Gocator Snapshot Sensors

USER MANUAL

Gocator 3210 & 3500 Series Sensors

Firmware version: 6.0.x.xx

Document revision: A

Copyright

Contact Information

LMI Technologies, Inc.
9200 Glenlyon Parkway
Burnaby BC V5J 5J8
Canada

Telephone: +1 604-636-1011
Fax: +1 604-516-8368

www.lmi3D.com

Table of Contents

Introduction

This documentation describes how to connect, configure, and use a Gocator. It also contains reference information on the device's protocols and job files, as well as an overview of the development kits you can use with Gocator. Finally, the documentation describes the Gocator emulator and accelerator applications.

The documentation applies to the following:

- Gocator 3210
- Gocator 3500 series

Notational Conventions

This documentation uses the following notational conventions:

⚠	Follow these safety guidelines to avoid potential injury or property damage.

	Consider this information in order to make best use of the product.

Gocator Overview

Gocator sensors are designed for 3D measurement and control applications. Sensors are configured using a web browser and can be connected to a variety of input and output devices. Sensors can also be configured using the provided development kits.

Safety and Maintenance

The following sections describe the safe use and maintenance of Gocator sensors.

Electrical Safety

⚠	Failure to follow the guidelines described in this section may result in electrical shock or equipment damage.

Sensors should be connected to earth ground
All sensors should be connected to earth ground through their housing. All sensors should be mounted on an earth grounded frame using electrically conductive hardware to ensure the housing of the sensor is connected to earth ground. Use a multi-meter to check the continuity between the sensor connector and earth ground to ensure a proper connection.

Minimize voltage potential between system ground and sensor ground
Care should be taken to minimize the voltage potential between system ground (ground reference for I/O signals) and sensor ground. This voltage potential can be determined by measuring the voltage between Analog_out- and system ground. The maximum permissible voltage potential is 12 V but should be kept below 10 V to avoid damage to the serial and encoder connections.

For a description of the connector pins, see *Gocator I/O Connector* on page 894.

Use a suitable power supply
The power supply used with sensors should be an isolated supply with inrush current protection or be able to handle a high capacitive load. Verify the voltage input requirements for your sensor in the sensor's specifications; for specifications, see *Sensors* on page 880.

Use care when handling powered devices
Wires connecting to the sensor should not be handled while the sensor is powered. Doing so may cause electrical shock to the user or damage to the equipment.

Handling, Cleaning, and Maintenance

	Dirty or damaged sensor windows (emitter or camera) can affect accuracy. Use caution when handling the sensor or cleaning the sensor's windows.

Keep sensor windows clean
Use dry, clean air to remove dust or other dirt particles. If dirt remains, clean the windows carefully with a soft, lint-free cloth and non-streaking glass cleaner or isopropyl alcohol. Ensure that no residue is left on the windows after cleaning.

Avoid excessive modifications to files stored on the sensor

Sensor settings are stored in flash memory inside the sensor. Flash memory has an expected lifetime of 100,000 writes. To maximize lifetime, avoid frequent or unnecessary file save operations.

Environment and Lighting

Avoid strong ambient light sources

The imager used in this product is highly sensitive to ambient light. Do not operate this device near windows or lighting fixtures that could influence measurement or data acquisition.If the unit must be installed in an environment with high ambient light levels, a lighting shield or similar device may need to be installed to prevent light from affecting measurement.

Avoid installing sensors in hazardous environments

To ensure reliable operation and to prevent damage to sensors, avoid installing the sensor in locations

- that are humid, dusty, or poorly ventilated;
- with a high temperature, such as places exposed to direct sunlight;
- where there are flammable or corrosive gases;
- where the unit may be directly subjected to harsh vibration or impact;
- where water, oil, or chemicals may splash onto the unit;
- where static electricity is easily generated.

Ensure that ambient conditions are within specifications

Sensors are suitable for operation between 0–50° C (0–40° C for Gocator 2500 sensors) and 25–85% relative humidity (non-condensing). The storage temperature is -30–70° C.

The Master network controllers are similarly rated for operation between 0–50° C.

⚠	The sensor must be heat-sunk through the frame it is mounted to. When a sensor is properly heat sunk, the difference between ambient temperature and the temperature reported in the sensor's health channel is less than 15° C.

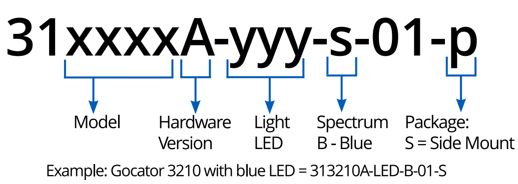
⚠	Sensors are high-accuracy devices, and the temperature of all of its components must therefore be in equilibrium. When the sensor is powered up, a warm-up time of at least one hour is required to reach a consistent spread of temperature in the sensor.

Getting Started

The following sections provide system and hardware overviews, in addition to installation and setup procedures.

Sensor Part Numbers

Use the following to understand sensor part numbers:



Upgrade Path

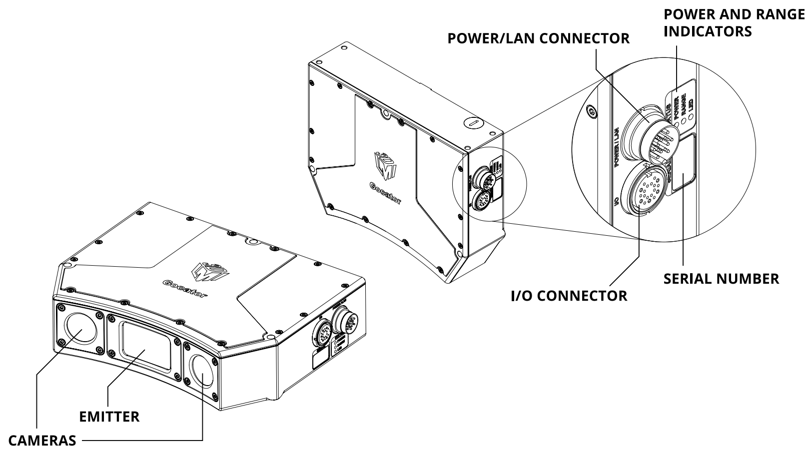
If you are upgrading from a 3.6 or 4.x firmware indicated in the upgrade path below, make sure to follow the sequence of firmware upgrades.

3.6 → 3.6 SR5 → 4.4 → 4.6 SR2 → 5.x/6.x

Hardware Overview

The following sections describe Gocator and its associated hardware.

Gocator Sensor



Gocator 3210

Item	Description
Camera	Observes light reflected from target surfaces.
Light Emitter	Emits structured light for 3D data acquisition.
I/O Connector	Accepts input/output signals.
Power/LAN Connector	Accepts power and connects to 1000 Mbit/s Ethernet network.
Power Indicator	Illuminates when power is applied (blue).
Range Indicator	Illuminates when the camera detects light that is within the sensor's measurement range (green).
LED Indicator	Illuminates when safety input is active (amber). Note that activating the safety input is *not* required to use Gocator snapshot sensors.
Serial Number	Unique sensor serial number.

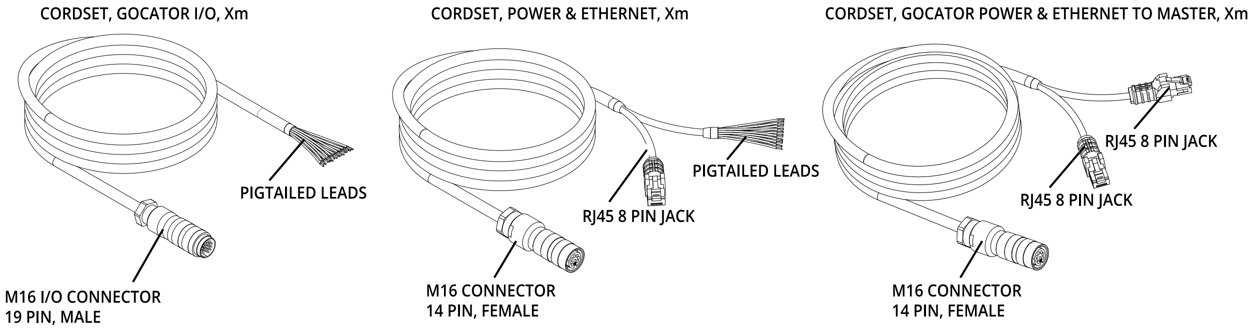
Gocator 3x00 Cordsets

G3 sensors use two types of cordsets.

The Power & Ethernet cordset provides power and safety interlock to the sensor. It is also used for sensor communication via 1000 Mbit/s Ethernet with a standard RJ45 connector. The Master version of the Power & Ethernet cordset provides direct connection between the sensor and a Master network controller (excluding Master 100); for more information, see *Master Network Controllers* on page 900.

The I/O cordset provides digital I/O connections, an encoder interface, RS-485 serial connection, and an analog output.

> If you use a Master 100 with a sensor, you must use the latest version of the Master 100 with a metal-shielded Master Power Port and a Power & Ethernet cordset with a metal-shielded Power/Sync RJ45 plug.

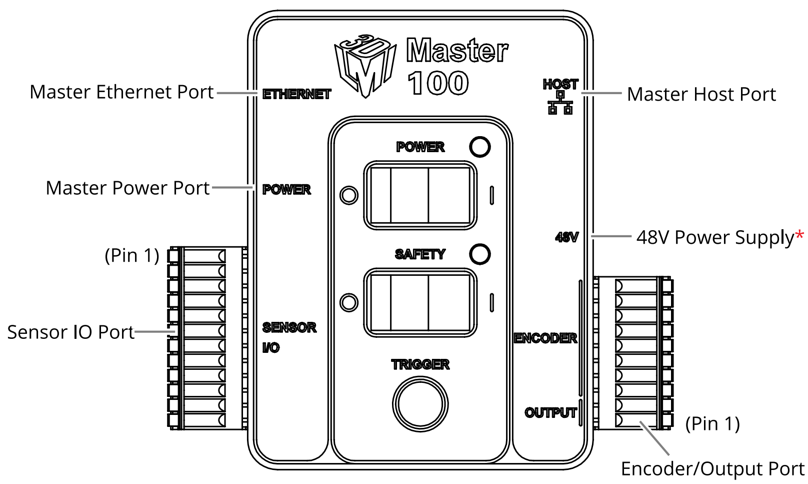


Cordsets are available with two types of cabling: standard and high flex. Use the high flex versions in applications where a tighter bend in the cordset is required, especially when the cordset must bend repeatedly. For more information, see *Cordset Bend Radius Limits* on page 26.

The maximum cordset length is 60 m. For pinout details, see *Gocator I/O Connector* on page 894 and *Gocator Power/LAN Connector* on page 892.

See *Accessories* on page 918 for cordset lengths and part numbers. Contact LMI for information on creating cordsets with customized lengths and connector orientations.

Master 100



Item	Description
Master Ethernet Port	Connects to the RJ45 connector labeled Ethernet on the Power/LAN to Master cordset.
Master Power Port	Connects to the RJ45 connector labeled Power/Sync on the Power/LAN to Master cordset. Provides power to the sensor.
Sensor I/O Port	Connects to the I/O cordset.
Master Host Port	Connects to the host PC's Ethernet port.
Power	Accepts power (+48 V).
Power Switch	Toggles sensor power.
Safety Switch	Toggles safety signal provided to the sensors [O= off, I= on]. Turning this switch on is *not* required with snapshot sensors.

Item	Description
Trigger	Signals a digital input trigger to the sensor.
Encoder	Accepts encoder A, B and Z signals.
Digital Output	Provides digital output.

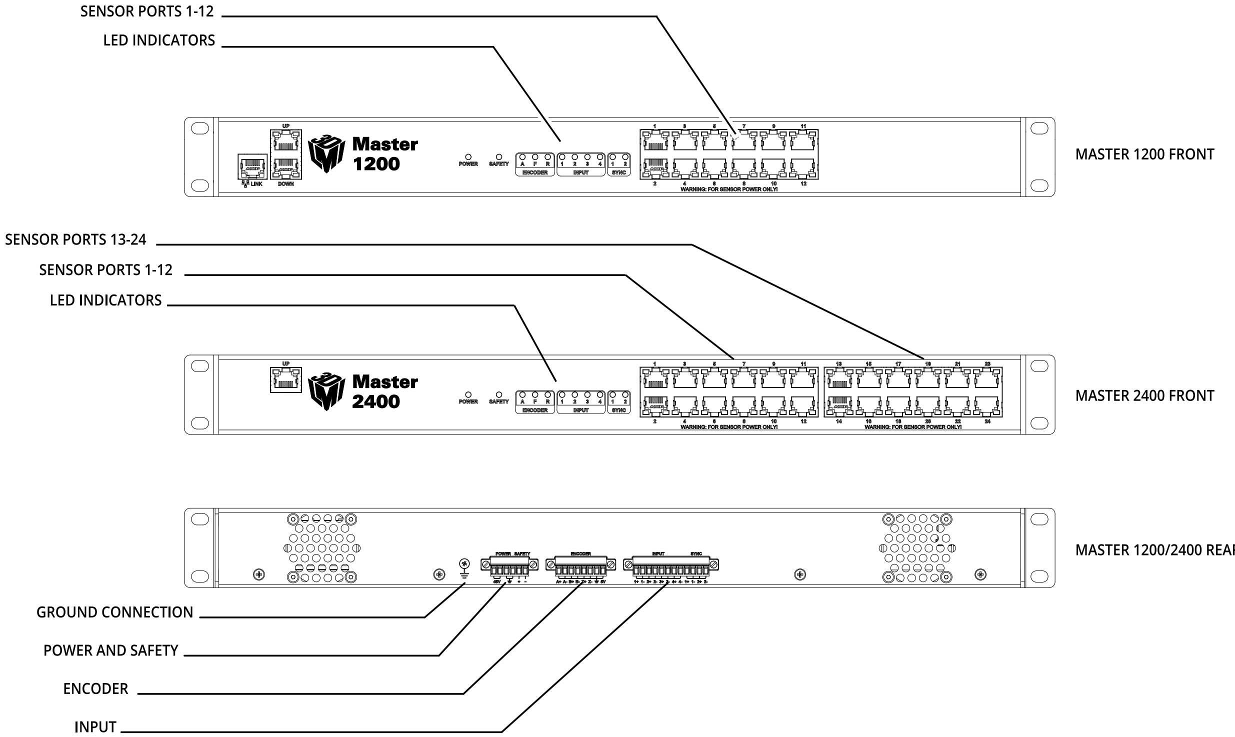
See *Master 100* on page 900 for pinout details.

Master 400 / 800 / 1200 / 2400

The Master 400, 800, 1200, and 2400 network controllers let you connect more than two sensors:

- Master 400: accepts four sensors
- Master 800 accepts eight sensors
- Master 1200: accepts twelve sensors
- Master 2400: accepts twenty-four sensors

Master 400 and 800



Master 1200 and 2400

Item	Description
Sensor Ports	Master connection for sensors (no specific order required).
Ground Connection	Earth ground connection point.
Power and Safety	Power and safety connections. Safety input is *not* required with snapshot sensors.
Encoder	Accepts encoder signal.
Input	Accepts digital input.

For pinout details for Master 400 or 800, see *Master 400/800* on page 902.

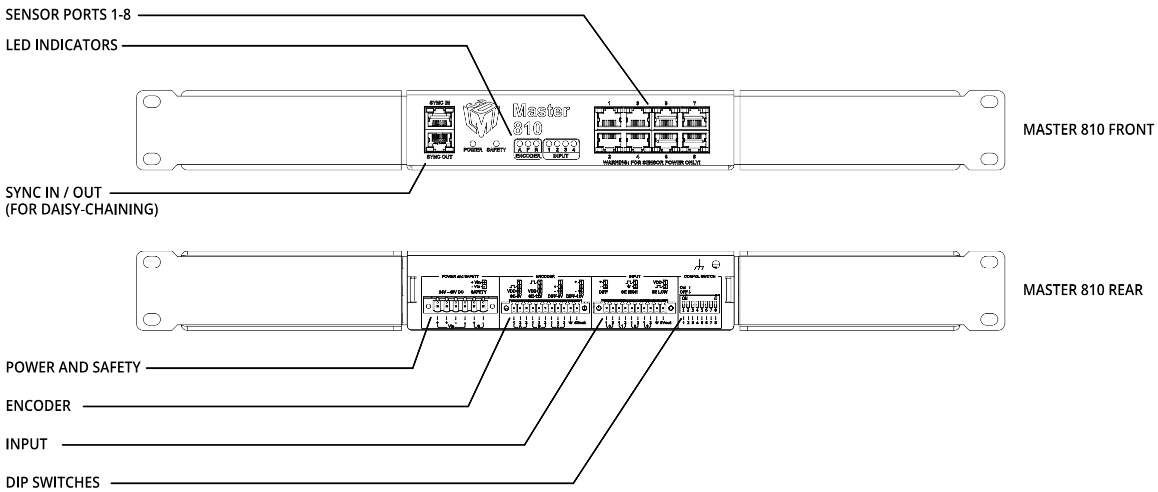
For pinout details for Master 1200 or 2400, see *Master 1200/2400* on page 915.

Master 810 / 2410

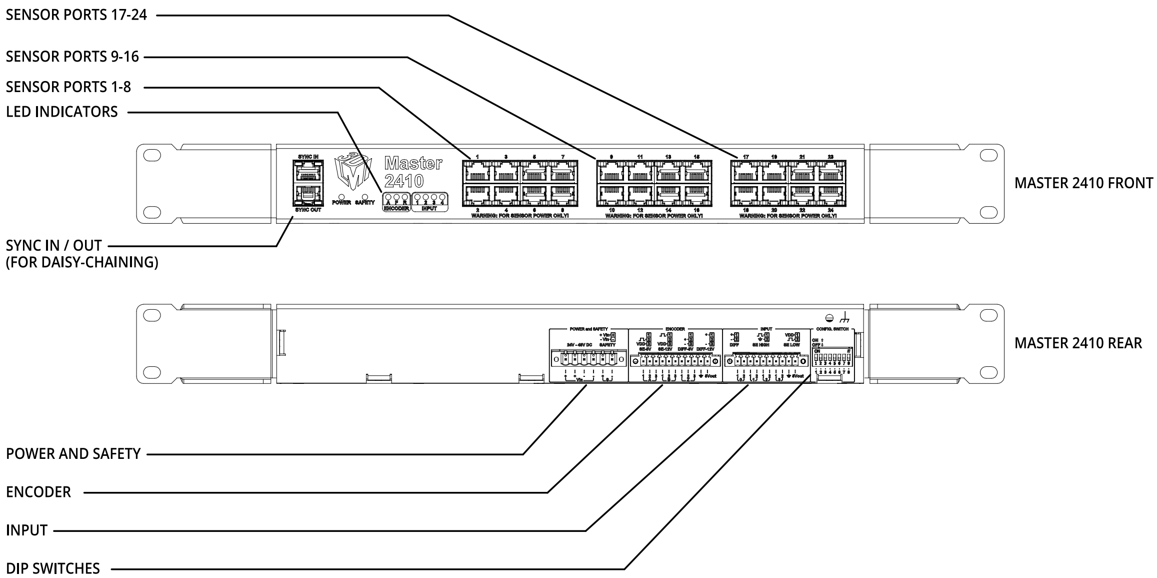
The Master 810 and 2410 network controllers let you connect multiple sensors to create a multi-sensor system:

- Master 810 accepts up to eight sensors
- Master 2410 accepts up to twenty-four sensors

Both models let you divide the quadrature frequency of a connected encoder to make the frequency compatible with the Master, and also set the debounce period to accommodate faster encoders. For more information, see *Configuring Master 810* on page 32. (Earlier revisions of these models lack the DIP switches.)



Master 810



Master 2410

Item	**Description**
Sensor Ports	Master connection for sensors (no specific order required).
Power and Safety	Power and safety connections. Safety input is *not* required with snapshot sensors.
Encoder	Accepts encoder signal.
Input	Accepts digital input.
DIP Switches	Configures the Master (for example, allowing the device to work with faster encoders). For information on configuring Master 810 and 2410 using the DIP switches, see *Configuring Master 810* on page 32.
LED Indicators	For more information, see *Master 810/2410* on page 905.

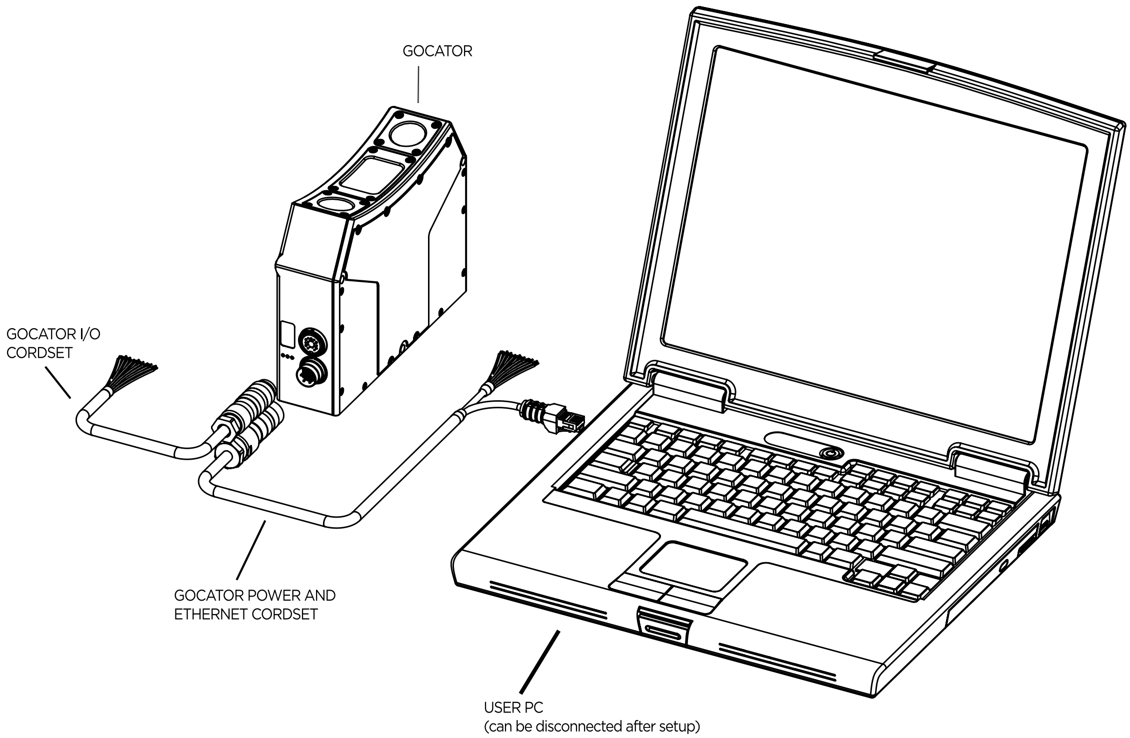
For pinout details, see *Master 810/2410* on page 905.

System Overview

Gocator sensors can be installed and used in scenarios where the target to be scanned is static in position relative to the sensor for the short duration of camera exposure. Sensors can be connected as standalone devices or in a multi-sensor system.

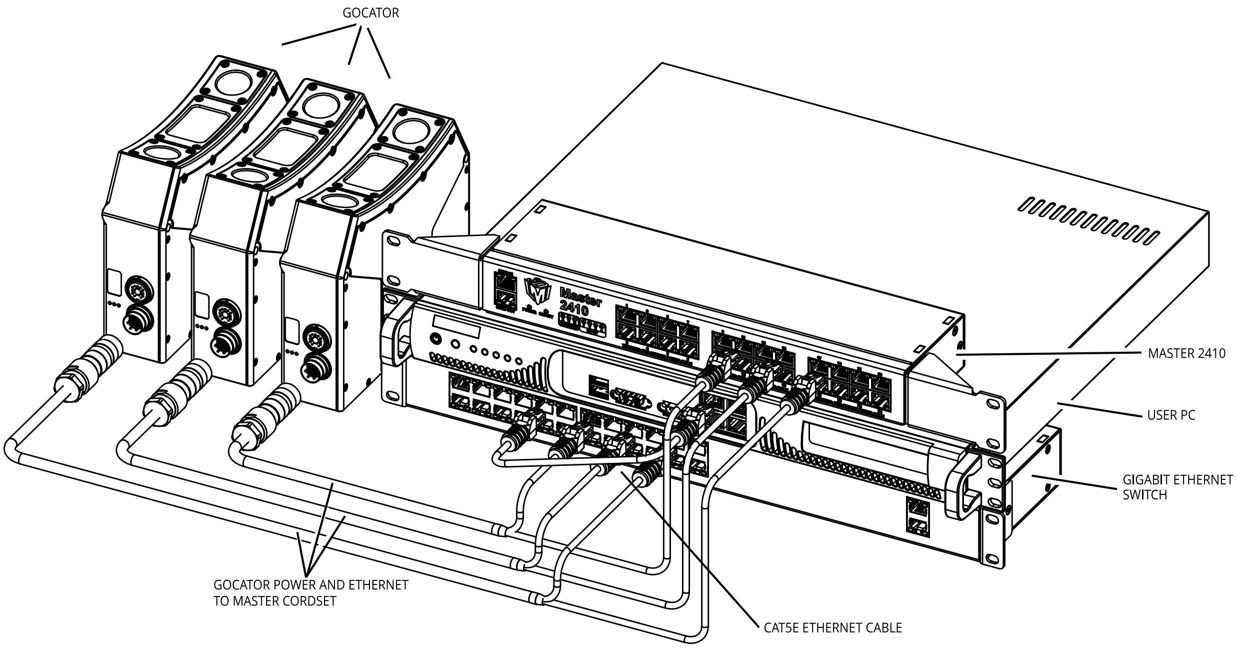
Standalone System

Standalone systems are typically used when only a single sensor is required. The device can be connected to a computer's Ethernet port for setup and can also be connected to devices such as encoders, photocells, or PLCs.



Multi-Sensor System

A Master network controller (excluding Master 100) can be used to connect two or more sensors into a multi-sensor system. Master cordsets are used to connect the sensors to a Master. The Master provides a single point of connection for power, safety, encoder, and digital inputs. A Master 400/800/810/1200/2400/2410 can be used to ensure that the scan timing is precisely synchronized across sensors. Sensors and client computers communicate via an Ethernet switch (1 Gigabit/s recommended).


GOCATOR
MASTER 2410
USER PC
GIGABIT ETHERNET
SWITCH
GOCATOR POWER AND ETHERNET
TO MASTER CORDSET
CAT5E ETHERNET CABLE

Installation

The following sections provide grounding, mounting, and orientation information.

