULUS .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	source _images .type	ACQUISITION TOOL	gjd .jrc.source_images.type .slot 3 command: gjd,ok {response} sjd .jrc.source_images.type TOOL .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	source _images .value	String	gjd .jrc.source_images.value .slot 3 command: gjd,ok {response} sjd .jrc.source_images.value TOOL .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	Enabled	true/false	gjd .jrc.enabled .slot 3 command: gjd,ok {response} sjd .jrc.enabled true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	Invert	true/false	gjd .jrc.invert .slot 3 command: gjd,ok {response} sjd .jrc.invert true .slot 3 command: sjd,ok

Table 119 BARCODE.configuration (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	data _formatting	COMPLEX	gjd .jrc.data_formatting .slot 3 command: gjd,ok {response} sjd .jrc.data_formatting COMPLEX .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	localData .friendly _name	String	gjd .jrc.localData.friendly_name .slot 3 command: gjd,ok {response} sjd .jrc.localData.friendly_name 'Reads'.slot 3 command: sjd,ok 'Reads'
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrc	localData .barcode _data_ formattin g_mode	STANDARD TRAILING_TAB ADVANCED TRAILING_LINE _BREAK	gjd .jrc.localData. barcode_data_formatting_mode .slot 3 command: gjd,ok {response} sjd .jrc.localData. barcode_data_formatting_mode STANDARD .slot 3 command: sjd,ok

Table 119 BARCODE.configuration (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration .jrd	decode.match_string .all (update all similiar tools in a job) .base64 (when base64 encoded string to be given) .friendly_name (update string should be within single quotes based on friendly name of a tool)		**string should be within single quotes** sjd .jrc.decode.match_string 'string to be matched' **To update all match strings of ocr tools in a job** sjd .jrc.decode.match_string 'string to be matched' .all **To update all match strings of ocr tools in a job of a given slot number** sjd .jrc.decode.match_string 'string to be matched' .all .slot 19 **To update match string (base64 encoded) to a tool , it should not be within any quotes** sjd .jrc.decode.match_string aGVsbG8gdBiYXNlNg .base64 **To update match string (base64 encoded) to all tools** sjd .jrc.decode.match_string aGVsbG8gdBiYXNlNg .base64 .all **To update match string (base64 encoded) to all tools with a given slot number** sjd .jrc.decode.match_string aGVsbG8gdBiYXNlNg .base64 .all .slot 19

data formatting

getjobdata .job_run_data.graph.nodes.BARCODE.configuration.barcode_data_formatting

.all_symbologies.data_types.encoding .slot 2

Settings related to job_run_data.graph.nodes.BARCODE.configuration.barcode_data_formatting.all_symbologies are configured using the getjobdata/setjobdata with the sub parameters displayed in the following table:

Table 120 barcode_data_formatting .all_symbologies

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .all_symbologies .jra	data_type _delimiter	NONE SPACE COMMA TAB LINEBREAK	gjd .jra.data_type_delimiter .slot 3 command: gjd,ok {response} sjd .jra.data_type_delimiter TAB ZEB . slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .all_symbologies .jra	data_types. encoding	TEXT BASE64	gjd .jra.data_types. encoding .slot 3 command: gjd,ok {response} sjd .jra.data_types. encoding TEXT . slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .all_symbologies .jra	data_types .type	DECODED_STRING PPM SUBDECODED _STRING ANGLE SYMBOLOGY CENTER_X CENTER_Y	gjd .jra.data_types_type .slot 3 command: gjd,ok {response} sjd .jra.data_types_type ANGLE .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .all_symbologies .jra	end_delimiter	NONE SPACE COMMA TAB LINEBREAK	gjd .jra.end_delmiter .slot 3 command: gjd,ok {response} sjd .jra.end_delimiter TAB .slot 3 command: sjd,ok

Table 120 barcode_data_formatting .all_symbologies (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .all_symbologies .jra	Prefix	String	gjd .jra.prefix .slot 3 command: gjd,ok {response} sjd .jra.prefix STRING .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .all_symbologies .jra	Suffix	String	gjd .jra.suffix .slot 3 command: gjd,ok {response} sjd .jra.suffix STRING . slot 3 command: sjd,ok

manycode

getjobdata .
job_run_data.graph.nodes.BARCODE.configuration.barcode_data_formatting.many_code.prefix .slot 2

gjd .jrm.prefix .slot 2

Settings related to
job_run_data.graph.nodes.BARCODE.configuration.barcode_data_formatting.many_code are configured using the getjobdata /setjobdata with the sub parameters in the following table:

Table 121 many_code

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .many_code .jrm	data_type_ delimiter	NONE SPACE COMMA TAB LINEBREAK	gjd .jrm.data_type_delimiter .slot 3 command: gjd,ok {response} sjd .jrm.data_type_delimiter TAB .slot 3 command: sjd,ok

Table 121 many_code (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .many_code .jrm	data_types. encoding	TEXT BASE64	gjd .jrm.data_types. encoding .slot 3 command: gjd,ok {response} sjd .jrm.data_types. encoding TEXT .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .many_code .jrm	data_types .type	DECODED_STRING PPM SUBDECODED _STRING ANGLE SYMBOLOGY CENTER_X CENTER_Y	gjd .jrm.data_types.type .slot 3 command: gjd,ok {response} sjd .jrm.data_types.type ANGLE .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .many_code .jrm	end_delimiter	NONE SPACE COMMA TAB LINEBREAK	gjd .jrm.end_delimiter .slot 3 command: gjd,ok {response} sjd .jrm.end_delimiter TAB .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .many_code .jrm	Prefix	String	gjd .jrm.prefix .slot 3 command: gjd,ok {response} sjd .jrm.prefix STRING .slot 3 command: sjd,ok

Table 121 many_code (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. barcode_data_ formatting .many_code .jrm	Suffix	String	gjd .jrm.suffix .slot 3 command: gjd,ok {response} sjd .jrm.suffix STRING . slot 3 command: sjd,ok

decode

getjobdata .job_run_data.graph.nodes.BARCODE.configuration.decode.Inverse1D .slot 2

The settings related to job_run_data.graph.nodes.BARCODE.configuration.decode can be configured using the getjobdata/setjobdata along with the sub parameter as shown in the below table

Table 122 BARCODE.configuration.decode

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	adaptive_roi_ scale_factor	SMALL MEDIUM LARGE	gjd .jrd.adaptive_roi_scale_factor .slot 3 command: gjd,ok {response} sjd .jrd.adaptive_roi_scale_factor SMALL .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	Inverse1D	Number	gjd .jrd.Inverse1D .slot 3 command: gjd,ok {response} sjd .jrd.Inverse1D 1 .slot 3 command: sjd,ok

Table 122 BARCODE.configuration.decode (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	advanced_decode_ settings. allow_rectangular _codes	true/false	gjd .jrd.advanced_decode_settings. allow_rectangular_codes .slot 3 command: gjd,ok {response} sjd .jrd.advanced_decode_settings. allow_rectangular_codes true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	advanced_decode_ settings. contrast_threshold	Number	gjd .jrd.advanced_decode_settings. contrast_threshold .slot 3 command: gjd,ok {response} sjd .jrd.advanced_decode_settings. contrast_threshold 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	advanced_decode_ settings. decode_strategy	FAST MODERATE EXHAUSTIVE	gjd .jrd.advanced_decode_settings. decode_strategy .slot 3 command: gjd,ok {response} sjd .jrd.advanced_decode_settings. decode_strategy FAST .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	advanced_decode_ settings. detection_method	UNIFORM FINDER_PATTERN QUIET_ZONE	gjd .jrd.advanced_decode_settings. detection_method .slot 3 command: gjd,ok {response} sjd .jrd.advanced_decode_settings. detection_method UNIFORM .slot 3 command: sjd,ok

Table 122 BARCODE.configuration.decode (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	advanced_decode_ settings. max_module_size	Number	gjd .jrd.advanced_decode_settings. max_module_size .slot 3 command: gjd,ok {response} sjd .jrd.advanced_decode_settings. max_module_size 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	advanced_decode_ settings. min_module_size	Number	gjd .jrd.advanced_decode_settings. min_module_size .slot 3 command: gjd,ok {response} sjd .jrd.advanced_decode_settings. min_module_size 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	advanced_decode_ settings. min_row_count	Number	gjd .jrd.mode .slot 3 command: gjd,ok {response} sjd .jrd.mode CONTINUOUS .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	advanced_decode_ settings. max_row_count	Number	gjd .jrd.advanced_decode_settings. max_row_count .slot 3 command: gjd,ok {response} sjd .jrd.advanced_decode_settings. max_row_count 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	advanced_decode_ settings. min_column_count	Number	gjd .jrd.advanced_decode_settings. min_column_count .slot 3 command: gjd,ok {response} sjd .jrd.advanced_decode_settings. min_column_count 1 .slot 3 command: sjd,ok