Gocator 3210: 24 VDC Operation and Cordset Length

Although LMI recommends operating Gocator 3210 at 48 VDC to fully exploit the sensor's capabilities, starting with Gocator firmware 4.7, Gocator 3210 can operate when connected to a 24 VDC power supply. This may sometimes be necessary, for example, because of local regulations.

When running the sensor with a 24 VDC power supply, because of limitations in the sensor and in cordsets, the projector intensity is reduced. As a result, when running a 3210 sensor at 24 VDC, you may need to increase exposure time with darker targets to compensate for the reduced projector intensity. For information on setting exposure, as well as the impact of running the sensor at 24 VDC on projector intensity, see *Exposure* on page 96.

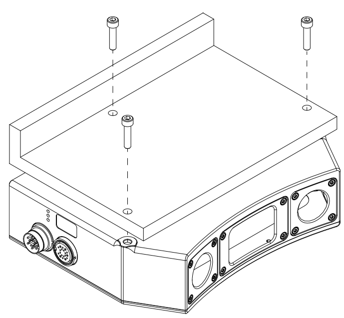
Note that because projector intensity decreases with longer cordsets (caused by the increased resistance and higher current), if limiting exposure time is important, try to limit the length of the cordset.

Note that if you run a 3210 sensor on 24 VDC, you must specify this voltage and the cordset length in the web interface. For more information, see *Networking and Power* on page 76.

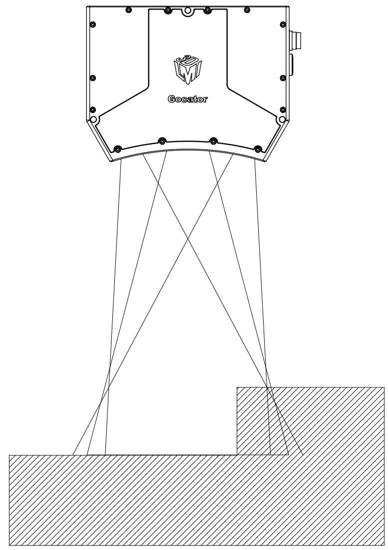
Mounting

Sensors should be mounted using a model-dependent number of screws. Some models also provide the option to mount using bolts in through-body holes. Refer to the dimension drawings of the sensors in *Specifications* on page 880 for the appropriate screw diameter, pitch, and length, and bolt hole diameter.

⚠	Proper care should be taken in order to ensure that the internal threads are not damaged from cross-threading or improper insertion of screws.



Sensors should not be installed near objects that might occlude a camera's view of the projected light.



⚠	The sensor must be heat sunk through the frame it is mounted to. When a sensor is properly heat sunk, the difference between ambient temperature and the temperature reported in the sensor's health channel is less than 15° C.

⚠	Gocator sensors are high-accuracy devices. The temperature of all of its components must be in equilibrium. When the sensor is powered up, a warm-up time of at least one hour is required to reach a consistent spread of temperature within the sensor.

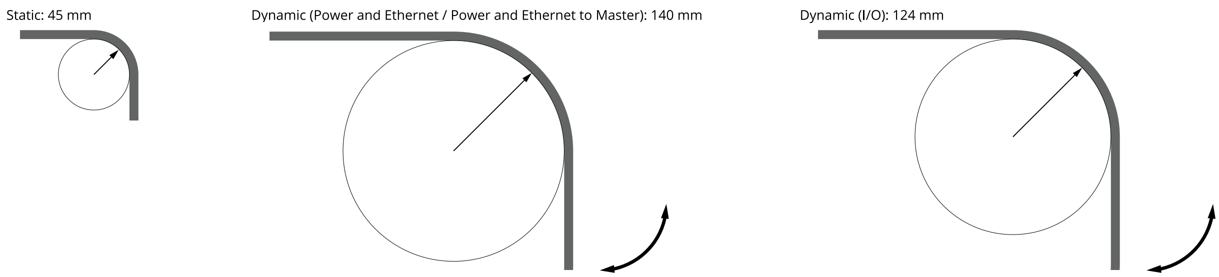
Cordset Bend Radius Limits

Limit bends in cordsets based on which type you are using (standard or high flex). The following table lists the bend limits, as well as number of bends performed in LMI's testing.

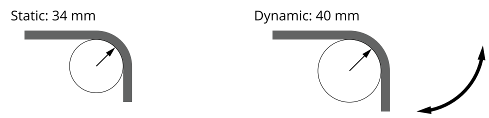
In the table below, part numbers do not include the length indicator.

Bend Radius Limits

Cable Type	Cordset	Part Numbers	Outer Diameter	Static (mm)	Dynamic (mm)	Number of Ticks (90° Tick Tock)	Number of Ticks (U-shape)
Standard	Power and Ethernet to Master Power and Ethernet	30858, 30877, 30861, 30880	8.89	45	140	> 2,000,000	
	I/O	30864,30883	8	45	124	> 2,000,000	
High Flex	Power and Ethernet to Master Power and Ethernet	301165, 301173, 301176, 301171	8.5	34	40	> 2,000,000	> 7,000,000
	I/O	301175,301172	9	34	40	> 2,000,000	> 7,000,000

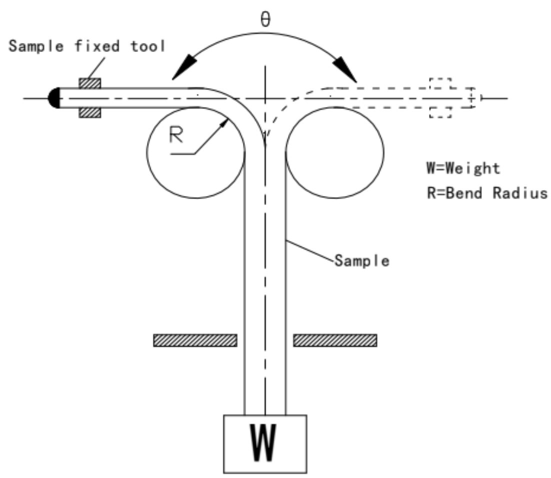


Standard cordset bend radius limits

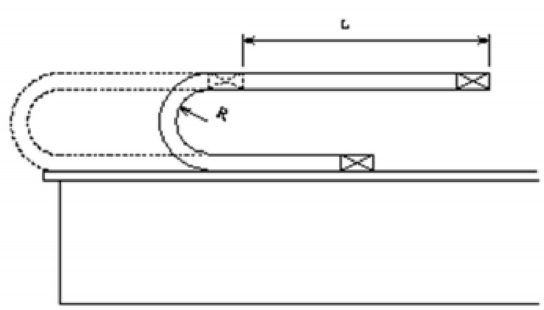


High flex cordset bend radius limits

The following illustrations show the test setups used to determine the number of bends.



Tick-tock test setup (θ = 180°)



U-shape test setup (L = 500 mm).

For cordset part numbers, see *Accessories* on page 918.

For more information on cordsets, see *Gocator 3x00 Cordsets* on page 18.

Grounding

Components of a sensor system should be properly grounded.

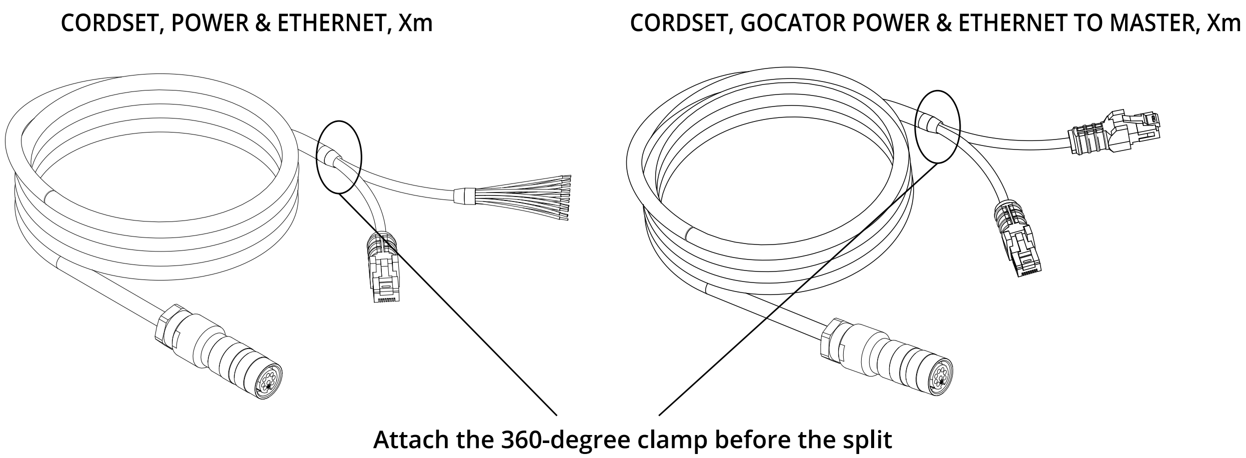
Gocator

Gocator sensors should be grounded to the earth/chassis through their housings and through the grounding shield of the Power I/O cordset. Sensors have been designed to provide adequate grounding through their mounting screws. Always check grounding with a multi-meter to ensure electrical continuity between the mounting frame and the sensor's connectors.

The frame or electrical cabinet that the sensor is mounted to must be connected to earth ground.

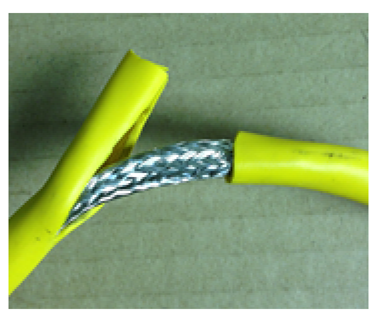
Recommended Practices for Cordsets

If you need to minimize interference with other equipment, you can ground the Power & Ethernet or the Power & Ethernet to Master cordset (depending on which cordset you are using) by terminating the shield of the cordset before the split. The most effective grounding method is to use a 360-degree clamp.



To terminate the cordset's shield:

1. Expose the cordset's braided shield by cutting the plastic jacket before the point where the cordset splits.



2. Install a 360-degree ground clamp.

Master Network Controllers

The rack mount brackets provided with all Masters are designed to provide adequate grounding through the use of star washers. Always check grounding with a multi-meter by ensuring electrical continuity between the mounting frame and RJ45 connectors on the front.

	When using the rack mount brackets, you *must* connect the frame or electrical cabinet to which the Master is mounted to earth ground.

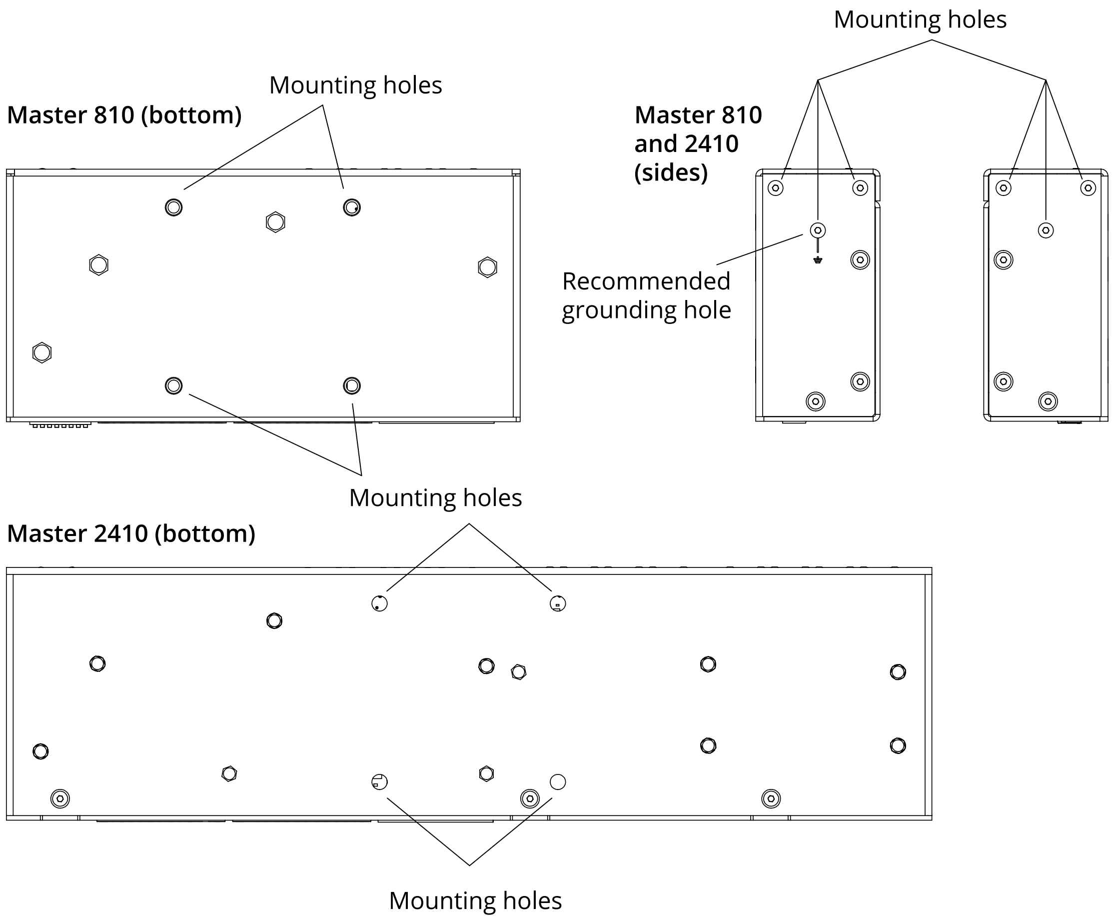
	You *must* check electrical continuity between the mounting frame and RJ45 connectors on the front using a multi-meter.

If you are mounting Master 810 or 2410 using the provided DIN rail mount adapters, you must ground the Master directly; for more information, see *Grounding When Using a DIN Rail (Master 810/2410)* below.

Grounding When Using a DIN Rail (Master 810/2410)

If you are using DIN rail adapters instead of the rack mount brackets, you must ensure that the Master is properly grounded by connecting a ground cable to one of the holes indicated below. The holes on the bottom of the unit accept M4 screws. The holes on the sides of the unit accept M3 screws.

	You can use any of the holes shown below. However, LMI recommends using the holes indicated on the housing by a ground symbol.



An additional ground hole is provided on the rear of Master 810 and 2410 network controllers, indicated by a ground symbol.

Additional Grounding Schemes

Potential differences and noise in a system caused by grounding issues can sometimes cause sensors to reset or otherwise behave erratically. If you experience such issues, see the *Gocator Grounding Guide* (https://downloads.lmi3d.com/gocator-grounding-guide) in the Download center for additional grounding schemes.

Installing DIN Rail Clips: Master 810 or 2410

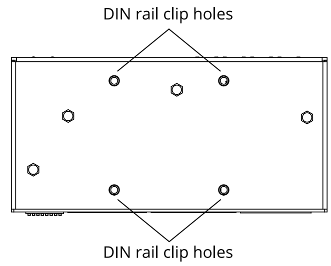
You can mount the Master 810 and 2410 using the included DIN rail mounting clips with M4x8 flat socket cap screws. The following DIN rail clips (DINM12-RC) are included:

Older revisions of Master 810 and 2410 network controllers use a different configuration for the DIN rail clip holes.

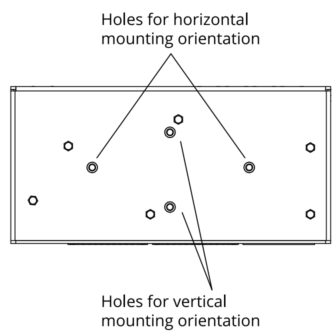
To install the DIN rail clips:

1. Remove the 1U rack mount brackets.
2. Locate the DIN rail mounting holes on the back of the Master (see below).

   Master 810:

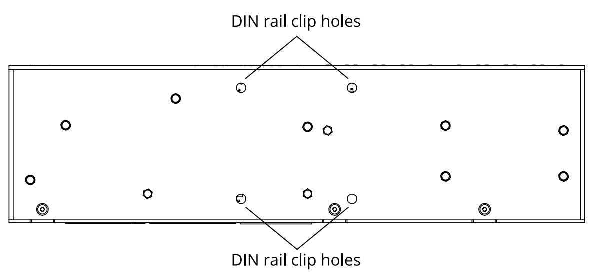


Current revision

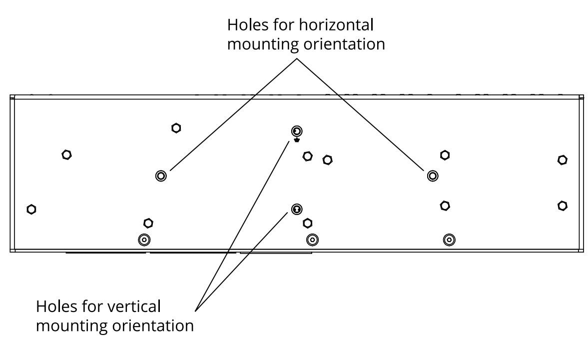


Older revision

   Master 2410:



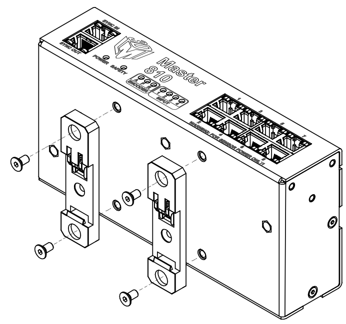
Current revision



Older revision

3. Attach the two DIN rail mount clips to the back of the Master using two M4x8 flat socket cap screws for each one.

   The following illustration shows the installation of clips on a Master 810 (current revision) for horizontal mounting:



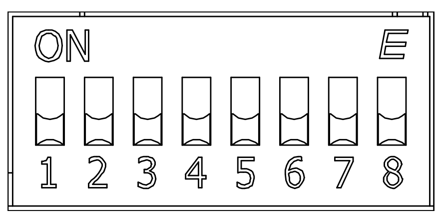
Ensure that there is enough clearance around the Master for cabling.

Configuring Master 810

If you are using Master 810 with an encoder that runs at a quadrature frequency higher than 300 kHz, you must use the device's divider DIP switches to limit the incoming frequency to 300 kHz.

Master 810 supports up to a maximum incoming encoder quadrature frequency of 6.5 MHz.

The DIP switches are located on the rear of the device.



Switches 5 to 8 are reserved for future use.

This section describes how to set the DIP switches on Master 810 to do the following:

- Set the divider so that the quadrature frequency of the connected encoder is compatible with the Master.
- Set the debounce period to accommodate faster encoders.

Setting the Divider

To set the divider, you use switches 1 to 3. To determine which divider to use, use the following formula:

Output Quadrature Frequency = Input Quadrature Frequency / Divider

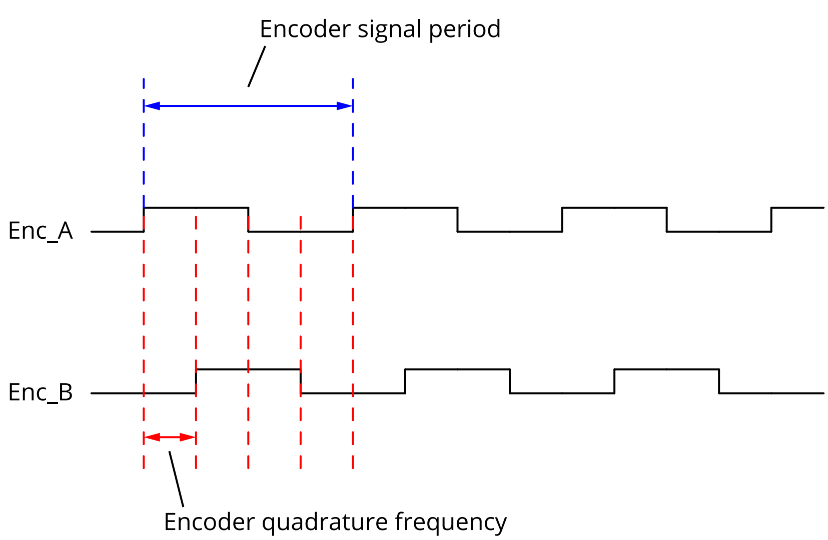
In the formula, use the *quadrature frequency* of the encoder (for more information, see *Encoder Quadrature Frequency* on the next page) and a divider from the following table so that the Output Quadrature Frequency is no more than 300 kHz.

Divider	Switch 1	Switch 2	Switch 3
1	OFF	OFF	OFF
2	ON	OFF	OFF
4	OFF	ON	OFF
8	ON	ON	OFF
16	OFF	OFF	ON
32	ON	OFF	ON
64	OFF	ON	ON
128	ON	ON	ON

> The divider works on debounced encoder signals. For more information, see *Setting the Debounce Period* below.

Encoder Quadrature Frequency

Encoder quadrature frequency is defined as illustrated in the following diagram. It is the frequency of encoder ticks. This may also be referred as the native encoder rate.



You must use a quadrature frequency when determining which divider to use (see *Setting the Divider* on the previous page). Consult the datasheet of the encoder you are using to determine its quadrature frequency.

> Some encoders may be specified in terms of encoder signal frequency (or period). In this case, convert the signal frequency to quadrature frequency by multiplying the signal frequency by 4.

Setting the Debounce Period

If the quadrature frequency of the encoder you are using is greater than 3 MHz, you must set the debounce period to “short.” Otherwise, set the debounce period to “long.”

You use switch 4 to set the debounce period.

Debounce period	Switch 4
short debounce	ON
long debounce	OFF

Network Setup

The following sections provide procedures for client PC and sensor network setup.

> DHCP is not recommended for sensors. If you choose to use DHCP, the DHCP server should try to preserve IP addresses. Ideally, you should use static IP address assignment (by MAC address) to do this.

> The following sections refer to using the sensor's web interface. For important information on browser compatibility, see *Browser Compatibility* on page 61.

Client Setup

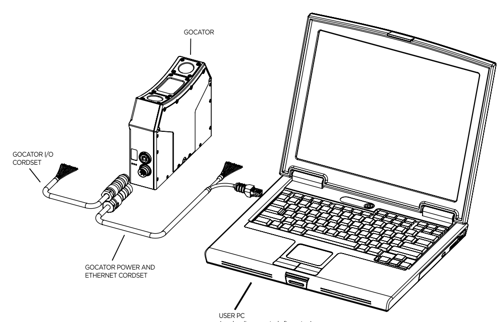
To connect to a sensor from a client PC, you must ensure the client's network card is properly configured.

Sensors are shipped with the following default network configuration:

Setting	Default
DHCP	Disabled
IP Address	192.168.1.10
Subnet Mask	255.255.255.0
Gateway	0.0.0.0

To connect to a sensor for the first time:

1. Connect cables and apply power.
   Sensor cabling is illustrated in *System Overview* on page 23.



2. Change the client PC's network settings.

   Windows 7

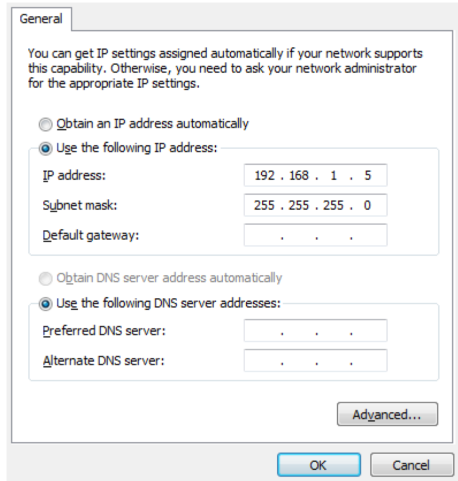
   a. Open the Control Panel, select **Network and Sharing Center**, and then click **Change Adapter Settings**.

   b. Right-click the network connection you want to modify, and then click **Properties**.

   c. On the **Networking** tab, click **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**.

   d. Select the **Use the following IP address** option.

   e. Enter IP Address "192.168.1.5" and Subnet Mask "255.255.255.0", then click **OK**.

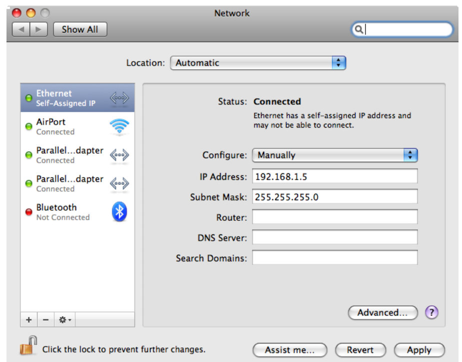


   Mac OS X v10.6

   a. Open the Network pane in **System Preferences** and select **Ethernet**.

   b. Set **Configure** to **Manually**.

   c. Enter IP Address "192.168.1.5" and Subnet Mask "255.255.255.0", then click **Apply**.



⚠ See *Troubleshooting* on page 879 if you experience any problems while attempting to establish a connection to the sensor.

Gocator Setup

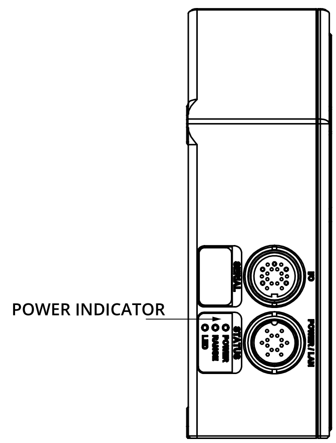
The Gocator is shipped with a default configuration that will produce 3D data for most targets.

The following describes how to set up a sensor system for operations. After you have completed the setup, you can perform a scan to verify basic sensor operation.

Running a Standalone Sensor System

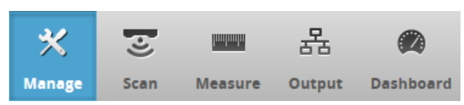
To configure a standalone sensor system:

1. Power up the sensor.
   The power indicator (blue) should turn on immediately.



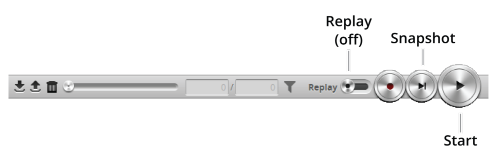
2. Enter the sensor's IP address (192.168.1.10) in a web browser.
   The sensor interface loads.
   If a password has been set, you will be prompted to provide it and then log in.

3. Go to the **Manage** page.



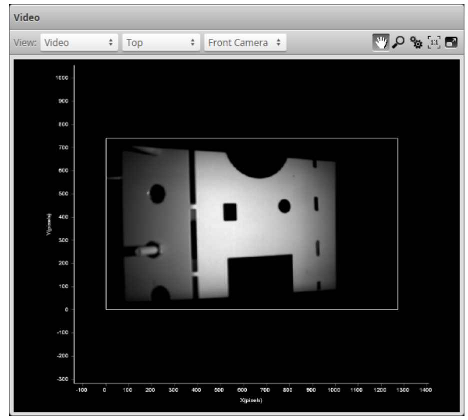
4. Ensure that Replay mode is off (the slider is set to the left).

Replay mode disables measurements.



5. Go to the **Scan** page.
6. Observe the profile in the data viewer
7. Press the **Start** button or the **Snapshot** on the **Toolbar** to start the sensor.
   The **Start** button is used to run sensors continuously.
   The **Snapshot** button is used to trigger the capture of a single frame.

8. Move a target into the sensor's projected light.
   If a target object is within the sensor's measurement range, the data viewer will display scan data, and the sensor's range indicator will illuminate.
   If no scan data is displayed in the data viewer, see *Troubleshooting* on page 879.



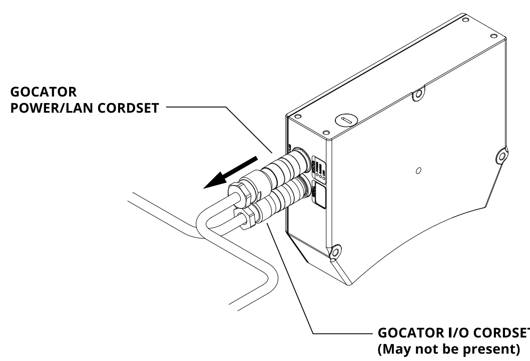
9. Press the **Stop** button.
   The projected light should turn off.

Running a Multi-Sensor System

All sensors are shipped with a default IP address of 192.168.1.10. Ethernet networks require a unique IP address for each device, so you must set up a unique address for each sensor.

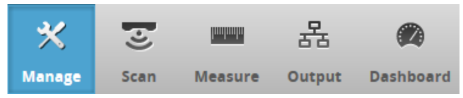
To configure a multi-sensor system:

1. Configure the first sensor in the system as described in *Running a Standalone Sensor System* on the previous page.
2. Unplug the Power/LAN connection of the configured sensor to power it down.

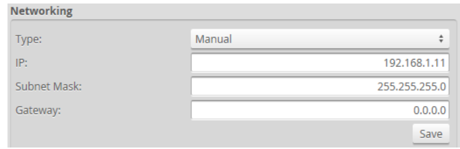


3. Connect the Power/LAN cordset to the new sensor.
   The power LED (blue) of the new sensor should turn on immediately.
4. Enter the new sensor's default IP address (192.168.1.10) in a web browser.
   The web interface loads.

5. Go to the **Manage** page.



6. Modify the IP address in the **Networking** category and click the **Save** button.
   You should increment the last octet of the IP address for each additional sensor you need to use. For example, if the IP address of the first sensor you configured is 192.168.1.10, use 192.168.1.11 for the second sensor; use 192.168.1.12 for the third; etc.
   When you click the **Save** button, you will be prompted to confirm your selection.



7. Power-cycle or reset the sensor.
   After changing a sensor's network configuration, the sensor must be reset or power-cycled before the change will take effect.
8. Repeat steps 3 to 7 for each additional sensor.

Required Ports

The following table lists the ports used by sensors, the Ethernet-based protocols, the SDK, and the PC-based accelerator. Use this information to determine whether you need to open ports on your network and to understand the traffic that a sensor system will produce over a network.

Ports used

Port	Data Packet Protocol	Description
80	TCP	Server for sensor web interface
502	TCP	Modbus protocol communication
2016	UDP	Internal (protocol-independent)
2017	TCP	Internal (protocol-independent)
2018	TCP	Internal (protocol-independent)
2019	TCP	Internal (protocol-independent)
2020	UDP	Gocator protocol discovery; SDK; accelerator
3189	TCP	Flash security policy server (only in Gocator 4.7 and earlier releases)
3190	TCP	Gocator protocol control channel; SDK; accelerator
3191	TCP	Emulator web port
3192	TCP	Gocator protocol upgrade channel; SDK; accelerator
3194	TCP	Gocator protocol health channel; SDK; accelerator
3195	TCP	Gocator protocol private data
3196	TCP	Gocator protocol discovery; SDK; accelerator

Port	Data Packet Protocol	Description
3197	UDP	Emulator scenario management (RPC)
3220	UDP	Gocator protocol discovery; SDK; accelerator
8190	TCP	ASCII protocol
44818	TCP	EtherNet/IP protocol (standard port)
44818	UDP	EtherNet/IP protocol (standard port)

For more information on how the different protocols use these ports, see the appropriate section in *Protocols* on page 650.

Next Steps

After you complete the steps in this section, the sensor system is ready to be configured for an application using the software interface. The interface is explained in the following sections:

Management and Maintenance (page 75)
Contains settings for sensor system layout, network, motion and alignment, handling jobs, and sensor maintenance.

Scan Setup and Alignment (page 87)
Contains settings for scan mode, trigger source, detailed sensor configuration, and performing alignment.

Models (page 126)
Contains settings for creating part matching models and sections.

Measurement and Processing (page 146)
Contains built-in measurement tools and their settings.

Output (page 511)
Contains settings for configuring output protocols used to communicate measurements to external devices.

Dashboard (page 524)
Provides monitoring of measurement statistics and sensor health.

Toolbar (page 64)
Controls sensor operation, manages jobs, and replays recorded measurement data.

How Gocator Works

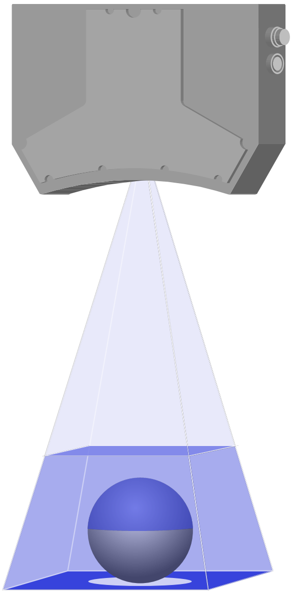
The following sections provide an overview of how Gocator acquires and produces data, detects and measures parts, and controls devices such as PLCs. Some of these concepts are important for understanding how you should mount sensors and configure settings such as active area.

You can use the Accelerator to speed up processing of data. For more information, see *Gocator Acceleration* on page 529.

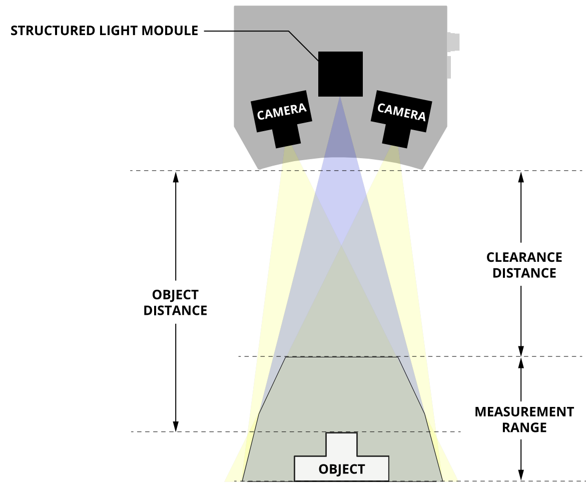
3D Acquisition

After a sensor system has been set up and is running, it is ready to start capturing 3D data.

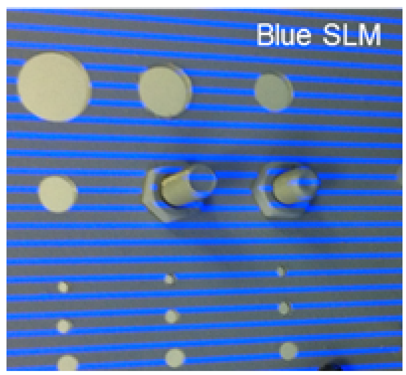
3x00 series sensors are 3D snapshot sensors, meaning they capture an entire surface in 3D in a single snapshot. These sensors project several structured light patterns in a rapid sequence onto the target. The reflection of the pattern off the target is captured by two cameras. The target must remain stationary during the camera exposure of the light patterns. The required exposure time depends on the shape, color, and reflectiveness of the target, but is often shorter than 1 second.



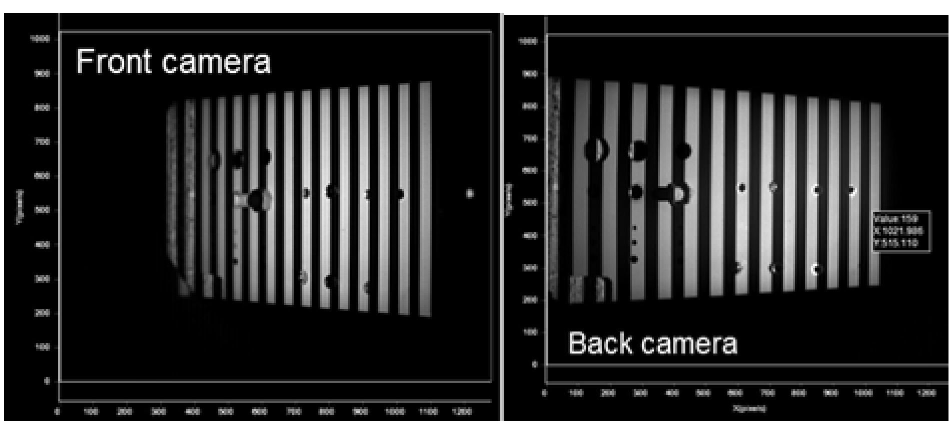
A Structured Light Modulator (SLM) produces a sequence of high resolution/high contrast light patterns using a blue LED. Two cameras capture the reflected light pattern from different viewing angles. The sensor can then use either stereo correlation or independent triangulation to generate 3D points from the light pattern.



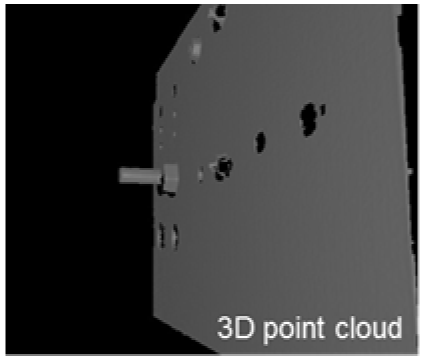
These are the steps of the acquisition of a 3D point cloud:



Step 1: Light pattern projected on target



Step 2: Reflected light captured by two cameras



Step 3: Use stereo correlation or independent triangulation to generate 3D point cloud

> Gocator sensors are always pre-calibrated to deliver 3D data in engineering units throughout their measurement range.

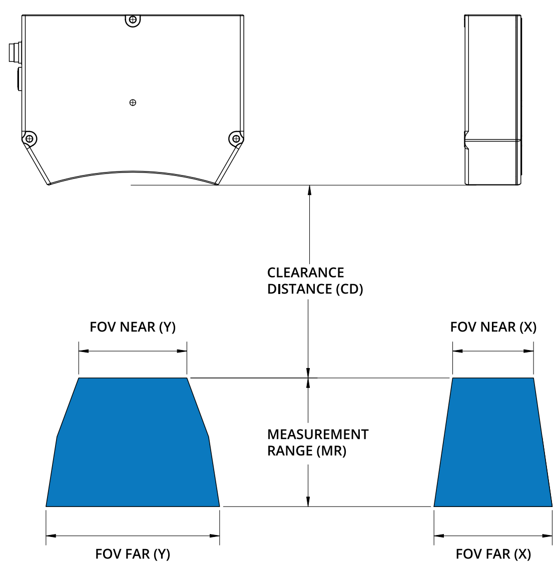
Clearance Distance, Field of View and Measurement Range

Clearance distance (CD), field of view (FOV), and measurement range (MR) are important concepts for understanding the setup of a sensor and for understanding results.

Clearance distance – The minimum distance from the sensor that a target can be scanned and measured. A target closer than this distance will result in invalid data.

Measurement range – The vertical distance, starting at the end of the clearance distance, in which targets can be scanned and measured. Targets beyond the measurement range will result in invalid data.

Field of view – The area on the XY plane along the measurement range. At the far end of the measurement range, the field of view is larger, but the resolution is lower. At the near end, the field of view is smaller, but the resolution is higher. When resolution is critical, if possible, place the target closer to the near end.



Stereo Correlation vs. Independent Triangulation

Stereo correlation means that the sensor locates the same point on the physical target in the two images captured at different viewing angles. Since the exact distance between the two cameras and the viewing angles are known, the distance to the point can be calculated. In order for stereo correlation to work and produce a 3D data point, the point on the target must be visible in both cameras. Stereo acquisition may produce more stable measurements on targets with a simple shape, but will be affected by occlusions on targets with complicated shapes and protruding features.

Independent triangulation means that each camera independently triangulates off the LED light pattern, based on the calibration process that takes place when the sensor is manufactured. Since a snapshot sensor has two cameras, a point on the physical target only needs to be visible to one of the cameras in order to generate a 3D point. Independent triangulation may improve performance on targets with complicated shapes that can cause occlusion, but it relies on the sensor's internal components being fully stable.

The **Reduce Occlusion** setting determines whether 3D data is acquired by using stereo correlation or both stereo correlation and independent triangulation. See *Reduce Occlusion* on page 93 for more details.

3D Data Output

Gocator measures the shape of the object calculated from either dual triangulation or stereo correlation. The sensor reports a series of 3D coordinates from the surface of the target in the sensor's field of view.

Coordinate Systems

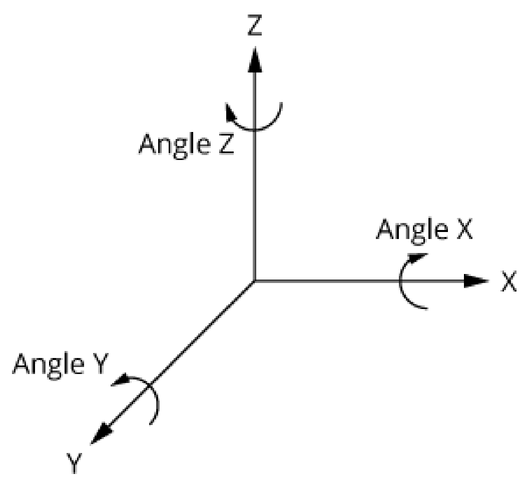
Data points are reported in one of three coordinate systems, which generally depends on the alignment state of the sensor.

- **Sensor coordinates**: Used on unaligned sensors.
- **System coordinates**: Used on aligned sensors.
- **Part and section coordinates**: Data can optionally be reported using a coordinate system relative to the part itself.

Understanding coordinate systems is an important part of understanding measurement results. These coordinate systems are described below.

> For all Gocator sensors, Y increases moving from the camera to the laser; for more information, see the coordinate system orientations illustrated in the specification drawings of your sensor in *Sensors* on page 880.

Gocator 3x00 sensors use Cartesian left-hand notation for defining 3D coordinates.



The Z axis represents the sensor's measurement range (MR), where the values increase toward the sensor. The X axis and Y axis represent the sensor's field of view (FOV).

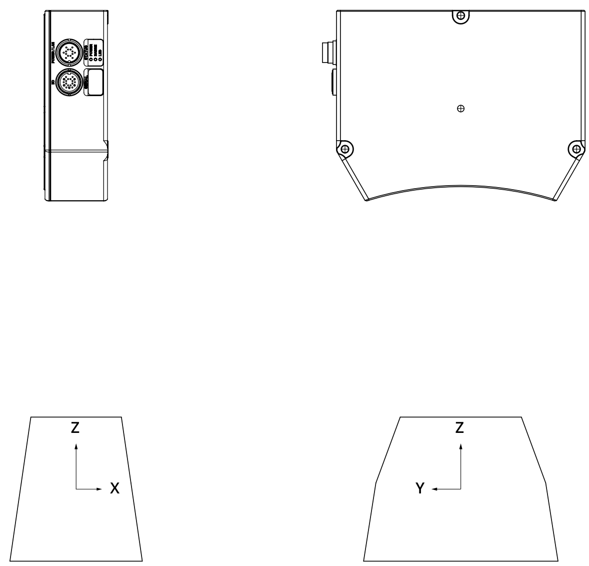
X offset, Y offset, and Z offset define the translations from the origin.

Rotations are specified based on rotating the target around the X axis (Angle X), followed by rotating around the Y axis (Angle Y), followed by rotating around the Z axis.

Sensor Coordinates

Unaligned sensors use *sensor* coordinates. The measurement range (MR) is along the Z axis. Most importantly, the origin is at the *center* of the measurement range and field of view, in other words, the center of the scan volume.

Before alignment, the origin of the sensor is at the center of the sensor's measurement range (MR) and field of view (FOV).



System Coordinates

Understanding system coordinates is important for two reasons. First, they are the direct result of performing the built-in alignment procedure. Second, they change how scan data is represented and how measurement results should be interpreted.

Performing the built-in alignment procedure on sensors adjusts the coordinate system in relation to sensor coordinates, resulting in *system coordinates* (for more information on sensor coordinates, see *Sensor Coordinates* above). For more information on aligning sensors, see *Alignment* on page 106.

The adjustments resulting from alignment are called *transformations* (offsets along the axes and rotations around the axes). Transformations are displayed in the **Sensor** panel on the **Scan** page. For more information on transformations in the web interface, see *Transformations* on page 95.

Alignment is used with a single sensor to compensate for mounting misalignment and to set a zero reference, such as a conveyor belt surface.

Y angle is positive when rotating from positive X to positive Z axis.

X angle is positive when rotating from positive Y to positive Z. Z angle is positive when rotating from positive X to positive Y.

Alignment can be used to establish a transformed coordinate system according to the user's needs. Alignment determines the adjustments to X, Y, and Z, as well as rotation angle around each axis.

Transformed coordinate systems can be associated with specific sensor jobs. For details, see *Alignment Types* on page 106.

When applying the transformations, the object is first rotated around X, then Y, and then Z, and then the offsets are applied.

Part and Section Coordinates

When you work with parts or sections extracted from scan data, a different coordinate system is available.

Part data can be expressed in aligned system coordinates or unaligned sensor coordinates. But part data can also be represented in *part coordinates*: data and measurement results are in a coordinate system that places the X and Y origins at the center of the part. The Z origin is at the surface surrounding the alignment target (if the sensor or system has been aligned) or in the center of the center of the measurement range (if the sensor or system has not been aligned).

> The **Frame of Reference** setting, in the **Part Detection** panel on the **Scan** page, controls whether part data is recorded using sensor/system coordinates or part coordinates.

Sections are *always* represented in a coordinate system similar to part coordinates: the X origin is always at the center of the extracted profile, and the Z origin is at the bottom of the alignment target (or in the center of the measurement range if the sensor is unaligned).

Switching between Coordinate Systems

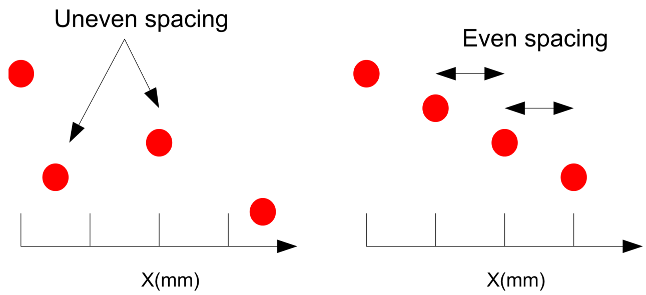
In many situations, when working with part data that has been recorded with **Frame of Reference** set to **Part**, it is useful to have access to the "real-world" coordinates, rather than part-relative coordinates. Sensors provide special "global" measurements, in the Bounding Box tools, that you can use in scripts to convert from part coordinates to sensor/system coordinates. Note that the same applies to sections.

For more information, see the Profile Bounding Box tool or the Surface Bounding Box tool, and the Script tool.

Resampling

The internal acquisition engine in the sensor produces a random 3D point cloud, where each individual point is a coordinate triplet (X, Y, Z). When the sensor's **Uniform Spacing** setting is disabled, the sensor returns this point cloud, However, in this scenario, no built-in measurement tools are available for use with the data, so you must implement measurements yourself. (For more information on this setting, see *Scan Modes* on page 88.)

When the sensor's **Uniform Spacing** setting is enabled, the random 3D point cloud is resampled to an even grid in the XY plane. The resampling divides the XY plane into fixed-size square "bins." 3D points are projected along the Z axis, perpendicular to the XY plane, and points that fall into the same bin will be combined into a single Z value. The size of the resampling bins can be set with the **X/Y Spacing Interval** setting; for details, see *Spacing Interval* on page 103. The XY resampling plane is established through the sensor's built-in alignment routine. That is, the resampling plane is set to match the plane described by the calibration plate; for more information, see *System Coordinates* on the previous page.



In the Ethernet data channel, only the Z values are reported, and the X and Y positions can be reconstructed through the 2D-array index at the receiving end (the client). Resampling reduces the complexity for the algorithms in the sensor's built-in measurement tools, allowing them to run on the embedded processors. All built-in measurement tools in the sensor operate on resampled data in Surface mode.

Data Generation and Processing

After scanning a target, a sensor can process the scan data to allow the use of more sophisticated measurement tools. This section describes the following concepts:

- Part detection
- Sectioning

Part Detection

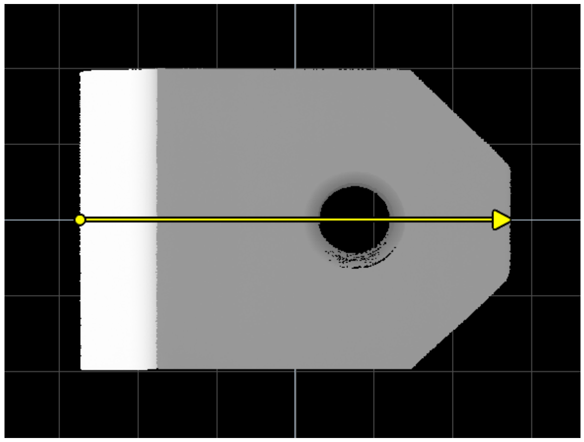
A sensor can isolate discrete parts on a surface into separate scans representing parts.

Gocator can then perform measurements on these isolated parts.

For more information on part detection, see *Part Detection* on page 111.

Sectioning

In Surface mode, the sensor can also extract a profile from a surface or part using a line you define on that surface or part. The resulting profile is called a “section.” A section can have any orientation on the surface, but its profile is parallel to the Z axis.



You can use most of Gocator's profile measurement tools on a section, letting you perform measurements that are not possible with surface measurement tools.

For more information on sections, see *Sections* on page 140.

Part Matching

The sensor can match scanned parts to the edges of a model based on a previously scanned part (see *Using Edge Detection* on page 127) or to the dimensions of a fitted bounding box or ellipse that encapsulate the model (see *Using Bounding Box and Ellipse* on page 136). When parts match, the sensor can rotate scans so that they are all oriented in the same way. This allows measurement tools to be applied consistently to parts, regardless of the orientation of the part you are trying to match.

Measurement

After Gocator scans a target and, optionally, further processes the data, the sensor is ready to take measurements on the scan data.

Gocator provides several measurement tools, each of which provides a set of individual measurements, giving you dozens of measurements ideal for a wide variety of applications to choose from. The configured measurements start returning pass/fail decisions, as well as the actual measured values, which are then sent over the enabled output channels to control devices such as PLCs, which can in turn control ejection or sorting mechanisms. (For more information on measurements and configuring measurements, see *Measurement and Processing* on page 146. For more information on output channels, see *Output and Digital Tracking* on page 59.)

> You can create custom tools that run your own algorithms. For more information, see *GDK* on page 847.

A part's position can vary on a transport system. To compensate for this variation, Gocator can anchor a measurement to the positional measurement (X, Y, or Z) or Z angle of an easily detectable feature, such as the edge of a part. The calculated offset between the two ensures that the anchored measurement will always be properly positioned on different parts.

Tool Chaining

Gocator's measurement and processing tools can be linked together: one tool uses another tool's output as input. This gives you a great deal of control and flexibility when it comes to implementing your application.

The following table lists the available outputs from Gocator's tools:

Gocator tool outputs

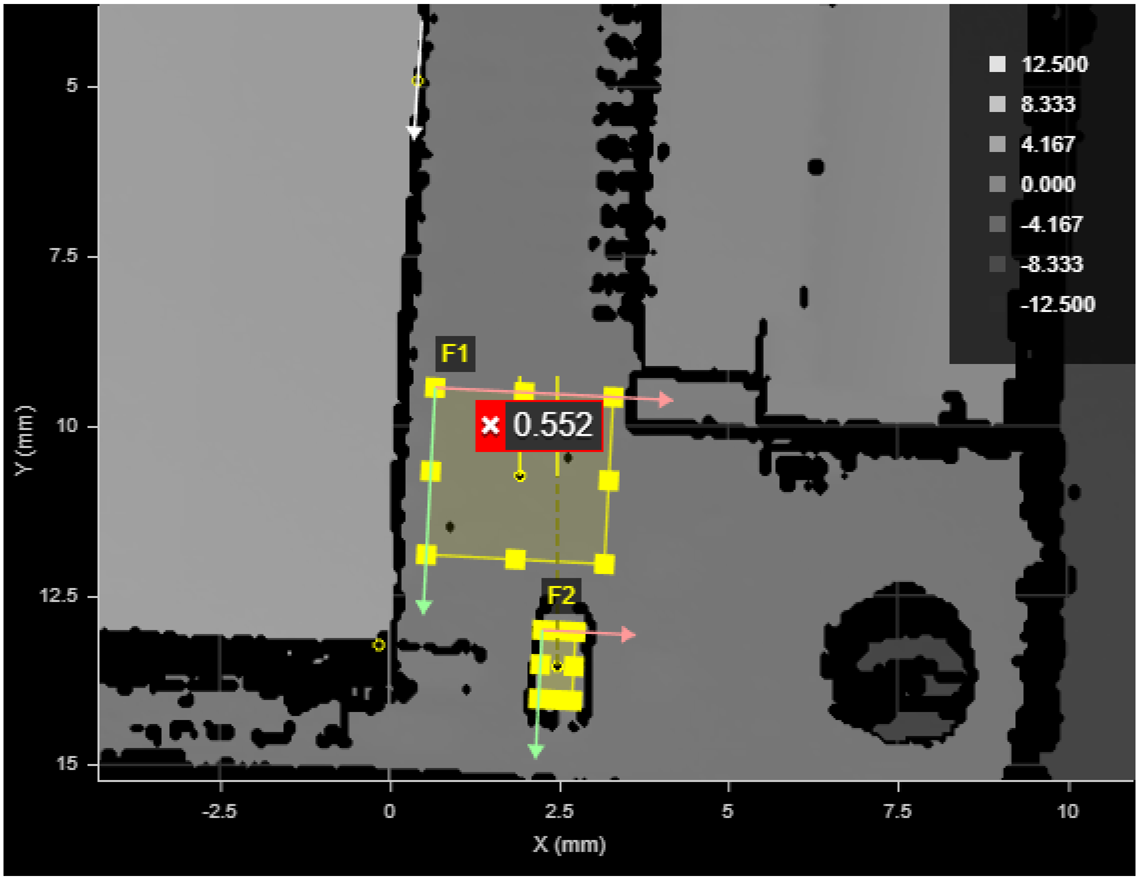
	Data Type	Supported Output Protocol	Visualization in Data Viewer	Input for Other Tools
Measurement	Single 64-bit value	SDK, PLC protocols	Rendered on tool's input data	Not supported as input, positional and Z angle measurements can be used by some tools for anchoring
Geometric Features	Structured data values: for example, point or line	Cannot be output via protocols	Rendered on tool's input data	Tools that accept the specific features
Tool Data	Binary data structure: Profile, Surface, or Generic	SDK	Rendered separately	Tools that accept the specific data type

The following sections describe these types of output and how you use them as input.