Table 122 BARCODE.configuration.decode (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. .job_run_data. graph.nodes. BARCODE. configuration. decode .jrd BARCODE. configuration. decode .jrd	advanced_decode_ settings. max_column_count	Number	gjd .jrd.advanced_decode_settings. max_column_count .slot 3 command: gjd,ok {response} sjd .jrd.advanced_decode_settings. max_column_count 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	Priority	NONE 1D 2D	gjd .jrd.priority .slot 3 command: gjd,ok {response} sjd .jrd.priority NONE .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	barcode_dpm	true/false	gjd .jrd.barcode_dpm .slot 3 command: gjd,ok {response} sjd .jrd.barcode_dpm true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	decode_all_ barcodes	true/false	gjd .jrd.decode_all_barcodes .slot 3 command: gjd,ok {response} sjd .jrd.decode_all_barcodes true .slot 3 command: sjd,ok

Table 122 BARCODE.configuration.decode (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	dpm	true/false	gjd .jrd.dpm .slot 3 command: gjd,ok {response} sjd .jrd.dpm true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	enable_adaptive_ roi_search	true/false	gjd .jrd.enable_adaptive_roi_search .slot 3 command: gjd,ok {response} sjd .jrd.enable_adaptive_roi_search true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	enable_bqm	true/false	gjd .jrd.enable_bqm .slot 3 command: gjd,ok {response} sjd .jrd.enable_bqm true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	enable_identical_ decode_symbols	true/false	gjd .jrd.enable_identical_decode_ symbols .slot 3 command: gjd,ok {response} sjd .jrd.enable_identical_decode_ symbols true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	enable_partial_ results	true/false	gjd .jrd.enable_partial_results .slot 3 command: gjd,ok {response} sjd .jrd.enable_partial_results true .slot 3 command: sjd,ok

Table 122 BARCODE.configuration.decode (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	enable_string_ match_on_ entire_payload	true/false	gjd .jrd.enable_string_match_on_ entire_payload .slot 3 command: gjd,ok {response} sjd .jrd.enable_string_match_on_ entire_payload true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	exhaustive_ manycode	true/false	gjd .jrd.exhaustive_manycode .slot 3 command: gjd,ok {response} sjd .jrd.mode exhaustive_manycode true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	match_mode	DISABLED STRING_EXACT STRING_CONTAINS REGEX	gjd .jrd.match_mode .slot 3 command: gjd,ok {response} sjd .jrd.match_mode DISABLED .slot 3 command: sjd,ok

Table 122 BARCODE.configuration.decode (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	match_string .all (update all similiar tools in a job) .base64 (when base64 encoded string is to be provided) .friendly_name (update based on the friendly name of a tool)	String	string should be within single quotes sjd .jrd.match_string 'string to be matched' **To update all match strings of barcode tools in a job:** sjd .jrd.match_string 'string to be matched' .all **To update all match strings of barcode tools in a job of a given slot number:** sjd .jrd.match_string 'string to be matched' .all .slot 19 **To update match string (base64 encoded) to a tool, it should not be within any quotes:** sjd .jrd.match_string aGVsbGBiYXNlNg .base64 **To update match string (base64 encoded) for all tools:** sjd .jrd.match_string aGVsbGBiYXNlNg .base64 .all **To update match string (base64 encoded) to all tools with a given slot number:** sjd .jrd.match_string aGVsbGBiYXNlNg .base64 .all .slot 19 **To update the match string to a specific tool based on a friendly name:** sjd .jrd.match_string 'this is to change only barcode1' .friendly_name 'barcode1'
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	multicode	true/false	gjd .jrd.multicode .slot 3 command: gjd,ok {response} sjd .jrd.multicode true .slot 3 command: sjd,ok