Anchoring Measurements

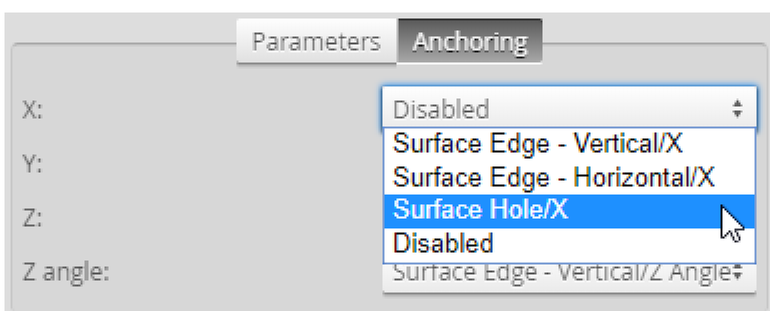
Tools can use the positional measurements (X, Y, or Z) of other tools as anchors to compensate for minor shifts of parts: anchored tools are "locked" to the positional measurements of the anchoring tool's measurements. Some tools can also use a Z Angle measurement as an anchor. Typically, you will use measurements from more easily found features on a target—such as an edge or a hole—as anchors to accurately place other positional and dimensional measurements. This can help improve repeatability and accuracy in the anchored tools. Note that anchoring measurements are used to calculate the offsets of the anchored tools: the results from these measurements are not used as part of the anchored tool's measurements.

Anchoring measurements are rendered as overlays on a tool's input data.



Height measurements rendered a tool's input: a small PCB component (F2) relative to nearby surface (F1), anchored to positional (X and Y) measurements of the hole (lower right) and to the Z angle of a larger component to the left (white arrow)

You enable anchoring on the **Anchoring** tab on the **Tools** panel:



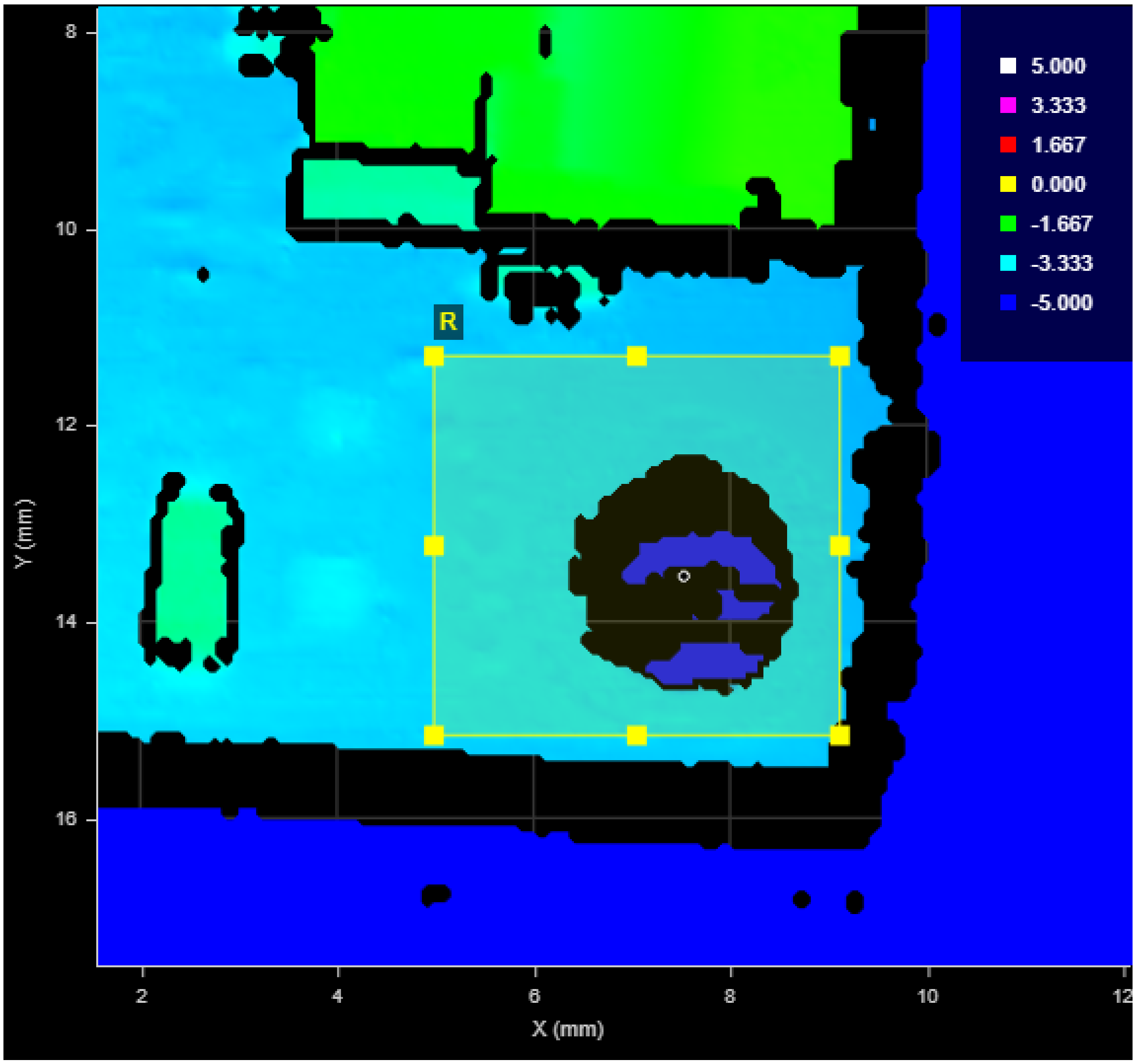
Note that anchoring is visualized on the *anchored* tool's input.

When combined with the matching and rotation capabilities of part matching, anchoring accounts for most sources of variation in part position and orientation and, consequently, avoids many measurement errors. For more information on anchoring, see *Measurement Anchoring* on page 169.

Geometric Features

Many of Gocator's measurement tools can output data structures such as points, lines, planes, and circles. These structures are called geometric features and contain the components you would expect: a point geometric feature contains X, Y, and Z components (representing the location of the point in 3D space). Examples of point geometric features output by Gocator's measurement tools are hole center points, the tip and base of studs, or a position on a surface.

Geometric features are rendered as overlays on a tool's input data.

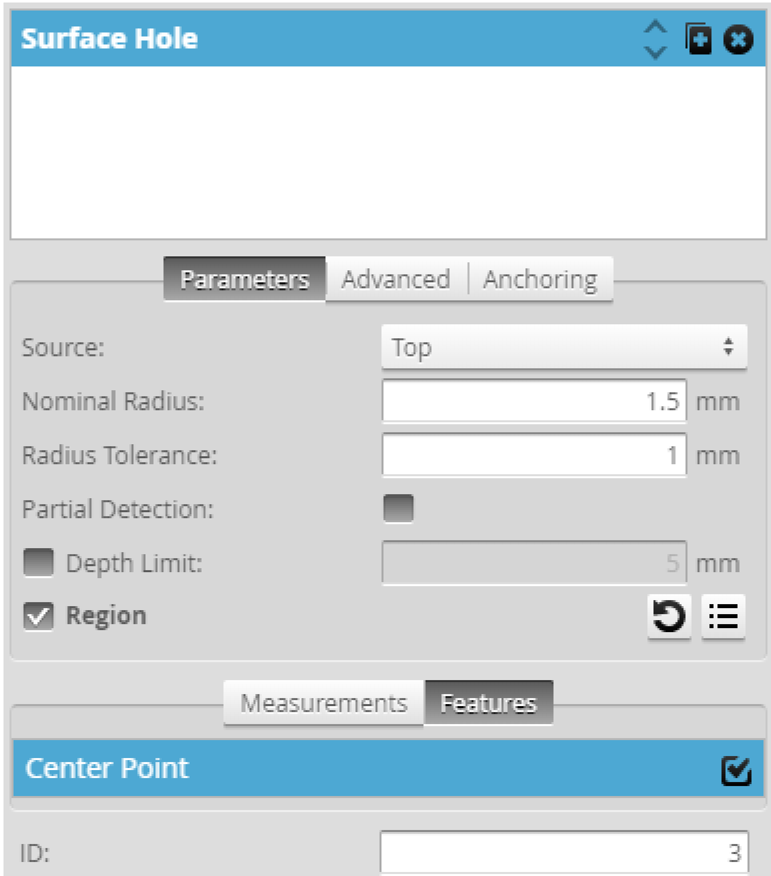


Point geometric feature (a hole's Center Point) rendered on a tool's input as a small white circle

Gocator's "Feature" tools (such as Feature Dimension and Feature Intersect) use geometric features as inputs. For example, because the point geometric feature representing the center of a hole has X, Y, and Z components, you can perform dimensional measurements between it and another geometric feature, such as another hole or an edge. The Feature Create tool takes one or more geometric features as input and generates *new* geometric features (for example, creating a line from two point geometric features).

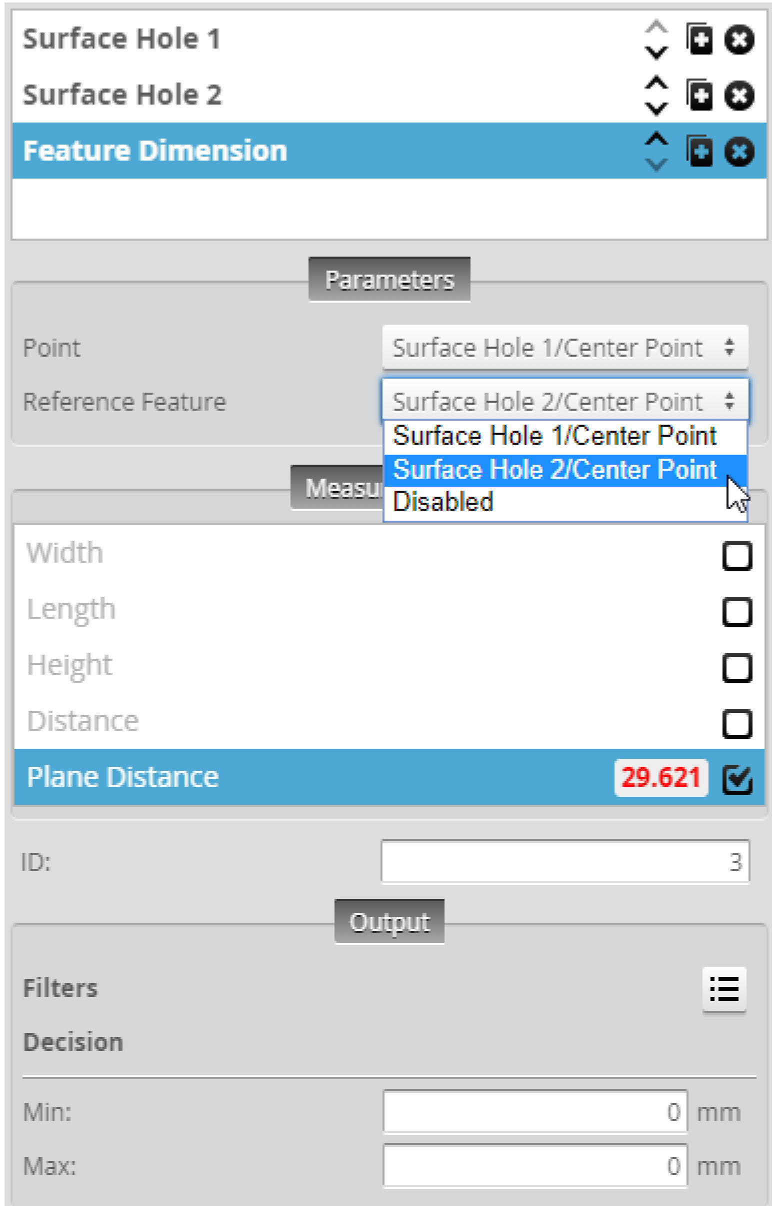
You can then perform measurements on those features directly in the tool or in other Feature measurement tools. You can also use angle measurements on the newly created features for anchoring. For more information on Feature tools, see *Feature Measurement* on page 484.

You enable geometric feature *output* on a tool's **Features** tab:



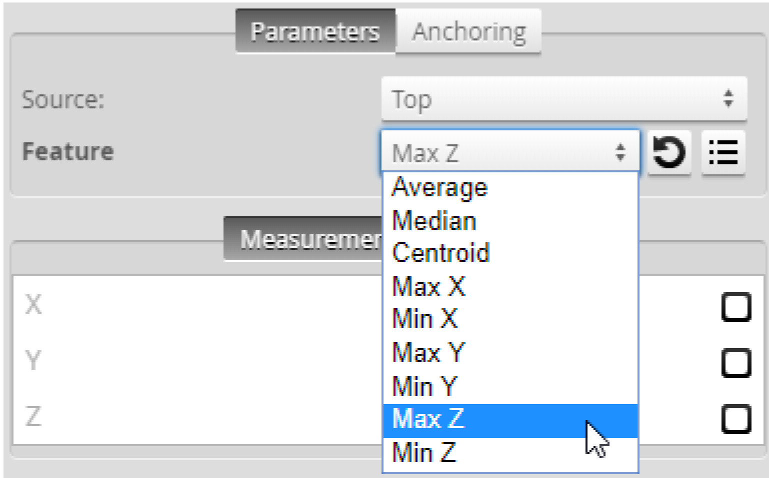
Center Point geometric feature of a Surface Hole tool enabled on Features tab

You enable geometric feature *inputs* on a Feature tool's **Parameters** tab:



Setting the Point and Reference Feature to the Center Point geometric features of two different holes

Geometric features are distinct from the "feature points" used by certain tools to determine which data point in a region should be used in a measurement, for example, the maximum versus the minimum on the Z axis of a data point in a region of interest:



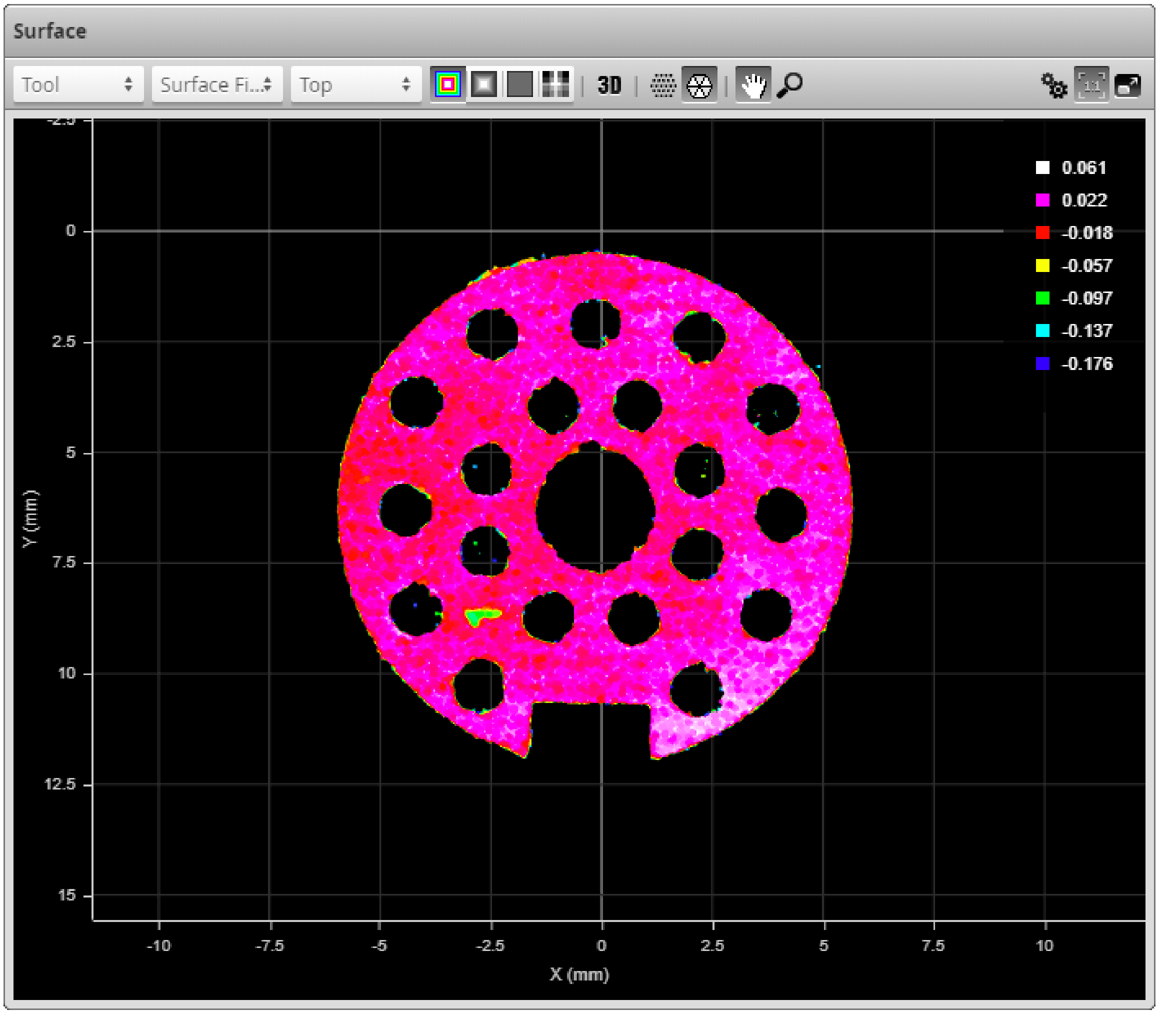
For more information on feature points, see *Feature Points* on page 162.

Tool Data

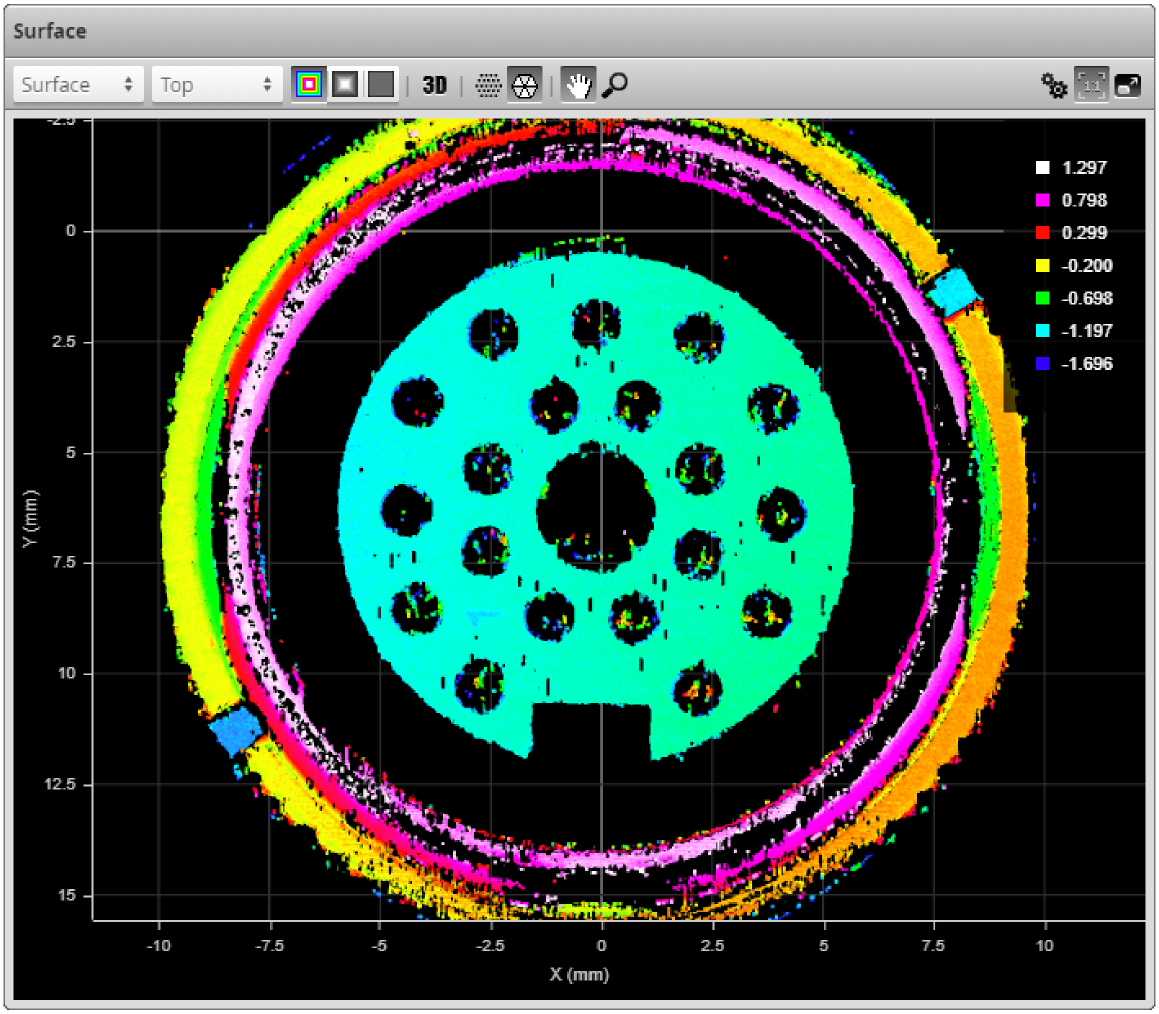
Some measurement and processing tools can output more complex data, which can be used as input by other tools or SDK applications. The following types of data are available: Profile, Surface, and Generic.

Profile and Surface tool data are identical in nature to the data produced by a sensor scan, except that they are the processed result from a tool. This kind of data can be used as input in compatible tools. Examples of this kind of this kind of data are the Stitched Surface output from the Surface Stitch tool, or the Filtered Surface output from the Surface Filter tool. Another important kind of data is the Transformed Surface produced by the Surface Transform tool, which transforms (shifting or rotating on the X, Y, and Z axes) the sensor's scan data; the Surface Transform tool supports a full 6 degrees of freedom. For more information, see *Transform* on page 465.

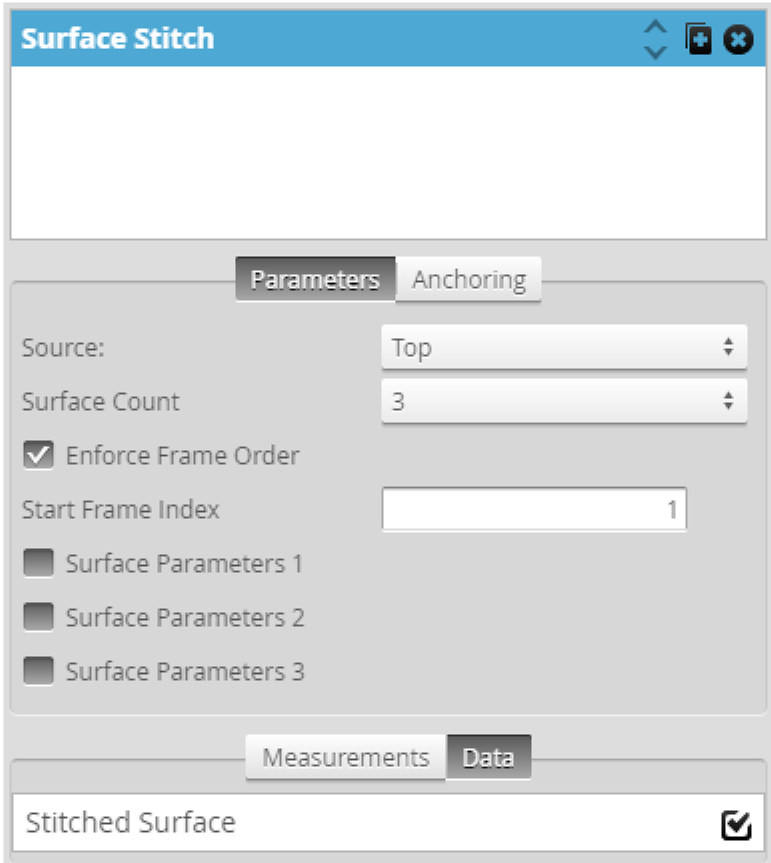
Both Profile and Surface tool data can be visualized in the data viewer, not as an overlay, however, but as independent data. The following is the output of the Surface Filter tool . Note that the first drop-down is set to Tool, to tell the sensor to display the tool data output, rather than the sensor output:



The following shows the scan data coming directly from the sensor's scan engine. Note that the first drop-down is set to **Surface**, rather than **Tool**.