Table 122 BARCODE.configuration.decode (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	no_read_string	String	gjd .jrd.no_read_string .slot 3 command: gjd,ok {response} sjd .jrd.no_read_string STRING .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	ocr	true/false	gjd .jrd.ocr .slot 3 command: gjd,ok {response} sjd .jrd.ocr true .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	number_barcodes _to_decode	Number	gjd .jrd. number_barcodes_to_decode .slot 3 command: gjd,ok {response} sjd .jrd. number_barcodes_to_decode 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	percentSymbolInRoi	Number	gjd .jrd.percentSymbolInRoi .slot 3 command: gjd,ok {response} sjd .jrd.percentSymbolInRoi 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. decode .jrd	sorting_type	ALPHABETICAL FIRST_DECODED HORIZONTAL VERTICAL SYMBOLOGY	gjd .jrd.sorting_type .slot 3 command: gjd,ok {response} sjd .jrd.sorting_type VERTICAL .slot 3 command: sjd,ok

symbology settings

getjobdata .job_run_data.graph.nodes.BARCODE.configuration.symbology_settings.CODE128.Code128Length1 .slot 2

gjd .jrm.prefix .slot 2

The settings related to job_run_data.graph.nodes.BARCODE.configuration.barcode_data_formatting.symbology_settings are configured using getjobdata/setjobdata with the sub parameters displayed in the following table:

Table 123 BARCODE.configuration.symbology_settings

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE128 .Code128Length1	Number	gjd .jrs.CODE128.Code128Length1 .slot 3 command: gjd,ok {response} sjd .jrs.CODE128.Code128Length1 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE128 .Code128Length2	Number	gjd .jrs.CODE128.Code128Length2 .slot 3 command: gjd,ok {response} sjd .jrs.CODE128.Code128Length2 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE128 .GS1_128	Number	gjd .jrs.CODE128.GS1_128 .slot 3 command: gjd,ok {response} sjd .jrs.CODE128.GS1_128 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE128 .ISBT_ Concatenation	Number	gjd .jrs.CODE128.ISBT_Concatenation .slot 3 command: gjd,ok {response} sjd .jrs.CODE128.ISBT_Concatenation 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE128 . ISBT_ Concatenation_ Redundancy	Number	gjd .jrs.CODE128. ISBT_Concatenation_Redundancy .slot 3 command: gjd,ok {response} sjd .jrs.CODE128. ISBT_Concatenation_Redundancy 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE128 .ISBT128	Number	gjd .jrs.CODE128.ISBT128 .slot 3 command: gjd,ok {response} sjd .jrs.CODE128.ISBT128 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE128 .IgnoreCode 128FNC4	Number	gjd .jrs.CODE128.IgnoreCode128FNC4 .slot 3 command: gjd,ok {response} sjd .jrs.CODE128.IgnoreCode128FNC4 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE39 .Code39 CheckDigit Verification	Number	gjd .jrs.CODE39. Code39CheckDigitVerification .slot 3 command: gjd,ok {response} sjd .jrs.CODE39. Code39CheckDigitVerification 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE39 .Code39 FullASCII Conversion	Number	gjd .jrs.CODE39. Code39FullASCIIConversion .slot 3 command: gjd,ok {response} sjd .jt.CODE39. Code39FullASCIIConversion 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE39 .ConvertCode39 toCode32	Number	gjd .jrs.CODE39. ConvertCode39toCode32 .slot 3 command: gjd,ok {response} sjd .jrs.CODE39. ConvertCode39toCode32 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE39 .LengthforCode39 Length1	Number	gjd .jrs.CODE39. LengthforCode39Length1 .slot 3 command: gjd,ok {response} sjd .jrs.CODE39. LengthforCode39Length1 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE39 .LengthforCode39 Length2	Number	gjd .jrs.mode .slot 3 command: gjd,ok {response} sjd .jrs.mode CONTINUOUS .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE39 .TransmitCode39 CheckDigit	Number	gjd .jrs.CODE39. TransmitCode39CheckDigit .slot 3 command: gjd,ok {response} sjd .jrs.CODE39. TransmitCode39CheckDigit 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE39 .TriopticCode39	Number	gjd .jrs.CODE39. TriopticCode39 .slot 3 command: gjd,ok {response} sjd .jrs.CODE39. TriopticCode39 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE93 .LengthforCode93 Length1	Number	gjd .jrs.CODE93. LengthforCode93Length1 .slot 3 command: gjd,ok {response} sjd .jrs.CODE93. LengthforCode93Length1 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	CODE93 .LengthforCode93 Length2	Number	gjd .jrs.CODE93. LengthforCode93Length2 .slot 3 command: gjd,ok {response} sjd .jrs.CODE93. LengthforCode93Length2 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	DATAMATRIX .DPM_Mode	Number	gjd .jrs. DATAMATRIX.DPM_Mode .slot 3 command: gjd,ok {response} sjd .jrs. DATAMATRIX.DPM_Mode 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	DATAMATRIX .Decode_Data_ Matrix_Mirror_ Images	Number	gjd .jrs.DATAMATRIX. Decode_Data_Matrix_Mirror_Images .slot 3 command: gjd,ok {response} sjd .jrs.DATAMATRIX. Decode_Data_Matrix_Mirror_Images 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	DATAMATRIX .InverseDataMatrix	Number	gjd .jrs. DATAMATRIX.InverseDataMatrix .slot 3 command: gjd,ok {response} sjd . DATAMATRIX.InverseDataMatrix 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	DATAMATRIX .GS1_Datamatrix	Number	gjd .jrs. DATAMATRIX.GS1_Datamatrix .slot 3 command: gjd,ok {response} sjd .jrs. DATAMATRIX.GS1_Datamatrix 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	DOTCODE .ECCErasure DotCode	Number	gjd .jrs. DOTCODE.ECCErasureDotCode .slot 3 command: gjd,ok {response} sjd .jrs. DOTCODE.ECCErasureDotCode 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	DOTCODE .InverseDotCode	Number	gjd .jrs. DOTCODE.InverseDotCode .slot 3 command: gjd,ok {response} sjd .jrs. DOTCODE.InverseDotCode 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	DOTCODE .MirrorDotCode	Number	gjd .jrs. DOTCODE.MirrorDotCode .slot 3 command: gjd,ok {response} sjd .jrs. DOTCODE.MirrorDotCode 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	DOTCODE .PrioritizeDotCode	Number	gjd .jrs. DOTCODE.PrioritizeDotCode .slot 3 command: gjd,ok {response} sjd .jrs. DOTCODE.PrioritizeDotCode 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	I25 .ConvertI2of5to EAN13	Number	gjd .jrs. I25.ConvertI2of5toEAN13 .slot 3 command: gjd,ok {response} sjd .jrs. I25.ConvertI2of5toEAN13 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	I25 .Febraban (I2of5)	Number	gjd .jrs. I25.Febraban(I2of5) .slot 3 command: gjd,ok {response} sjd .jrsI25.Febraban(I2of5) 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	I25 .LengthforI2of5 Length1	Number	gjd .jrs. I25.LengthforI2of5Length1 .slot 3 command: gjd,ok {response} sjd .jrs. I25.LengthforI2of5Length1 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	I25 .LengthforI2of5 Length2	Number	gjd .jrs. I25.LengthforI2of5Length2 .slot 3 command: gjd,ok {response} sjd .jrs. I25.LengthforI2of5Length2 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	I25 .TransmitI2of5 CheckDigit	Number	gjd .jrs. I25.TransmitI2of5CheckDigit .slot 3 command: gjd,ok {response} sjd .jrs. I25.TransmitI2of5CheckDigit 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	MAXICODE	Number	gjd .jrs.MAXICODE .slot 3 command: gjd,ok {response} sjd .jrs.MAXICODE 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	PDF-417 .MicroPDF	Number	gjd .jrs.PDF-417.MicroPDF .slot 3 command: gjd,ok {response} sjd .jrs.PDF-417.MicroPDF 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	QRCODE .GS1_QR	Number	gjd .jrs.QRCODE.GS1_QR .slot 3 command: gjd,ok {response} sjd .jrs.QRCODE.GS1_QR .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	QRCODE .Linked_QR_Mode	Number	gjd .jrs.QRCODE.GS1_QR .slot 3 command: gjd,ok {response} sjd .jrs.QRCODE.GS1_QR 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	QRCODE .MicroQREnable	Number	gjd .jrs.QRCODE.MicroQREnable .slot 3 command: gjd,ok {response} sjd .jrs.QRCODE.MicroQREnable 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .Bookland_Format	Number	gjd .jrs.UPCEAN.Bookland_Format .slot 3 command: gjd,ok {response} sjd .jrs.UPCEAN.Bookland_Format 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .BooklandEAN	Number	gjd .jrs. UPCEAN.BooklandEAN .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.BooklandEAN 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .Convert_UPC_ E_to_A	Number	gjd .jrs. UPCEAN.Convert_UPC_E_to_A .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.Convert_UPC_E_to_A 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .Convert_UPC_ E1_to_A	Number	gjd .jrs. UPCEAN.Convert_UPC_E1_to_A .slot 3 command: gjd,ok {response} sjd .jrs.mode UPCEAN.Convert_UPC_E1_to_ 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .EAN_13_JAN13	Number	gjd .jrs. UPCEAN.EAN_13_JAN13 .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.EAN_13_JAN13 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .EAN_8_JAN8	Number	gjd .jrs. UPCEAN.EAN_8_JAN8 .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.EAN_8_JAN8 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .EAN_8_JAN_8_ Extend	Number	gjd .jrs. UPCEAN.EAN_8_JAN_8_Extend .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.EAN_8_JAN_8_Extend 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .ISSN_EAN	Number	gjd .jrs. UPCEAN.ISSN_EAN .slot 3 command: gjd,ok {response} sjd .jrs .UPCEAN.ISSN_EAN 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .Transmit_UPC_E1_ Check_Digit	Number	gjd .jrs. UPCEAN. Transmit_UPC_E1_Check_Digit .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN. Transmit_UPC_E1_Check_Digit 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .TransmitUPC_A CheckDigit	Number	gjd .jrs. UPCEAN.TransmitUPC_ACheckDigit .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.TransmitUPC_ACheckDigit 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN . UCC_Coupon_ Extended_Code	Number	gjd .jrs. UPCEAN. UCC_Coupon_Extended_Code .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN. UCC_Coupon_Extended_Code 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN . UPC_EAN_JAN Supplementals	Number	gjd .jrs. UPCEAN. UPC_EAN_JANSupplementals .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN. UPC_EAN_JANSupplementals 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .UPC/EAN /JAN _Supplemental _Redundancy	Number	gjd .jrs. UPCEAN.UPC/EAN / JAN_Supplemental_Redundancy .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.UPC/EAN / JAN_Supplemental_Redundancy 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .UPC_A	Number	gjd .jrs. UPCEAN.UPC_A .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.UPC_A 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .UPC_A_ Preamble	Number	gjd .jrs. UPCEAN.UPC_A_Preamble .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.UPC_A_Preamble 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .UPC_E	Number	gjd .jrs. UPCEAN.UPC_E .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.UPC_E 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .UPC_E_ Preamble	Number	gjd .jrs. UPCEAN.UPC_E_Preamble .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.UPC_E_Preamble 1 .slot 3 command: sjd,ok

Table 123 BARCODE.configuration.symbology_settings (Continued)

Command	Parameter	Sub Parameter	Range	Example
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .UPC_E1	Number	gjd .jrs. UPCEAN.UPC_E1 .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.UPC_E1 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .UPC_E1_ Preamble	Number	gjd .jrs. UPCEAN.UPC_E1_Preamble .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.UPC_E1_Preamble 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .UserDefined Supp1	Number	gjd .jrs.UPCEAN.UserDefinedSupp1 .slot 3 command: gjd,ok {response} sjd .jrs.UPCEAN.UserDefinedSupp1 1 .slot 3 command: sjd,ok
getjobdata setjobdata	.job_run_data. graph.nodes. BARCODE. configuration. symbology_settings .jrs	UPCEAN .UserDefined Supp2	Number	gjd .jrs. UPCEAN.UserDefinedSupp2 .slot 3 command: gjd,ok {response} sjd .jrs. UPCEAN.UserDefinedSupp2 1 .slot 3 command: sjd,ok

uploadjob

Use the `uploadjob` command to upload the job to the device.

NOTE: This command requires the zjob in base64 format as an argument.

Table 124 uploadjob

Command	short	Parameter	Data	Range	Example
`uploadjob`	uj	-	zjob (base64 format)	-	Uploadjob base64_data command:uploadjob,ok

NOTE: Use a python script to transfer large base64 files.