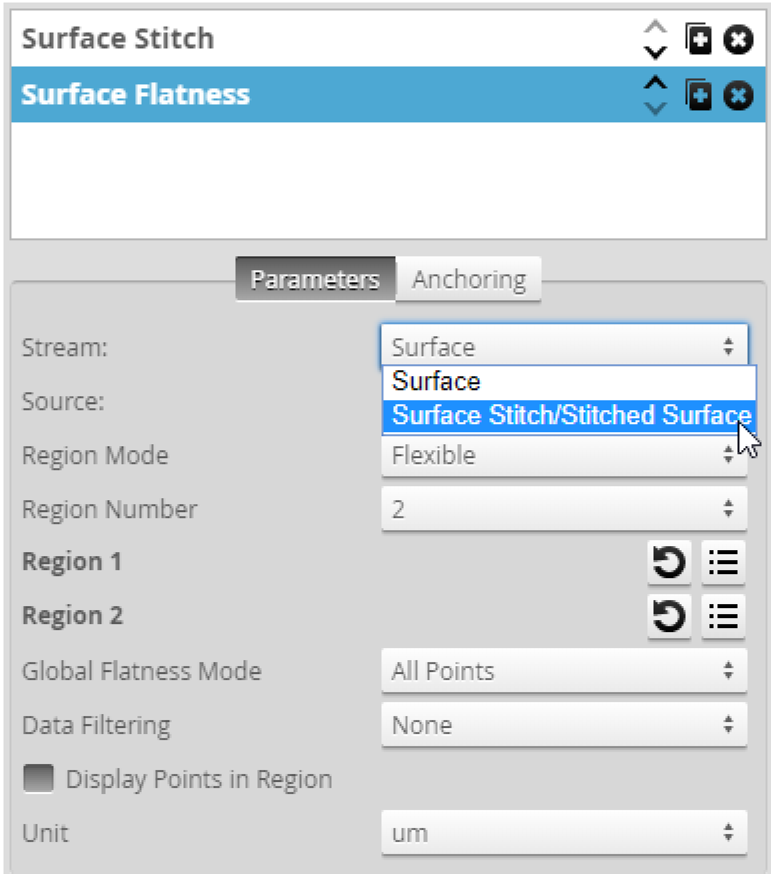


You enable this processed *output* in a tool's **Data** tab:



Stitched Surface tool enabled in Surface Stitch tool

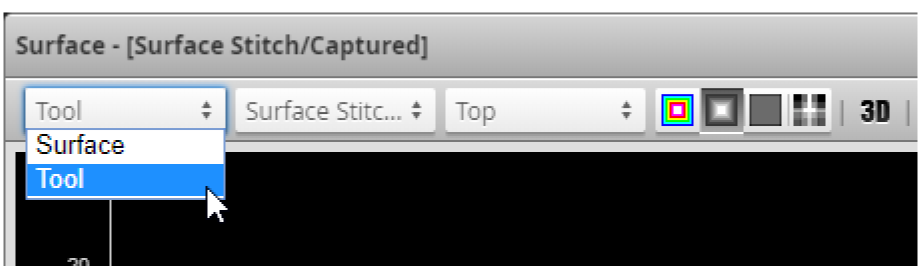
You enable tool data *input* on a tool's **Parameters** tab, using the **Stream** drop-down:



Setting a Surface Flatness tool's input to a Surface Stitch tool's data output

Generic tool data can't be visualized. It can however be accessed from GDK tools or SDK applications you create. Examples of Generic tool data are the Segments Array data produced by the Surface Segmentation tool, or the Output Measurement data produced by the Surface Flatness. For more information on the SDK, see *GoSDK* on page 836. Generic tool data is enabled in the same way as Profile and Surface tool data, from the tool's **Data** tab.

You may need to switch the first data viewer drop-down to "Tool" to view Profile or Surface tool data:



Output and Digital Tracking

After Gocator has scanned and measured parts, the last step in the operation flow is to output the results and/or measurements.

One of the main functions of Gocator sensors is to produce pass/fail decisions, and then control something based on that decision. Typically, this involves rejecting a part through an eject gate, but it can also involve making decisions on good, but different, parts. This is described as “output” in Gocator. Gocator supports the following output types:

- Ethernet (which provides industry-standard protocols such as Modbus, EtherNet/IP, and ASCII, in addition to the Gocator protocol)
- Digital
- Analog
- Serial interfaces

An important concept is digital output tracking. Production lines can place an ejection or sorting mechanism at different distances from where the sensor scans the target. For this reason, Gocator lets you schedule a delayed decision over the digital interfaces. Because the conveyor system on a typical production line will use an encoder or have a known, constant speed, targets can effectively be “tracked” or "tagged." Gocator will know when a defective part has traveled far enough and trigger a PLC to activate an ejection/sorting mechanism at the correct moment. For more information on digital output tracking, see *Digital Output* on page 516.

Gocator Web Interface

The following sections describe how to configure sensors using the Gocator web interface.

Browser Compatibility

LMI recommends Chrome, Firefox, or Edge for use with the Gocator web interface.

If you choose to use other browsers, please note the following limitations.

Internet Explorer 11 Switches to Software Rendering+

If you use sensors with large datasets on Internet Explorer 11, you may encounter the following issue.

If the PC connected to a sensor is busy, Internet Explorer may switch to software rendering after a specific amount of time. If this occurs, data is not displayed in the data viewer, and the only reliable way to recover from the situation is to restart the browser.

It is possible to remove the time limit that causes this issue, but you must modify the computer’s registry. To do so, follow Microsoft's instructions at https://support.microsoft.com/en-us/help/3099259/update-to-add-a-setting-to-disable-500-msec-time-limit-for-webgl-frame.

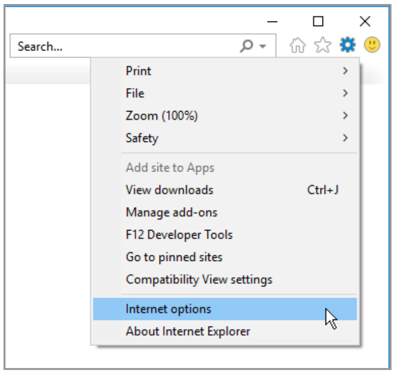
Internet Explorer 11 Displays "Out of Memory"

If you use sensors with large datasets on Internet Explorer 11, you may encounter the following issue.

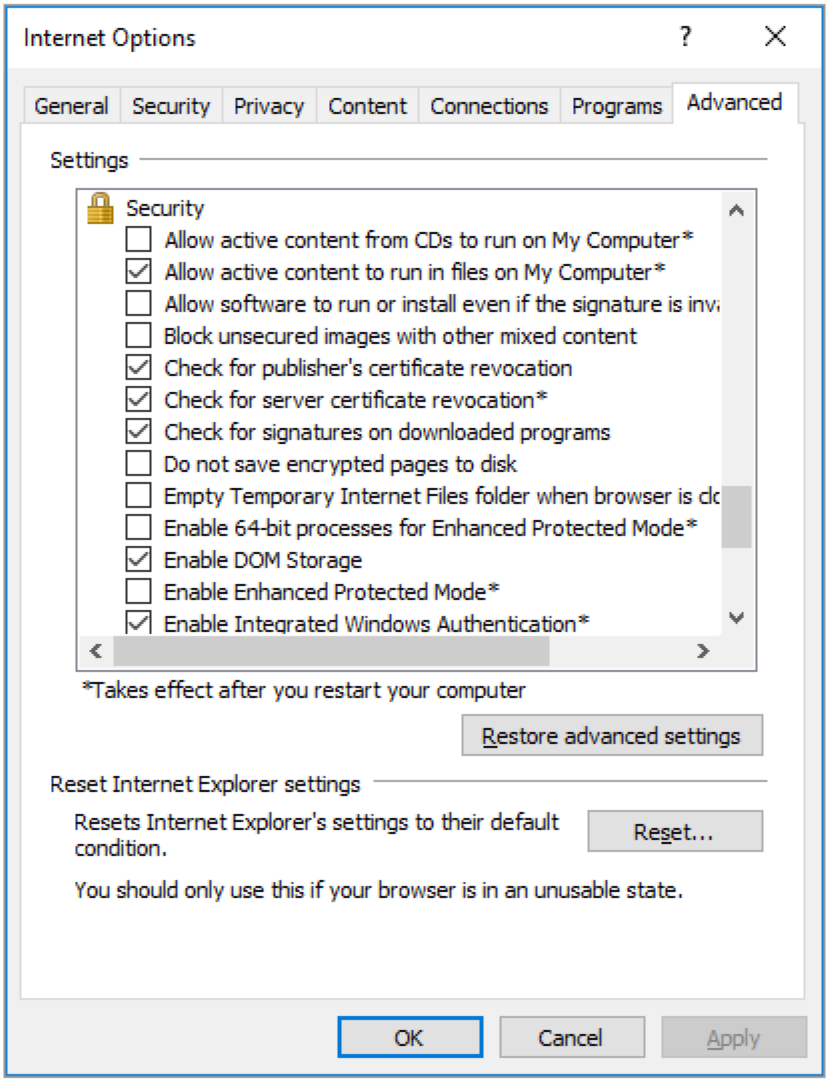
In some situations, you may encounter “Out of Memory” errors in the sensor's web interface. This issue can be resolved by checking two options in Internet Explorer.

To correct out of memory issues in Internet Explorer 11:

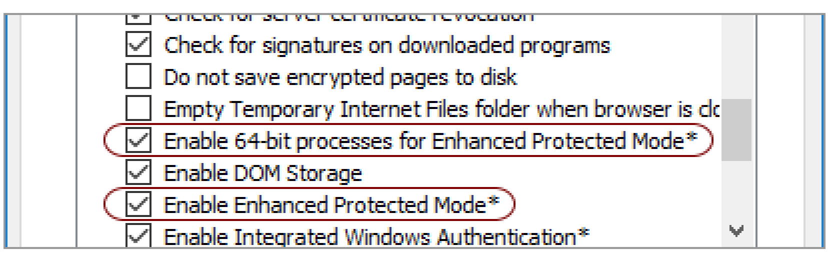
1. In upper right corner, click the settings icon ( ), and choose **Internet options**.



2. In the dialog that opens, click the **Advanced** tab, and scroll down to the **Security** section.



3. In the dialog, check *both* "Enable 64-bit processes for Enhanced Protected Mode" *and* "Enable Enhanced Protected Mode".



4. Click **OK** and then restart your computer for the changes to take effect.

Other Internet Explorer 11 Limitations

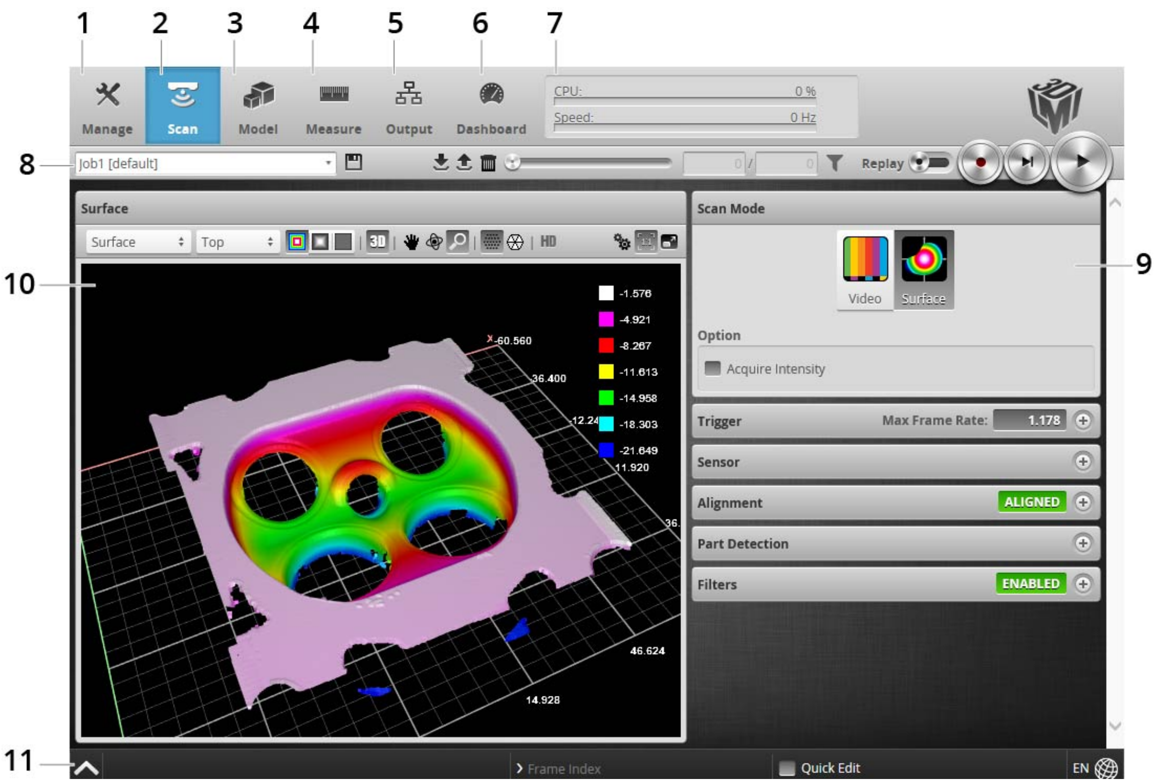
Drag-and-drop operations in the Tools Diagram panel are not supported in Internet Explorer 11 (for more information, see *Working with the Tools Diagram* on page 177).

You may also experience significant performance issues when using multiple data viewers in Internet Explorer 11 (for more information, see *Using Multiple Data Viewer Windows* on page 147).

User Interface Overview

Gocator sensors are configured by connecting to the IP address of a sensor with a web browser.

The web interface is shown below.

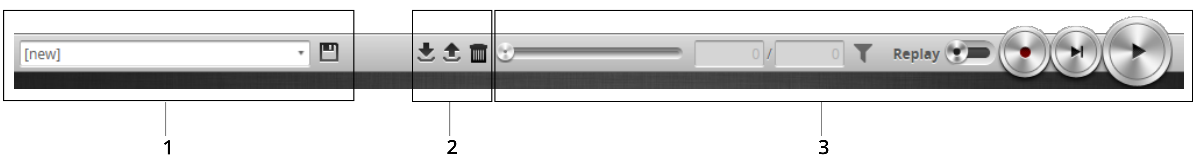


	Element	Description
1	Manage page	Contains settings for sensor system layout, network, motion and alignment, handling jobs, and sensor maintenance. See *Management and Maintenance* on page 75.
2	Scan page	Contains settings for scan mode, trigger source, detailed sensor configuration, and performing alignment. See *Scan Setup and Alignment* on page 87.
3	Model page	Lets you set up sections and part matching. See *Models* on page 126
4	Measure page	Contains built-in measurement tools and their settings. See *Measurement and Processing* on page 146.
5	Output page	Contains settings for configuring output protocols used to communicate measurements to external devices. See *Output* on page 511.

	Element	Description
6	Dashboard page	Provides monitoring of measurement statistics and sensor health. See *Dashboard* on page 524.
7	CPU Load and Speed	Provides important sensor performance metrics. See *Metrics Area* on page 72.
8	Toolbar	Controls sensor operation, manages jobs, and filters and replays recorded data. See *Toolbar* below.
9	Configuration area	Provides controls to configure scan and measurement tool settings.
10	Data viewer	Displays sensor data, tool setup controls, and measurements. See *Data Viewer* on page 115 for its use when the **Scan** page is active and on page 147 for its use when the **Measure** page is active.
11	Status bar	Displays log messages from the sensor (errors, warnings, and other information) and frame information, and lets you switch the interface language. For more information, see *Status Bar* on page 72.

Toolbar

The toolbar is used for performing operations such as managing jobs, working with replay data, and starting and stopping the sensor.



	Element	Description
1	Job controls	For saving and loading jobs.
2	Replay data controls	For downloading, uploading, and exporting recorded data.
3	Sensor operation / replay control	Use the sensor operation controls to start sensors, enable and filter recording, and control recorded data.

Creating, Saving and Loading Jobs (Settings)

A sensor can store several hundred jobs. Being able to switch between jobs is useful when a sensor is used with different constraints during separate production runs. For example, width decision minimum and maximum values might allow greater variation during one production run of a part, but might allow less variation during another production run, depending on the desired grade of the part.

Most of the settings that can be changed in the sensor's web interface, such as the ones in the **Manage**, **Measure**, and **Output** pages, are temporary until saved in a job file. Each sensor can have multiple job files. If there is a job file that is designated as the default, it will be loaded automatically when the sensor is reset.

When you change sensor settings using the sensor web interface in the emulator, some changes are saved automatically, while other changes are temporary until you save them manually. The following table lists the types of information that can be saved in a sensor.

Setting Type	Behavior
Job	Most of the settings that can be changed in the sensor's web interface, such as the ones in the **Manage**, **Measure**, and **Output** pages, are temporary until saved in a job file. Each sensor can have multiple job files. If there is a job file that is designated as the default, it will be loaded automatically when the sensor is reset.
Alignment	Alignment can either be fixed or dynamic, as controlled by the **Alignment Reference** setting in **Motion and Alignment** in the **Manage** page. Alignment is saved automatically at the end of the alignment procedure when **Alignment Reference** is set to **Fixed**. When **Alignment Reference** is set to **Dynamic**, however, you must manually save the job to save alignment.
Network Address	Network address changes are saved when you click the **Save** button in **Networking** on the **Manage** page. The sensor must be reset before changes take effect.

The job drop-down list in the toolbar shows the jobs stored in the sensor. The job that is currently active is listed at the top. The job name will be marked with "[unsaved]" to indicate any unsaved changes.

To create a job:

1. Choose **[New]** in the job drop-down list and type a name for the job.
2. Click the **Save** button or press **Enter** to save the job.
   The job is saved to sensor storage using the name you provided. Saving a job automatically sets it as the default, that is, the job loaded when then sensor is restarted.

To save a job:

- Click the **Save** button .

  The job is saved to sensor storage. Saving a job automatically sets it as the default, that is, the job loaded when then sensor is restarted.

To load (switch) jobs:

- Select an existing file name in the job drop-down list.

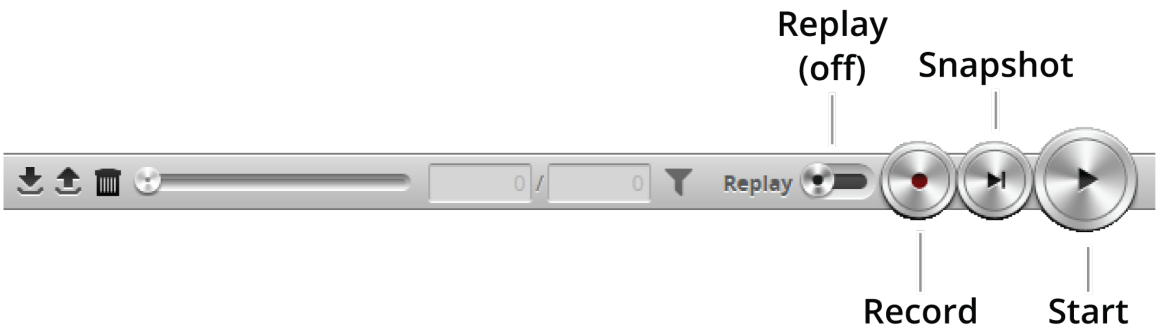
  The job is activated. If there are any unsaved changes in the current job, you will be asked whether you want to discard those changes.

You can perform other job management tasks—such as downloading job files from a sensor to a computer, uploading job files to a sensor from a computer, and so on—in the **Jobs** panel in the **Manage** page. See *Jobs* on page 78 for more information.

Recording, Playback, and Measurement Simulation

Sensors can record and replay recorded scan data, and also simulate measurement tools on recorded data. This feature is most often used for troubleshooting and fine-tuning measurements, but can also be helpful during setup.

Recording and playback are controlled using the toolbar controls.



Recording and playback controls when replay is off

To record live data:

1. Toggle **Replay** mode off by setting the slider to the left in the **Toolbar**.

	Replay mode disables measurements.

2. (Optional) Configure recording filtering.
   For more information on recording filtering, see *Recording Filtering* on page 68.

3. Click the **Record** button to enable recording.

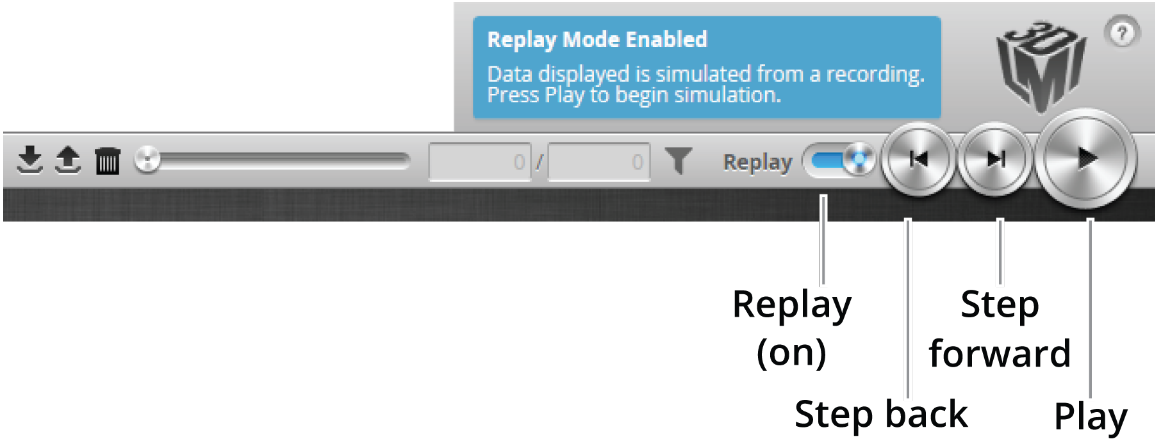
   

   The center of the Record button turns red.

   When recording is enabled (and replay is off), the sensor will store the most recent data as it runs. Remember to disable recording if you no longer want to record live data. (Press the **Record** button again to disable recording).

4. Press the **Snapshot** button or **Start** button.
   The **Snapshot** button records a single frame. The **Start** button will run the sensor continuously and all frames will be recorded, up to available memory. When the memory limit is reached, the oldest data will be discarded.

	Newly recorded data is appended to existing replay data unless the sensor job has been modified.



Playback controls when replay is on

To replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue and a Replay Mode Enabled message is displayed.

2. Use the **Replay** slider or the **Step Forward**, **Step Back**, or **Play** buttons to review data.
   The **Step Forward** and **Step Back** buttons move the current replay location forward and backward by a single frame, respectively.
   The **Play** button advances the replay location continuously, animating the playback until the end of the replay data.
   The **Stop** button (replaces the **Play** button while playing) can be used to pause the replay at a particular location.
   The **Replay** slider (or **Replay Position** box) can be used to go to a specific replay frame.

To simulate measurements on replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue and a Replay Mode Enabled message is displayed.
   To change the mode, **Replay Protection** must be unchecked.

2. Go to the **Measure** page.
   Modify settings for existing measurements, add new measurement tools, or delete measurement tools as desired. For information on adding and configuring measurements, see *Measurement and Processing* on page 146.

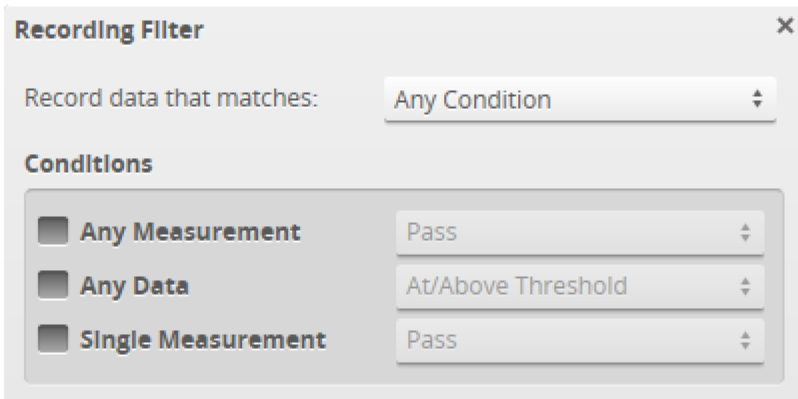
3. Use the **Replay Slider**, **Step Forward**, **Step Back**, or **Play** button to simulate measurements.
   Step or play through recorded data to execute the measurement tools on the recording.
   Individual measurement values can be viewed directly in the data viewer. Statistics on the measurements that have been simulated can be viewed in the **Dashboard** page; for more information on the dashboard, see *Dashboard* on page 524.

To clear replay data:

1. Stop the sensor if it is running by clicking the **Stop** button.
2. Click the **Clear Replay Data** button .

Recording Filtering

Replay data is often used for troubleshooting. But replay data can contain thousands of frames, which makes finding a specific frame to troubleshoot difficult. Recording filtering lets you choose which frames the sensor records, based on one or more conditions, which makes it easier to find problems.



How a sensor treats conditions

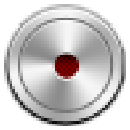
Setting	Description
Any Condition	The sensor records a frame when any condition is true.
All Conditions	The sensor only records a frame if all conditions are true.

Conditions

Setting	Description
Any Measurement	The sensor records a frame when *any* measurement is in the state you select. The following states are supported: • pass • fail or invalid • fail and valid • valid • invalid
Single Measurement	The sensor records a frame if the measurement with the ID you specify in **ID** is in the state you select. This setting supports the same states as the **Any Measurement** setting (see above).
Any Data	**At/Above Threshold**: The sensor records a frame if the number of valid points in the frame is above the value you specify in **Range Count Threshold**. **Below Threshold**: The sensor records a frame if the number of valid points is below the threshold you specify. In Surface mode, the number of valid points in the surface is compared to the threshold, not any sections that may be defined.

To set recording filtering:

1. Make sure recording is enabled by clicking the Record button.



2. Click the Recording Filtering button .
3. In the Recording Filtering dialog, choose how the sensor treats conditions:

   For information on the available settings, see *How a sensor treats conditions* on the previous page.
4. Configure the conditions that will cause the sensor to record a frame:

   For information on the available settings, see *Conditions* on the previous page.
5. Click the "x" button or outside of the Recording Filtering dialog to close the dialog.

   The recording filter icon turns green to show that recording filters have been set.

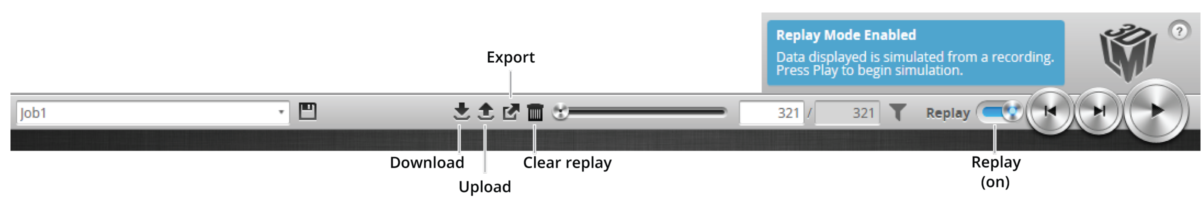
   When you run the sensor, it only records the frames that satisfy the conditions you have set.

Downloading, Uploading, and Exporting Replay Data

Replay data (recorded scan data) can be downloaded from a sensor to a client computer, or uploaded from a client computer to a sensor.

Data can also be exported from a sensor to a client computer in order to process the data using third-party tools.

	You can only upload replay data to the same sensor model that was used to create the data.



	Replay data is not loaded or saved when you load or save jobs.

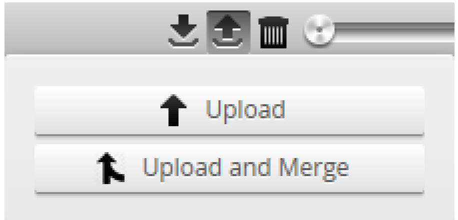
To download replay data:

1. Click the Download button .
2. In the **File Download** dialog, click **Save**.
3. In the **Save As...** dialog, choose a location, optionally change the name, and click **Save**.

To upload replay data:

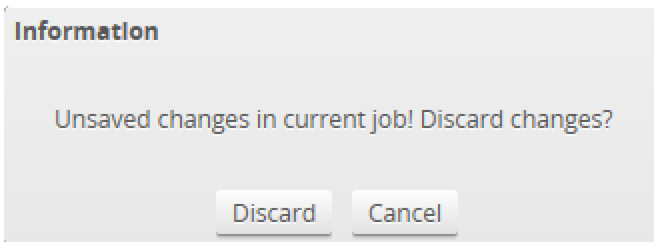
1. Click the Upload button .

   The Upload menu appears.



2. In the Upload menu, choose one of the following:
   - **Upload**: Unloads the current job and creates a new unsaved and untitled job from the content of the replay data file.
   - **Upload and merge**: Uploads the replay data and merges the data's associated job with the current job. Specifically, the settings on the **Scan** page are overwritten, but all other settings of the current job are preserved, including any measurements or models.

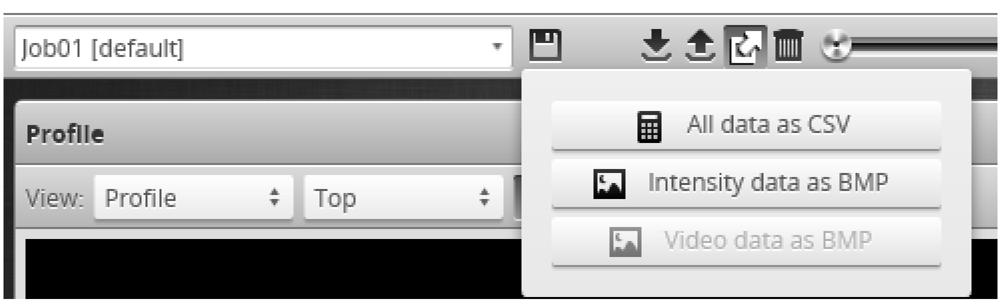
   If you have unsaved changes in the current job, the firmware asks whether you want to discard the changes.



3. Do one of the following:
   - Click **Discard** to discard any unsaved changes.
   - Click **Cancel** to return to the main window to save your changes.
4. If you clicked **Discard**, navigate to the replay data to upload from the client computer and click **OK**.
   The replay data is loaded, and a new unsaved, untitled job is created.

Replay data can be exported using the CSV format.

	Surface intensity data cannot be exported to the CSV format. It can only be exported separately as a bitmap.



To export replay data in the CSV format:

1. In the **Scan Mode** panel, switch to Profile or Surface.
2. Switch to Replay mode.
3. Click the Export button and select **All Data as CSV**.

   Only data at the current replay location is exported.
   Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Recording, Playback, and Measurement Simulation* on page 66.
4. (Optional) Convert exported data to another format using the CSV Converter Tool. For information on this tool, see *CSV Converter Tool* on page 862.

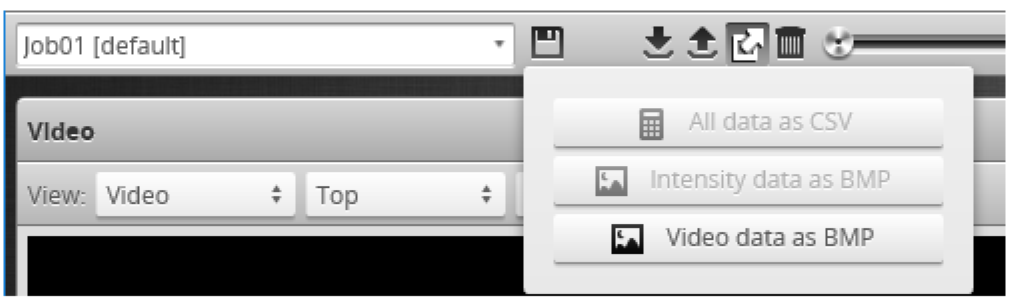
> The decision values in the exported data depend on the *current* state of the job, not the state during recording. For example, if you record data when a measurement returns a *pass* decision, change the measurement's settings so that a *fail* decision is returned, and then export to CSV, you will see a *fail* decision in the exported data.

Recorded intensity data can be exported to a bitmap (.BMP format). **Acquire Intensity** must be checked in the **Scan Mode** panel while data was being recorded in order to export intensity data.

To export recorded intensity data to the BMP format:

- Switch to Replay mode and click the **Export** button and select **Intensity data as BMP**.

  Only the intensity data in the current replay location is exported.
  Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Recording, Playback, and Measurement Simulation* on page 66.



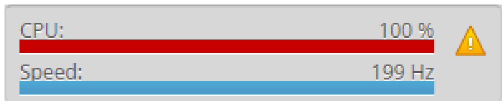
To export video data to a BMP file:

1. In the **Scan Mode** panel, switch to Video mode.
   Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Recording, Playback, and Measurement Simulation* on page 66.
2. Switch to Replay mode.
3. Click the Export button and select **Video data as BMP**.

Metrics Area

The **Metrics** area displays two important sensor performance metrics: CPU load and speed (current frame rate).

The **CPU** bar in the **Metrics** panel (at the top of the interface) displays how much of the CPU is being utilized.

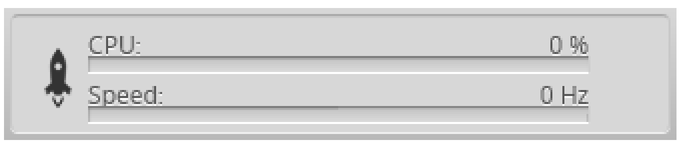


CPU at 100%

The **Speed** bar displays the frame rate of the sensor. A warning symbol (⚠) will appear next to it if triggers (external input or encoder) are dropped because the external rate exceeds the maximum frame rate.

Open the log for details on the warning. For more information on logs, see *Log* on the next page.

When a sensor is accelerated a "rocket" icon appears in the metrics area.



Data Viewer

The data viewer is displayed in both the **Scan** and the **Measure** pages, but displays different information depending on which page is active.

When the **Scan** page is active, the data viewer displays sensor data and can be used to adjust the active area and other settings. Depending on the selected operation mode (page 88), the data viewer can display video images, sections, or surfaces. For details, see *Data Viewer* on page 115.

When the **Measure** page is active, the data viewer displays sensor data onto which representations of measurement tools and their measurements are superimposed. For details, see *Data Viewer* on page 147.

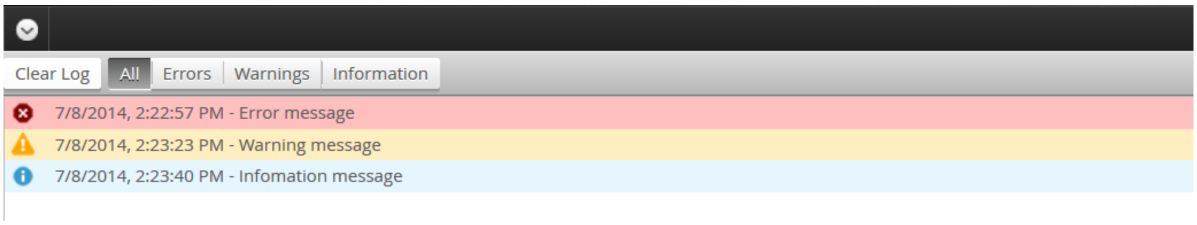
Status Bar

The status bar lets you do the following:

- See sensor messages in the log.
- See frame information.
- Change the interface language.
- Switch to Quick Edit mode.

Log

The log, located at the bottom of the web interface, is a centralized location for all messages that the sensor displays, including warnings and errors.



A number indicates the number of unread messages:

To use the log:

1. Click on the Log open button at the bottom of the web interface.
2. Click on the appropriate tab for the information you need.

Frame Information

The area to the right of the status bar displays useful frame information, both when the sensor is running and when viewing recorded data.

This information is especially useful when you have enabled recording filtering. If you look at a recording playback, when you have enabled recording filtering, some frames can be excluded, resulting in variable "gaps" in the data.

The following information is available:

Frame Index: Displays the index in the data buffer of the current frame. The value resets to 0 when the sensor is restarted or when recording is enabled.

Master Time: Displays the recording time of the current frame, with respect to when the sensor was started.

Encoder Index: Displays the encoder value at the time of the last encoder Z index pulse. Note this is not the same as the encoder value at the time the frame was captured.

Timestamp: Displays the timestamp the current frame, in microseconds from when the sensor was started.

To switch between types of frame information:

- Click the frame information area to switch to the next available type of information.

Quick Edit Mode

When working with a very large number of measurement tools (for example, a few dozen) or a very complex user-created GDK tool, you can switch to a "Quick Edit" mode to make configuration faster.

When this mode is enabled, the data viewer and measurement results are not refreshed after each setting change. Also, when Quick Edit is enabled, in Replay mode, stepping through frames or playing back scan data does not change the displayed frame.

When a sensor is running, Quick Edit mode is ignored: all changes to settings are reflected immediately in the data viewer.

Interface Language

The language button on the right side of the status bar at the bottom of the interface lets you change the language of the interface.

To change the language:

1. Click the language button at the bottom of the web interface.

2. Choose a language from the list.

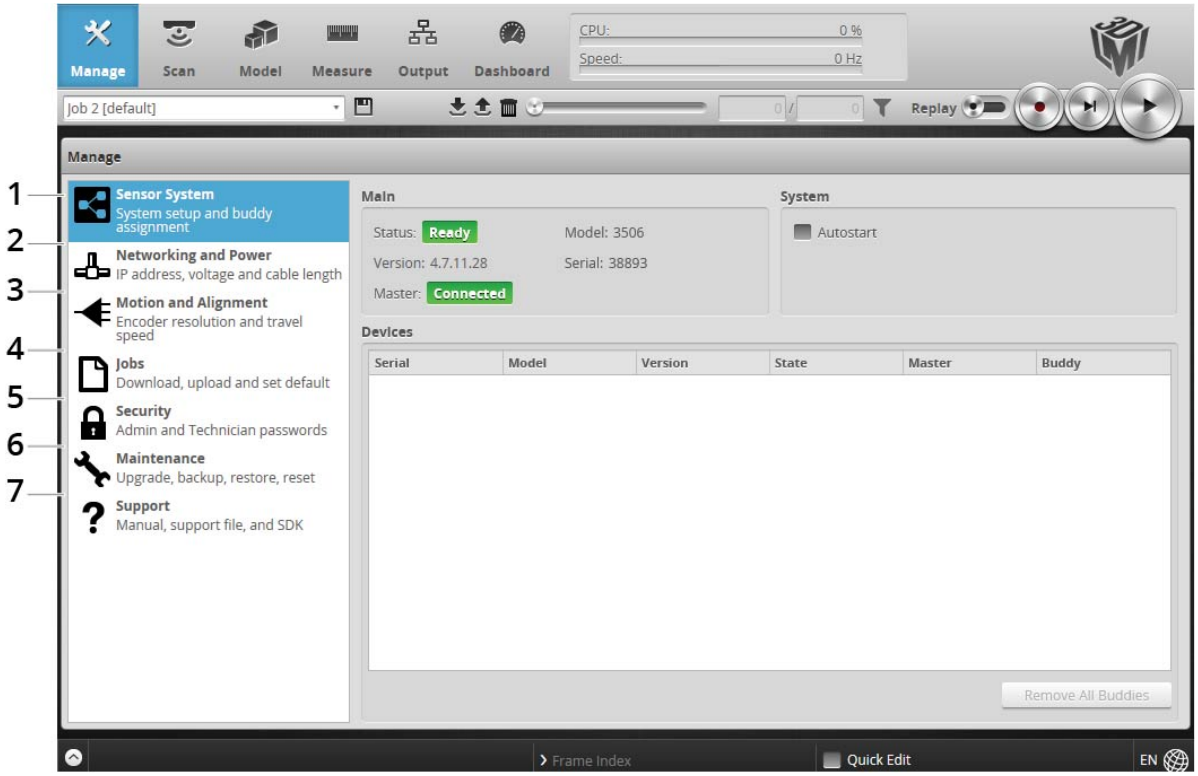
   The interface reloads on the page you were working in, displaying the page using the language you chose. The sensor state is preserved.

Management and Maintenance

The following sections describe how to set up the sensor connections and networking, how to calibrate encoders and choose the alignment reference, and how to perform maintenance tasks.

Manage Page Overview

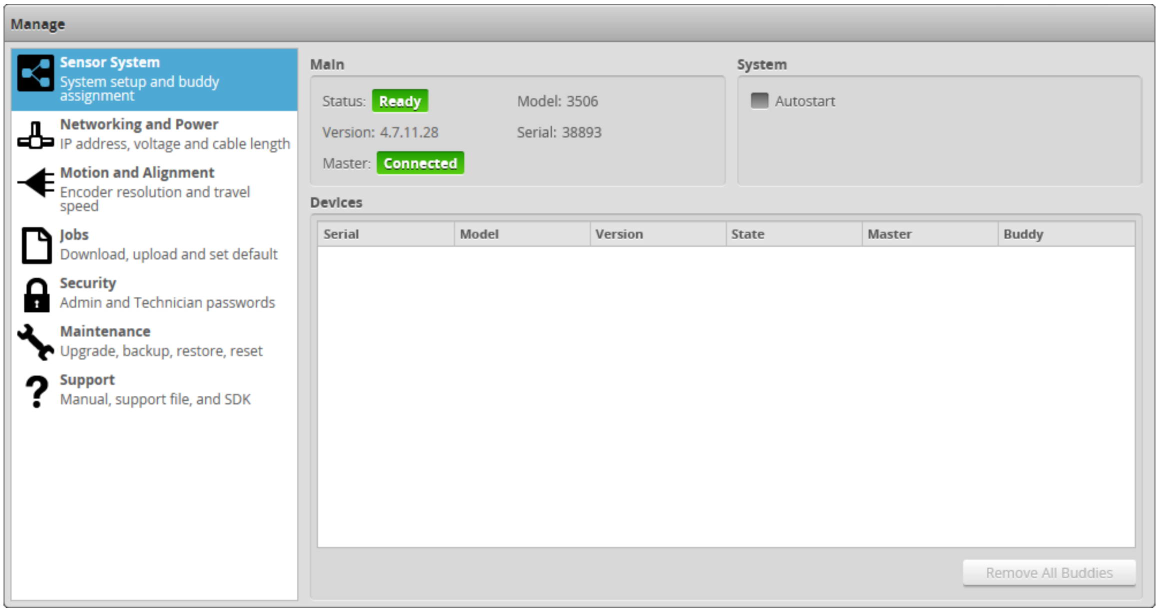
The sensor's system and maintenance tasks are performed on the **Manage** page.



	Element	Description
1	Sensor System	Contains sensor information and the autostart setting. See *Sensor System* on the next page.
2	Networking and Power	Contains settings for configuring the network, as well as power and cordset length. See *Networking and Power* on the next page.
3	Motion and Alignment	Contains settings to configure the encoder. See *Motion and Alignment* on page 77.
4	Jobs	Lets you manage jobs stored on the sensor. See *Jobs* on page 78.
5	Security	Lets you change passwords. See *Security* on page 80.
6	Maintenance	Lets you upgrade firmware, create/restore backups, and reset sensors. See *Maintenance* on page 81.
7	Support	Lets you open an HTML version or download a PDF version of the manual, download the SDK, or save a support file. Also provides device information. See *Support* on page 84

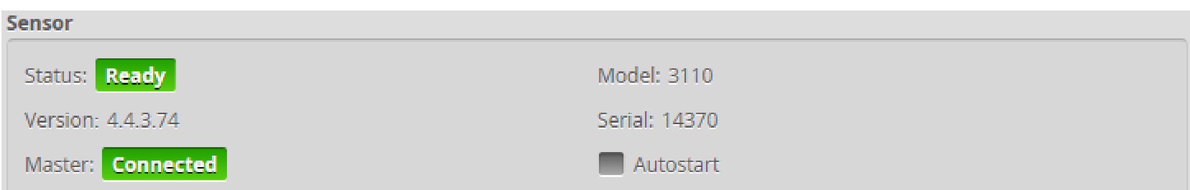
Sensor System

The following sections describe the **Sensor System** category on the **Manage** page. This category provides sensor information and the autostart setting.



Sensor Autostart

With the **Autostart** setting enabled, scanning and measurements begin automatically when the sensor is powered on. Autostart must be enabled if the sensor will be used without being connected to a computer.



To enable/disable Autostart:

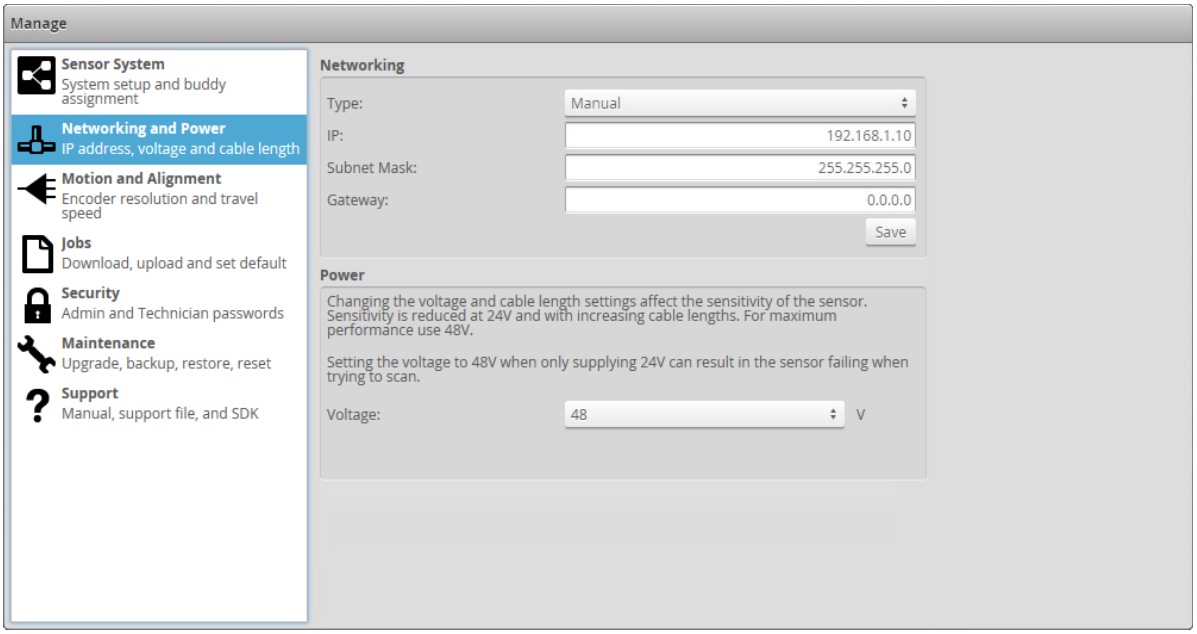
1. Go to the **Manage** page and click on the **Sensor System** category.
2. Check/uncheck the **Autostart** option in the **Main** section.

Networking and Power

The **Networking** category on the **Manage** page provides network settings. Settings must be configured to match the network to which the sensors are connected.

The category also provides power and cordset length settings.

If you run a Gocator 3210 sensor at 24 volts, you must properly configure the power and cordset length (see below); otherwise, the sensor will fail to complete scans.

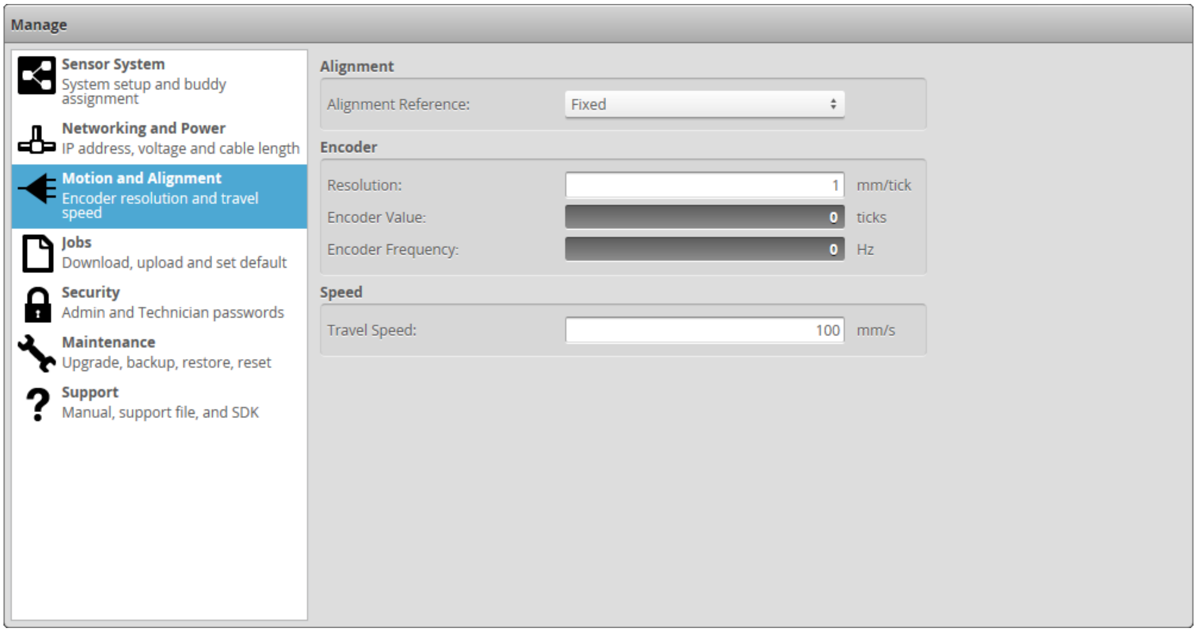


To configure the network settings:

1. Go to the **Manage** page.
2. In the **Networking** category, specify the Type, IP, Subnet Mask, and Gateway settings.
   The sensor can be configured to use DHCP or assigned a static IP address by selecting the appropriate option in the **Type** drop-down.
3. For 3210 sensors, configure **Voltage** and **Cordset Length**.
   When a 3210 sensor is run at 24 volts and with a longer cordset, the sensor must lower the projector's intensity to limit the current going to the sensor. As a result, you may need to increase the sensor's exposure to compensate for the lower projector intensity compared to running at 48 volts; for more information, see *Running Gocator 3210 on 24 VDC* on page 100.
4. Click on the **Save** button.
   You will be prompted to confirm your selection.

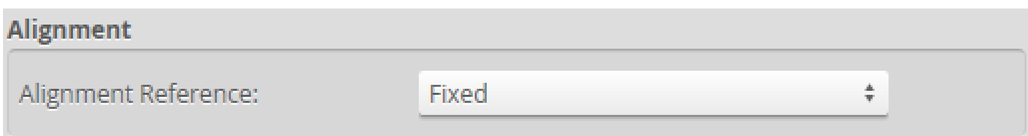
Motion and Alignment

The **Motion and Alignment** category on the **Manage** page lets you configure alignment reference, encoder resolution, and travel speed, and confirm that encoder signals are being received by the sensor.



Alignment Reference

The **Alignment Reference** setting can have one of two values: **Fixed** or **Dynamic**.



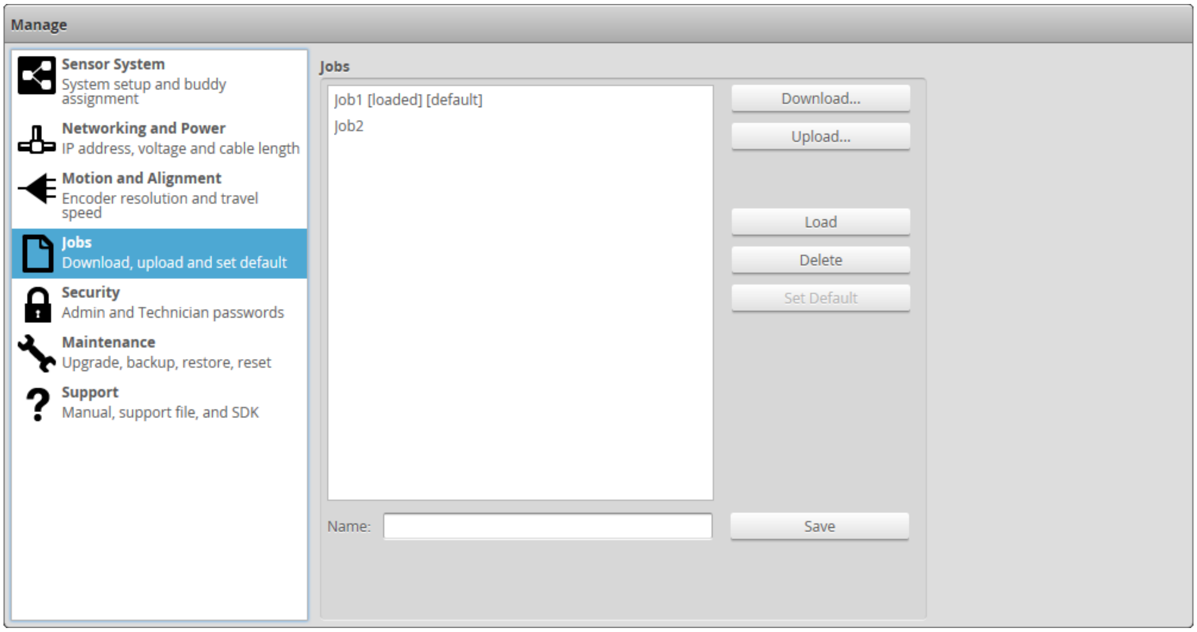
Setting	Description
Fixed	A single, global alignment is used for all jobs. This is typically used when the sensor mounting is constant over time and between scans, for example, when the sensor is mounted in a permanent position over a conveyor belt.
Dynamic	A separate alignment is used for each job. This is typically used when the sensor's position relative to the object scanned is always changing, for example, when the sensor is mounted on a robot arm moving to different scanning locations.

To configure alignment reference:

1. Go to the **Manage** page and click on the **Motion and Alignment** category.
2. In the Alignment section, choose **Fixed** or **Dynamic** in the **Alignment Reference** drop-down.

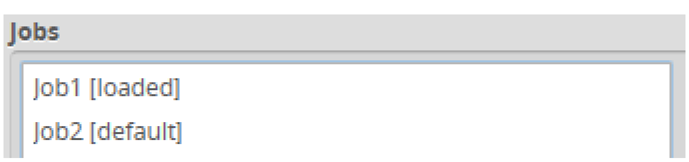
Jobs

The **Jobs** category on the **Manage** page lets you manage the jobs stored on a sensor.

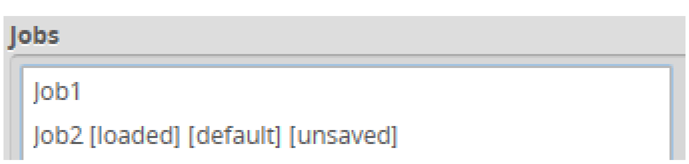


Element	Description
Name field	Used to provide a job name when saving files.
Jobs list	Displays the jobs that are currently saved in the sensor's flash storage.
Save button	Saves current settings to the job using the name in the **Name** field.
Load button	Loads the job that is selected in the job list. Reloading the current job discards any unsaved changes.
Delete button	Deletes the job that is selected in the job list.
Set as Default button	Sets the selected job as the default to be loaded when the sensor starts. When the default job is selected, this button is used to clear the default.
Download... button	Downloads the selected job to the client computer.
Upload... button	Uploads a job from the client computer.

Jobs can be loaded (currently activated in sensor memory) and set as default independently. For example, Job1 could be loaded, while Job2 is set as the default. Default jobs load automatically when a sensor is power cycled or reset.



Unsaved jobs are indicated by "[unsaved]".



To save a job:

1. Go to the **Manage** page and click on the **Jobs** category.
2. Provide a name in the **Name** field.
   To save an existing job under a different name, click on it in the **Jobs** list and then modify it in the **Name** field.
3. Click on the **Save** button or press **Enter**.
   Saving a job automatically sets it as the default, that is, the job loaded when then sensor is restarted.

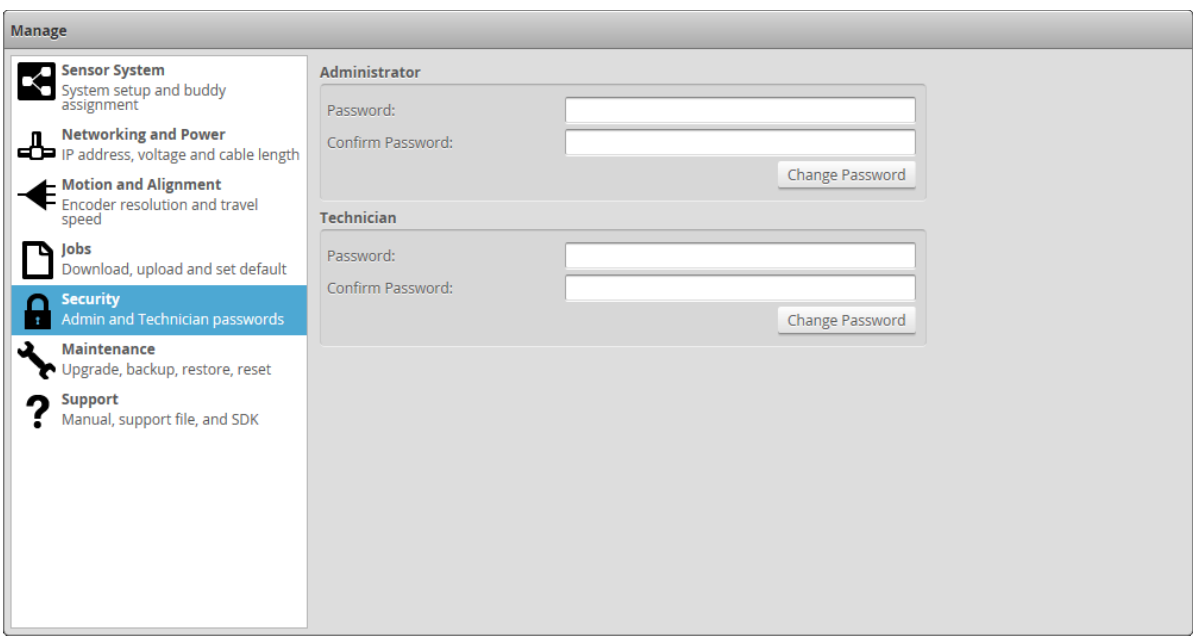
To download, load, or delete a job, or to set one as a default, or clear a default:

1. Go to the **Manage** page and click on the **Jobs** category.
2. Select a job in the **Jobs** list.
3. Click on the appropriate button for the operation.

Security

You can prevent unauthorized access to a sensor by setting passwords. Each sensor has two accounts: Administrator and Technician.

By default, no passwords are set. When you start a sensor, you are prompted for a password only if a password has been set.



Account Types

Account	Description
Administrator	The Administrator account has privileges to use the toolbar (loading and saving jobs, recording and viewing replay data), to view all pages and edit all settings, and to perform setup procedures such as

Account	Description
	sensor alignment.
Technician	The Technician account has privileges to use the toolbar (loading and saving jobs, recording and viewing replay data), to view the **Dashboard** page, and to start or stop the sensor.

The Administrator and Technician accounts can be assigned unique passwords.

To set or change the password for the Administrator account:

1. Go to the **Manage** page and click on the **Security** category.
2. In the **Administrator** section, enter the Administrator account password and password confirmation.
3. Click **Change Password**.
   The new password will be required the next time that an administrator logs in to the sensor.

To set or change the password for the Technician account:

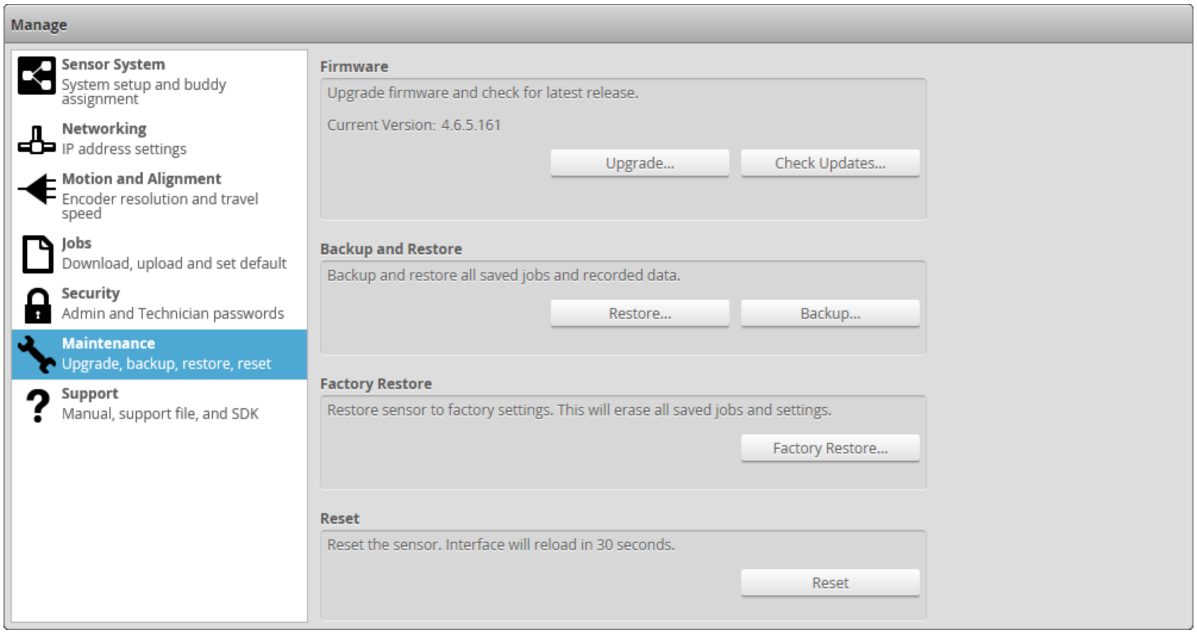
1. Go to the **Manage** page and click on the **Security** category.
2. In the **Technician** section, enter the Technician account password and password confirmation.
3. Click **Change Password**.
   The new password will be required the next time that a technician logs in to the sensor.

If the administrator or technician password is lost, the sensor can be recovered using a special software tool. See *Sensor Discovery Tool* on page 861 for more information.

Maintenance

The **Maintenance** category in the **Manage** page is used to do the following:

- upgrade the firmware and check for firmware updates;
- back up and restore all saved jobs and recorded data;
- restore the sensor to factory defaults;
- reset the sensor.

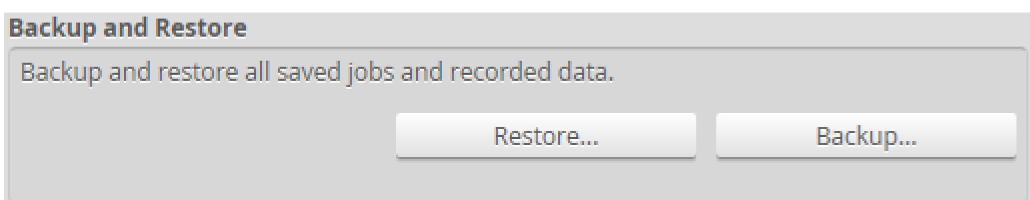


Sensor Backups and Factory Reset

You can create sensor backups, restore from a backup, and restore to factory defaults in the **Maintenance** category.

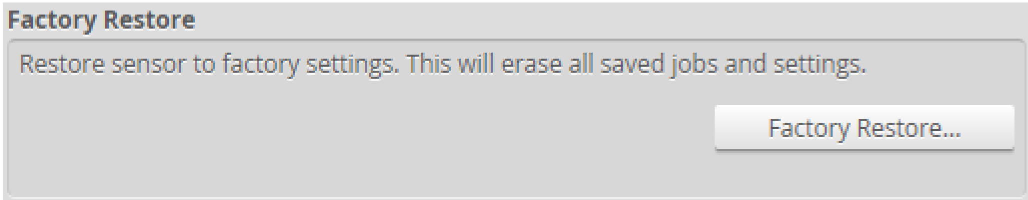
Backup files contain all of the information stored on a sensor, including jobs and alignment.

> An Administrator should create a backup file in the unlikely event that a sensor fails and a replacement sensor is needed. If this happens, the new sensor can be restored with the backup file.



To create a backup:

1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Backup...** button under **Backup and Restore**.
3. When you are prompted, save the backup.
   Backups are saved as a single archive that contains all of the files from the sensor.



To restore from a backup:

1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Restore...** button under **Backup and Restore**.
3. When you are prompted, select a backup file to restore.
   The backup file is uploaded and then used to restore the sensor. Any files that were on the sensor before the restore operation will be lost.

To restore a sensor to its factory default settings:

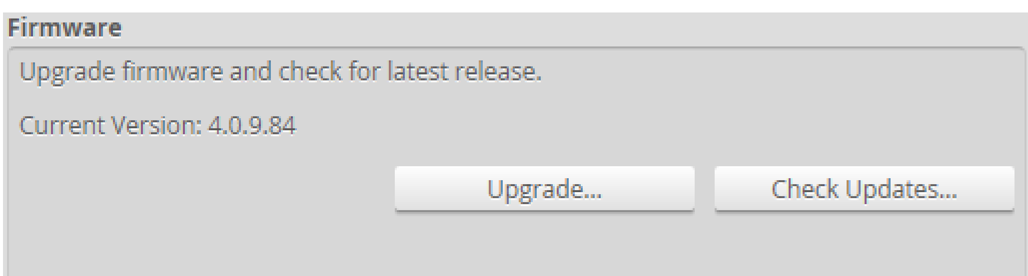
1. Go to the **Manage** page and click on **Maintenance**.
2. Consider making a backup.
   Before proceeding, you should perform a backup. Restoring to factory defaults cannot be undone.
3. Click the **Factory Restore...** button under **Factory Restore**.
   You will be prompted whether you want to proceed.

Firmware Upgrade

LMI recommends routinely updating firmware to ensure that sensors always have the latest features and fixes.

If you are upgrading from a 3.6 or 4.x firmware indicated in the upgrade path below, make sure to follow the sequence of firmware upgrades.

3.6 → 3.6 SR5 → 4.4 → 4.6 SR2 → 5.x/6.x



To download the latest firmware:

1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Check Updates...** button in the **Firmware** section.

3. Download the latest firmware.
   If a new version of the firmware is available, follow the instructions to download it to the client computer.

If the client computer is not connected to the Internet, firmware can be downloaded and transferred to the client computer by using another computer to download the firmware from LMI's website: *http://www.lmi3D.com/support/downloads*.

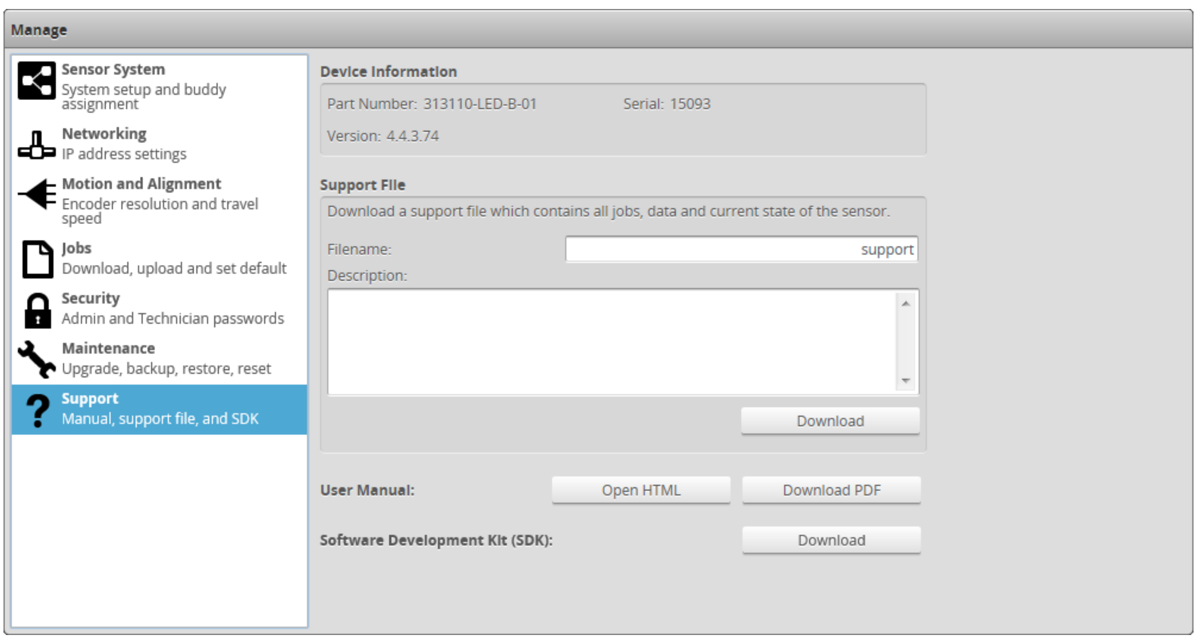
To upgrade the firmware:

1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Upgrade...** button in the **Firmware** section.
3. Locate the firmware file in the **File** dialog and then click open.
4. Wait for the upgrade to complete.
   After the firmware upgrade is complete, the sensor will self-reset. If a buddy has been assigned, it will be upgraded and reset automatically.

Support

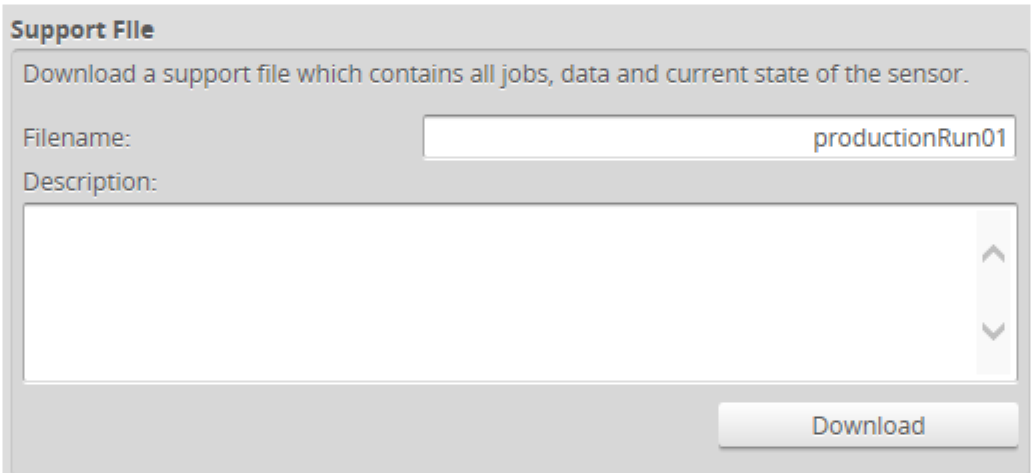
The **Support** category in the **Manage** page is used to do the following:

- Open an HTML version or download a PDF version of the manual.
- Download the SDK.
- Save a support file.
- Get device information.



Support Files

You can download a support file from a sensor and save it on your computer. You can then use the support file to create a scenario in the emulator (for more information on the emulator, see *Gocator Emulator* on page 537). LMI's support staff may also request a support file to help in troubleshooting.



To download a support file:

1. Go to the **Manage** page and click on the **Support** category.
2. In **Filename**, type the name you want to use for the support file.

   When you create a scenario from a support file in the emulator, the filename you provide here is displayed in the emulator's scenario list.

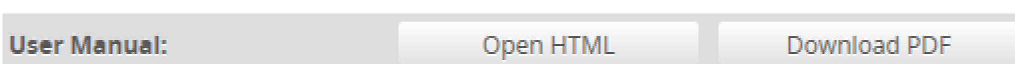
   Support files end with the .gs extension, but you do not need to type the extension in **Filename**.
3. (Optional) In **Description**, type a description of the support file.

   When you create a scenario from a support file in the emulator, the description is displayed below the emulator's scenario list.
4. Click **Download**, and then when prompted, click **Save**.

   ⚠ Downloading a support file stops the sensor.

Manual Access

You can access the Gocator manuals from within the Web interface.



You may need to configure your browser to allow pop-ups to open or download the manual.

To access the manuals:

1. Go to the **Manage** page and click on the **Support** category
2. Next to **User Manual**, click one of the following:

- **Open HTML**: Opens the HTML version of the manual in your default browser.
- **Download PDF**: Downloads the PDF version of the manual to the client computer.

Software Development Kit

You can download the Gocator SDK from within the Web interface.

To download the SDK:

1. Go to the **Manage** page and click on the **Support** category
2. Next to **Software Development Kit (SDK)**, click **Download**
3. Choose the location for the SDK on the client computer.

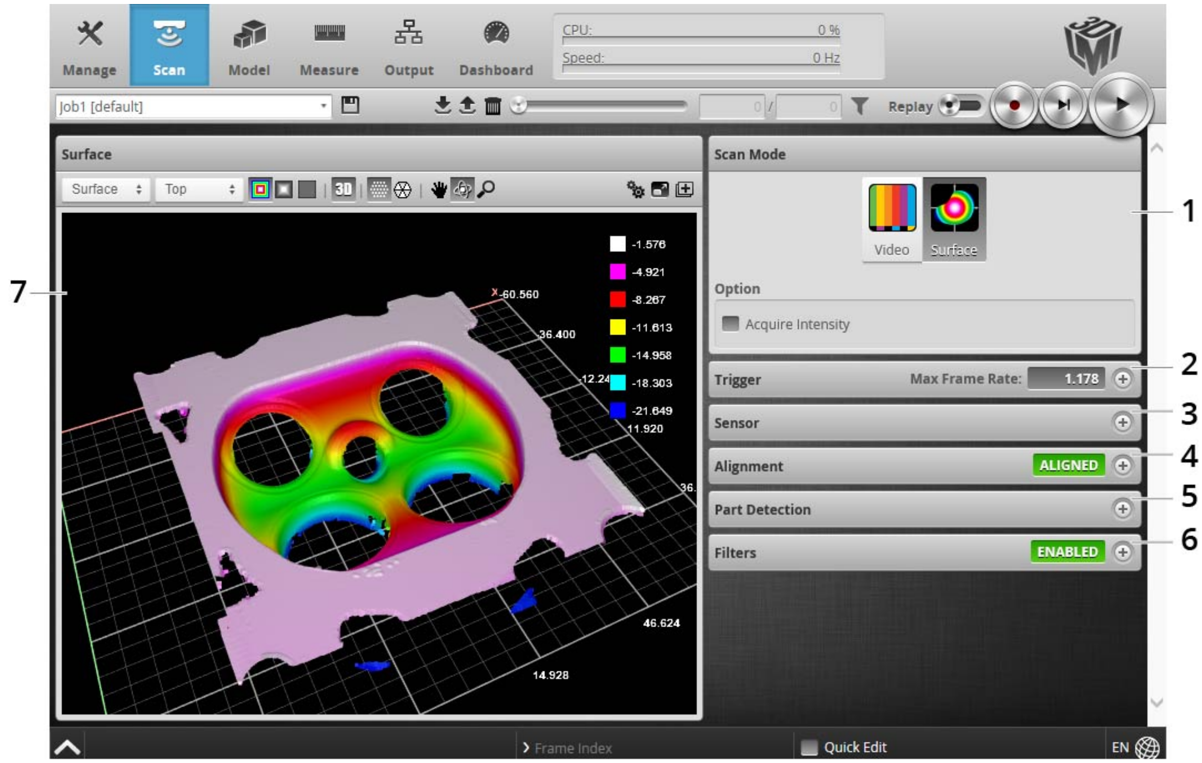
For more information on the SDK, see *Development Kits* on page 836.

Scan Setup and Alignment

The following sections describe the steps to configure sensors for data acquisition using the **Scan** page. Setup and alignment should be performed before adding and configuring measurements or outputs.

Scan Page Overview

The **Scan** page lets you configure sensors and perform alignment.



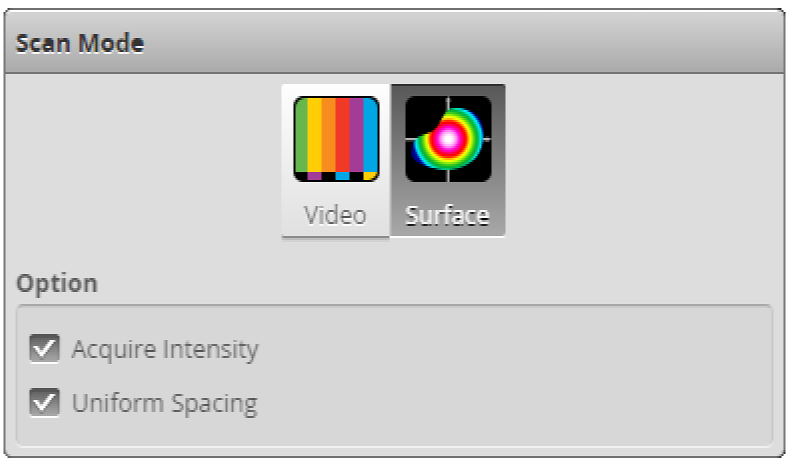
	Element	Description
1	Scan Mode panel	Contains settings for the current scan mode and other options. See *Scan Modes* on the next page.
2	Trigger panel	Contains trigger source and trigger-related settings. See *Triggers* on page 89.
3	Sensor panel	Contains settings for an individual sensor, such as active area or exposure. See *Sensor* on page 92.
4	Alignment panel	Used to perform alignment. See *Alignment* on page 106.
5	Part Detection panel	Used to set the part detection logic for sorting data into discrete objects. See *Part Detection* on page 111.
6	Filters panel	Contains settings for post-processing of the profiles. See *Filters* on page 108.
7	Data Viewer	Displays sensor data and adjusts regions of interest. Depending on the current operation mode, the data viewer can display video images or scan data. See *Data Viewer* on page 115.

The following table provides quick references for specific goals that you can achieve from the panels in the **Scan** page.

Goal	Reference
Select a trigger source that is appropriate for the application.	Triggers (page 89)
Ensure that camera exposure is appropriate for scan data acquisition.	Exposure (page 96)
Find the right balance between data quality, speed, and CPU utilization.	Active Area (page 94) Exposure (page 96) Job File Structure (page 552)
Calibrate the system so that 3D data can be aligned to a reference plane.	
Set up the part detection logic to create discrete objects from scan data.	Part Detection (page 111)

Scan Modes

The sensor web interface supports a video mode and one or more data acquisition modes. The scan mode can be selected in the **Scan Mode** panel.



Mode and Option	Description
Video	Outputs video images from the sensor. This mode is useful for configuring exposure time and troubleshooting stray light or ambient light problems.
Surface	Outputs 3D point clouds and performs surface measurements.Part detection can be enabled on a surface to identify discrete parts (*Part Detection* on page 111).
Uniform Spacing	When this option is enabled, data points are resampled to a uniform spacing . Set the size of the spacing in the **Spacing** tab (see *Spacing Interval* on page 103). When the option is disabled, the sensor outputs unprocessed range data. The sensor reports data points in (x, y, z) coordinate tuples. No measurement tools are available when the option is disabled. Disable this option to extract ranges from the sensor at the highest possible rate. The Y offset, X angle, and Z angle transformations cannot be non-

Mode and Option	Description
	zero when **Uniform Spacing** is unchecked. Therefore, when aligning a sensor using a bar alignment target with **Uniform Spacing** unchecked, set the **Degrees of Freedom** setting to **X, Z, Y Angle**, which prevents these transformations from being non-zero. If you are using a layout in which sensors are angled around the Y axis in order to capture "side" data, you must uncheck **Uniform Spacing**. However, currently, only a limited set of built-in measurement tools are able to perform measurements on the resulting data. If more complex measurements are required, data can be processed using an SDK-based application instead.
Acquire Intensity	When this option is enabled, an intensity value will be produced for each data point. For more information on intensity, see *Intensity Output* on page 125.

Triggers

A trigger is an event that causes a sensor to take a single 3D snapshot. Triggers are configured in the **Trigger** panel.

When a trigger is processed, the LED light pattern is strobed and the cameras expose to produce images. The resulting images are processed inside the sensor to yield a 3D point cloud. The data can then be used for measurement.

The top-right of the **Trigger** panel displays the maximum speed at which an object could be captured at, calculated based on the exposure values, active area and the number of projection patterns required.

The sensor can be triggered by one of the sources described in the table below.

If the sensor is connected to a Master 400 or higher, encoder and digital (external) input signals over the IO cordset are *ignored*. The sensor instead receives these signals from the Master; for encoder and digital input pinouts on Masters, see the section corresponding to your Master in *Master Network Controllers* on page 900.

If the sensor is connected to a Master 100 (or no Master is used), the sensor receives signals over the IO cordset. For information on connecting encoder and digital input signals to a sensor in these cases, see *Encoder Input* on page 897 and *Digital Input* on page 896, respectively.

Trigger Source	Description
Time	Sensors have an internal clock that can be used to generate fixed-frequency triggers. The external input can be used to enable or disable the time triggers.
External Input	A digital input can provide triggers in response to external events (e.g., photocell). The external input triggers on the rising edge of the signal. When triggers are received at a frequency higher than the maximum frame rate, some triggers may not be accepted. The **Trigger Drops Indicator** in the **Dashboard** page can be used to check for this condition. For information on the maximum input trigger rate, see *Maximum Input Trigger Rate* on page 92.

Trigger Source	Description
Software	A network command can be used to send a software trigger. See *Protocols* on page 650 for more information.

Depending on the setup and measurement tools used, the CPU utilization may exceed 100%, which reduces the overall acquisition speed. For the estimated acquisition speeds under different settings, see *Estimated Performance and Scan Rates* on page 535.

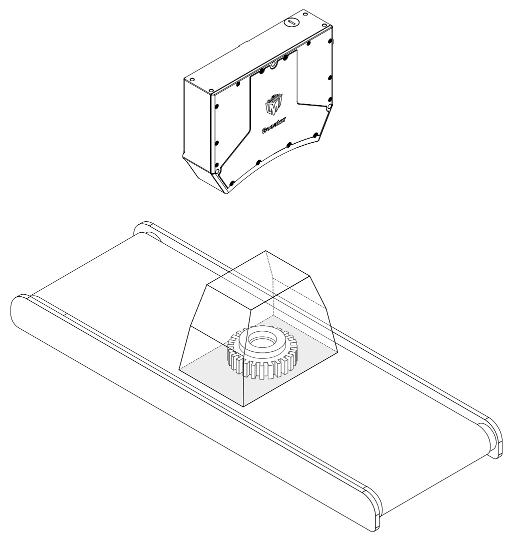
For examples of typical real-world scenarios, see *Trigger Examples* below. For information on the settings used with each trigger source, see *Trigger Settings* below.

Trigger Examples

Example: External Input + Conveyor
External input triggering can be used to produce a snapshot for 3D measurement.

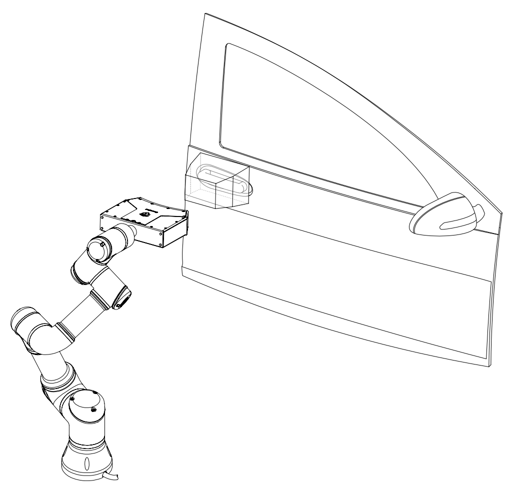
For example, a photocell can be connected as an external input to generate a trigger pulse when a target object has moved into position.

An external input can also be used to gate the trigger signals when time or encoder triggering is used. For example, a photocell could generate a series of trigger pulses as long as there is a target in position.



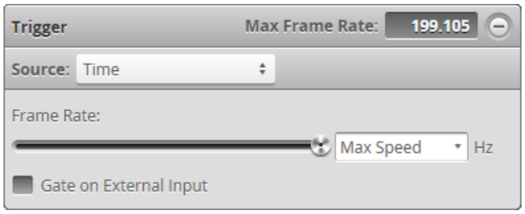
Example: Software Trigger + Robot Arm
Software triggering can be used to produce a snapshot for 3D measurement.

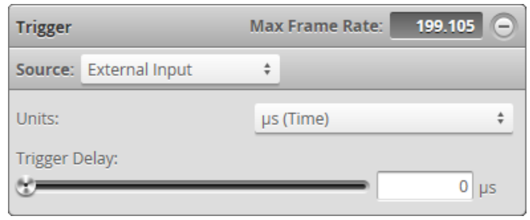
A software trigger can be used in systems that use external software to control the activities of system components.

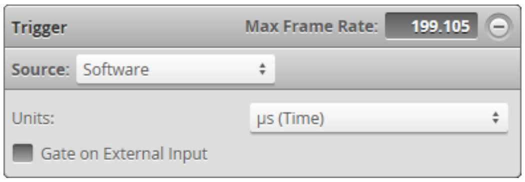


Trigger Settings

The trigger source is selected using the **Trigger** panel in the **Scan** page.







After specifying a trigger source, the **Trigger** panel shows the parameters that can be configured.

Parameter	Trigger Source	Description
Source	All	Selects the trigger source (**Time**, **Encoder**, **External Input**, or **Software**).
Frame Rate	Time	Controls the frame rate. Select **Max Speed** from the drop-down to lock to the maximum frame rate. Fractional values are supported. For example, 0.1 can be entered to run at 1 frame every 10 seconds.
Gate on External Input	Time, Encoder	External input can be used to enable or disable data acquisition in a sensor. When this option is enabled, the sensor will respond to time or encoder triggers only when the external input is asserted. See *Digital Input* on page 896 for more information on connecting external input to sensors.
Units	External Input, Software	Specifies whether the trigger delay, output delay, and output scheduled command operate in the time. The unit is implicitly set to microseconds with Time trigger source. .
Trigger Delay	External Input	Controls the amount of time or the distance the sensor waits before producing a frame after the external input is activated. This is used to compensate for the positional difference between the source of the external input trigger (e.g., photocells) and the sensor. Trigger delay is only supported in single exposure mode; for details, see *Exposure* on page 96.

To configure the trigger source:

1. Go to the **Scan** page.
2. Expand the **Trigger** panel by clicking on the panel header.

3. Select the trigger source from the drop-down.
4. Configure the settings.
   See the trigger parameters above for more information.
5. Save the job in the **Toolbar** by clicking the **Save** button.

Maximum Input Trigger Rate

The maximum external input trigger rate in a system including Master 400 or higher is 20 kHz.

When using a standalone sensor or a sensor connected to a Master 100, the maximum trigger rate is 32 kHz. This rate is limited by the fall time of the signal, which depends on the Vin and duty cycles. To achieve the maximum trigger rate, the Vin and duty cycles must be adjusted as follows:

Maximum Speed	Vin	Maximum Duty Cycle
32 kHz	3.3 V	88%
32 kHz	5 V	56%
32 kHz	7 V	44%
32 kHz	10 V	34%

At 50% duty cycle, the maximum trigger rates are as follows:

Vin	Maximum Speed
3.3 V	34 kHz
5 V	34 kHz
10 V	22 kHz

Maximum Encoder Rate

On a standalone sensor, with the encoder directly wired into the I/O port or through a Master 100, the maximum encoder rate is about 1 MHz.

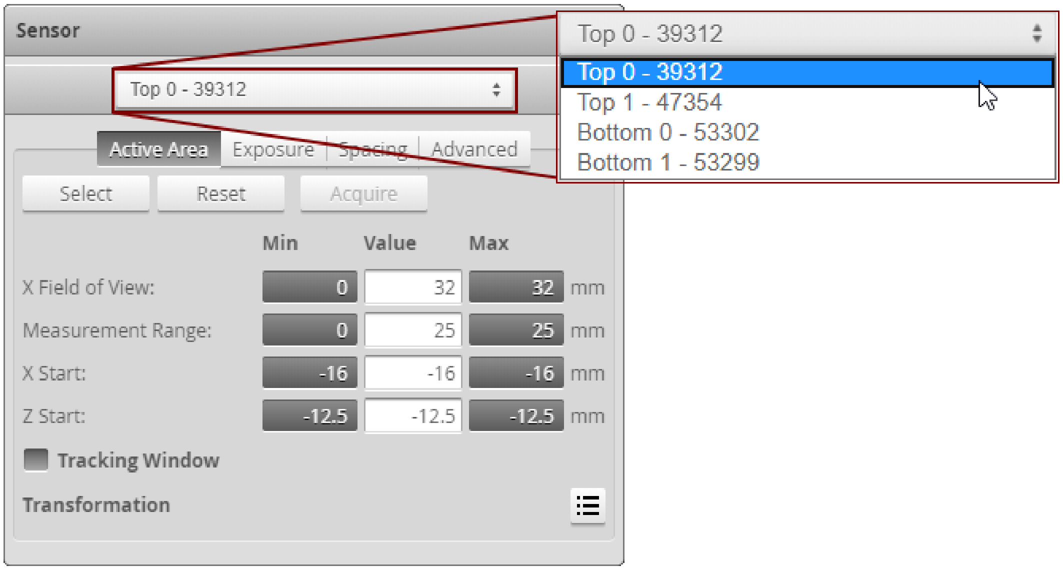
For sensors connected through a Master 400 or higher, with the encoder signal supplied to the Master, the maximum rate is about 300 kHz.

Sensor

The following sections describe the settings that are configured in the **Sensor** panel on the **Scan** page.

If you are using a mixed-model dual- or multi-sensor system, after adding Buddy sensors, you should check in the **Sensor** panel that the settings for each Buddy sensor has a valid and in-range value. Otherwise, the system may not start or be able to perform alignment. A Buddy sensor's settings may become invalid after being added to a system because Gocator automatically carries certain settings from the Main sensor to the Buddy sensors, which may be incompatible with a Buddy sensor. For example, if Main sensor were a wide FOV model and its active area is set to be greater than the maximum possible active area of a small FOV Buddy sensor, the Buddy sensor's active area settings would be invalid. You would need to modify the Buddy sensor's

To check these settings, use the drop-down at the top of the **Sensor** panel to select each sensor, and check that there are no errors indicated in the setting fields for each sensor. Check in all of the tabs in the panel, but especially the **Active Area** tab.



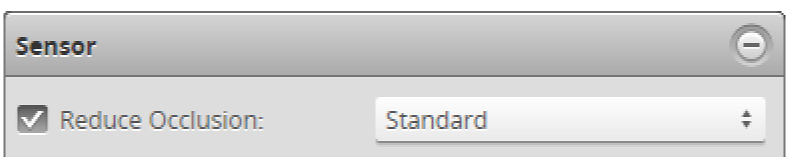
Sensor drop-down in a four-sensor system.

Reduce Occlusion

When the **Reduce Occlusion** option is disabled, 3D data is acquired using *only* stereo correlation between the two cameras, meaning that a point on the target must be visible in both cameras to produce a 3D data point.

When this option is enabled (default), in addition to stereo correlation, each camera independently triangulates off the LED light pattern, which may improve performance on targets with complicated shapes that can cause occlusions.

For more information, see *Stereo Correlation vs. Independent Triangulation* on page 44.



To enable or disable the Reduce Occlusion option:

1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, the **Sensor** panel will not be displayed.
3. Expand the **Sensor** panel by clicking on the panel header.
4. Check or uncheck the **Reduce Occlusion** checkbox.
5. In the dropdown next to **Reduce Occlusion**, choose one of the following:
   Standard: The standard algorithm for merging images from each camera is used.

High Quality: An enhanced algorithm is used to merge images from each camera, which reduces the seam where images merge. Processing time may be increased.

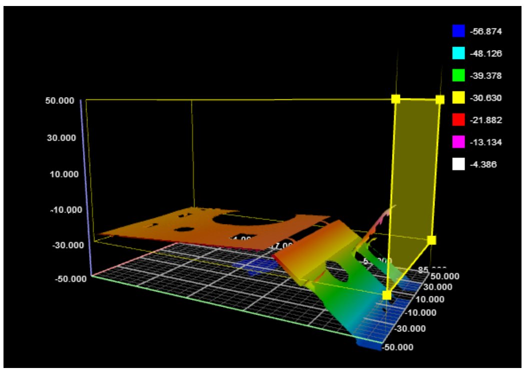
6. Save the job in the **Toolbar** by clicking the **Save** button.

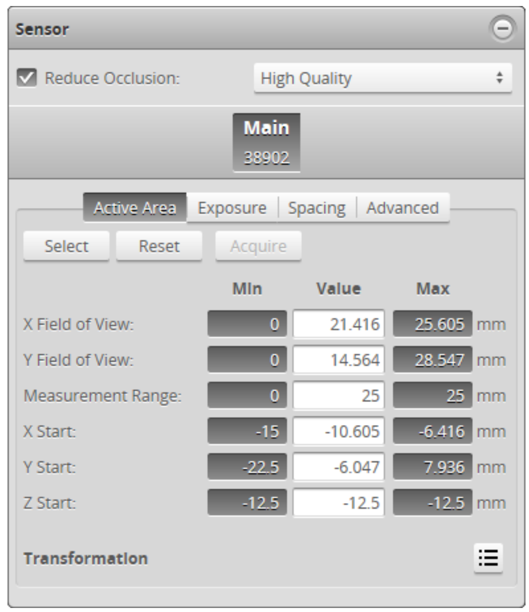
Active Area

Active area refers to the region within the sensor's maximum field of view that is used for data acquisition.

By default, the active area covers the sensor's entire field of view. By reducing the active area, the sensor can operate at higher speeds. You can also reduce the active area to exclude areas that are affected by ambient light.

Active area is set in the **Active Area** tab on the **Sensor** panel.





To set the active area:

1. Go to the **Scan** page.
2. Choose a mode other than Video mode.
3. Expand the **Sensor** panel by clicking on the panel header or the button.
4. Click on the **Active Area** tab.
5. Click **Select**.
6. Click **Acquire** to see a scan while setting the active area.
   Acquiring a scan while setting the active area can help you determine where to size and place the active area.
7. Set the active area.
   Adjust the active area graphically in the data viewer or enter the values manually in the fields.

The 2D view lets you adjust the size and position of the active area on the X and Z axis. The 3D view lets you adjust the size and position in the X, Y, and Z axis. For more information, see *Regions* on page 152.

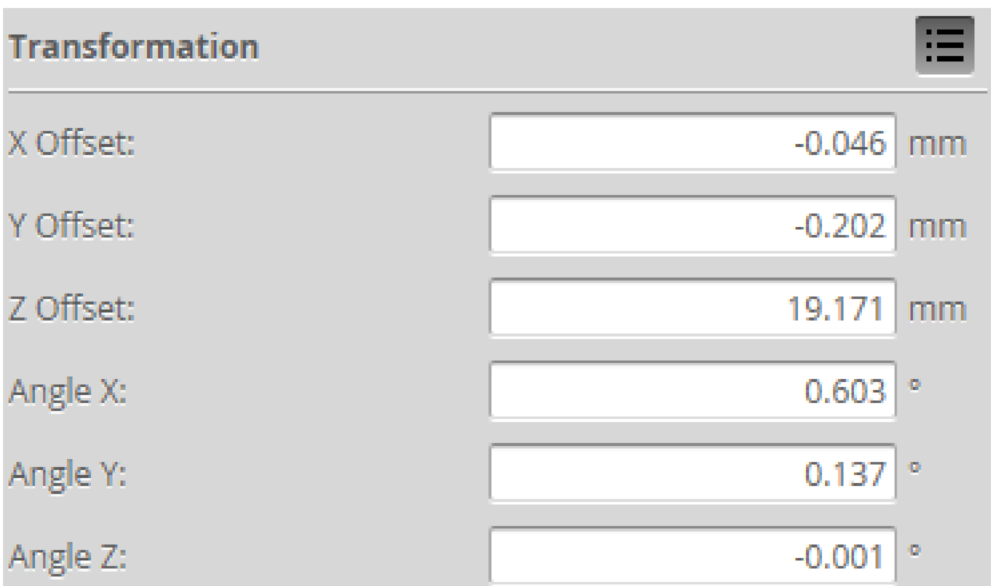
8. Click the **Save** button in the **Sensor** panel.
   Click the **Cancel** button to cancel setting the active area.

9. Save the job in the **Toolbar** by clicking the **Save** button.

> Scanning devices are usually more accurate at the near end of their measurement range. If your application requires a measurement range that is small compared to the maximum measurement range of the sensor, mount the sensor so that the active area can be defined at the near end of the measurement range.

Transformations

The transformation settings determine how data is converted from sensor coordinates to system coordinates (for an overview on coordinate systems, see *Coordinate Systems* on page 45). Typically, transformations are set when you align a sensor. However, you can also manually set values using the **Transformations** section of the **Active Area** tab on the Sensor panel.



Parameter	Description
X Offset	Specifies the shift along the X axis. A positive value shifts the data to the right.
Y Offset	Specifies the shift along the Y axis.
Z Offset	Specifies the shift along the Z axis. A positive value shifts the data toward the sensor.
Angle X	Specifies the tilt around the X axis.
Angle Y	Specifies the tilt around the Y axis.
Angle Z	Specifies the tilt around the Z axis.

When applying the transformations, the object is first rotated around X, then Y, and then Z, and then the offsets are applied.

	Setting **Angle X** or **Angle Z**, and to a lesser extent **Y Offset**, to a non-zero value increases CPU usage when scanning, which reduces the maximum scan speed.

	Artifacts may appear in scan data when **Angle Z** or **Angle X** is set to a non-zero value if encoder trigger spacing is set too high (resulting in a low sampling rate).

To configure transformation settings:

1. Go to the **Scan** page.
2. Choose a mode other than Video mode in the **Scan Mode** panel.
   If Video mode is selected, you will not be able to change the settings.
3. Expand the **Sensor** panel by clicking on the panel header.
4. Expand the Transformations area by clicking on the expand button.
   See the table above for more information.
5. Set the parameter values.
   See the table above for more information.

	The Y offset, X angle, and Z angle transformations cannot be non-zero when **Uniform Spacing** is unchecked. Therefore, when aligning a sensor using a bar alignment target with **Uniform Spacing** unchecked, set the **Degrees of Freedom** setting to **X, Z, Y Angle**, which prevents these transformations from being non-zero.

6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the transformation settings are applied correctly after the sensor is restarted.

Exposure

Exposure determines the duration of camera and light-source on-time. Longer exposures can be helpful to detect light on dark or distant surfaces, but increasing exposure time decreases the maximum speed. Different target surfaces may require different exposures for optimal results. Sensors provide two exposure modes for the flexibility needed to scan different types of target surfaces.

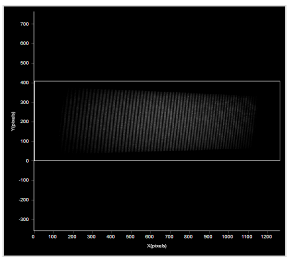
	Due to sensor architecture, exposure values provided by the user in the interface are divided by a factor of 1.024 internally. So for example, setting an exposure value of 1000 µs results in the sensor using a 977 µs exposure internally. This, in addition to various overhead factors, can result in a discrepancy between Max Frame Rate displayed on the **Trigger** panel and the speed reported in the metrics area, but this is only obvious at higher frame rates.

	If you are running a 3210 sensor on 24 VDC, see*Running Gocator 3210 on 24 VDC* on page 100 for information on properly setting exposure.

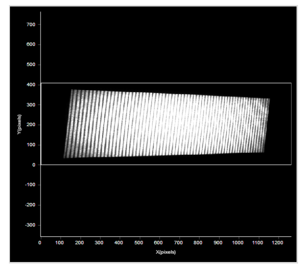
Exposure Mode	Description
Single	Uses a single exposure for all objects. Used when the surface is uniform and is the same for all targets.
Multiple	Uses multiple exposures to create a single scan. Used when the target surface has a varying reflectance within a single scan (e.g., white and black).

For more information on the different types of exposure options, see the sections below.

Video mode lets you see how the light appears on the camera and identify any stray light or ambient light problems. When exposure is tuned correctly, the projected light should be clearly visible along the entire length of the viewer. If it is too dim, increase the exposure value; if it is too bright decrease exposure value.

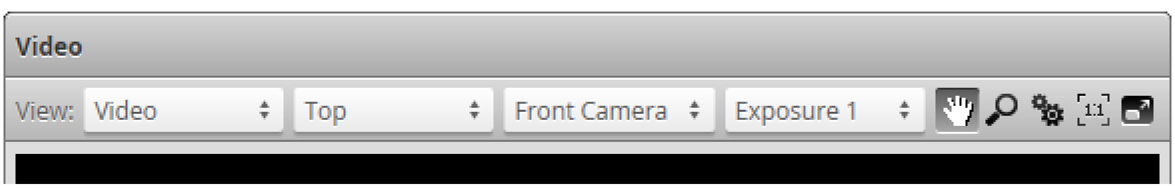


Under-exposure:
Light pattern is not fully detected.
Increase the exposure value.



Over-exposure:
Light pattern is saturated in the center.
Decrease the exposure value.

When the sensor is in Multiple exposure mode, select which exposure to view using the second drop-down box next to "View" in the data viewer. This drop-down is only visible in Video scan mode when the **Multiple** option is selected in the **Exposure** section in the **Sensor** panel.



When the sensor is set to Video mode, a **Pattern** parameter is available. (When this parameter is set to **Standard Sequence**, a **Pattern Index** parameter is also available.)

Pattern Settings

Parameter	Description
Pattern	One of the following: **Default**: The default projector pattern. **Focus**: A special pattern used to help focus the sensor. For more information, see *Using the Focus Pattern* on the next page. **Standard Sequence**: Enables the Pattern Index parameter (see below).

Parameter	Description
Pattern Index	The index of the pattern sequence to display. Choose the pattern that produces the best data. The indices represent Phase Pattern Sequences, followed by Stripe Pattern Sequences in reverse order. The lower indices are the higher frequency phase code patterns, and the higher indices are the lower frequency binary patterns. Index 1 [Phase Pattern Sequence Image 5]: Highest frequency sinusoid. Index 2 [Phase Pattern Sequence Image 4] [...] Index 5 [Phase Pattern Sequence Image 1]: Lowest frequency sinusoid. Index 6 [Stripe Pattern Sequence Image 7]: Highest bar count. Index 7 [Stripe Pattern Sequence Image 6] [...] Index 12 [Stripe Pattern Sequence Image 1]: Lowest bar count) Index 13 [Reference Image 1]

Using the Focus Pattern

You can use a special "line" focus pattern to aid in focusing the sensor. A focused sensor produces more accurate scans, which in turn results in more reliable measurements.

To use the focus pattern:

1. Place a representative target in view of the sensor.

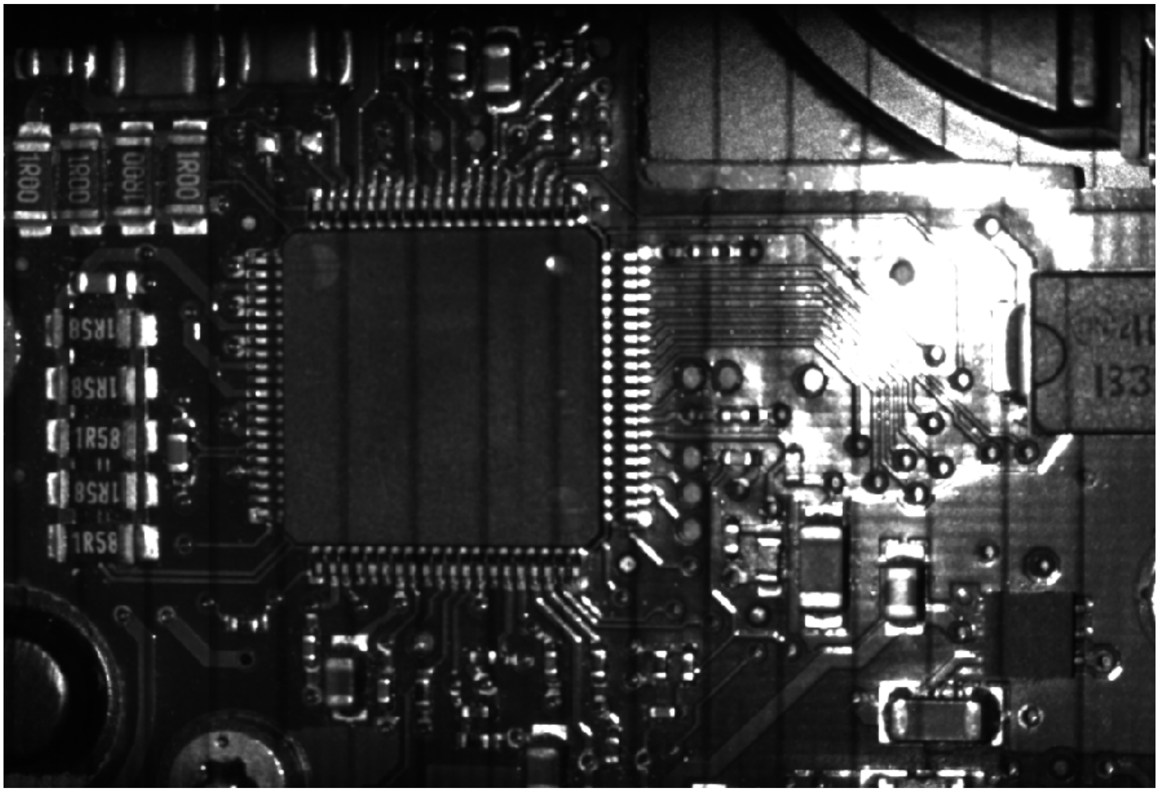
   The target surface should be similar to the material that will normally be measured.

2. Go to the **Scan** page and choose Video mode.
3. Expand the **Sensor** panel by clicking on the panel header or the button.
4. Click the **Exposure** tab.
5. In **Pattern**, choose **Focus**.

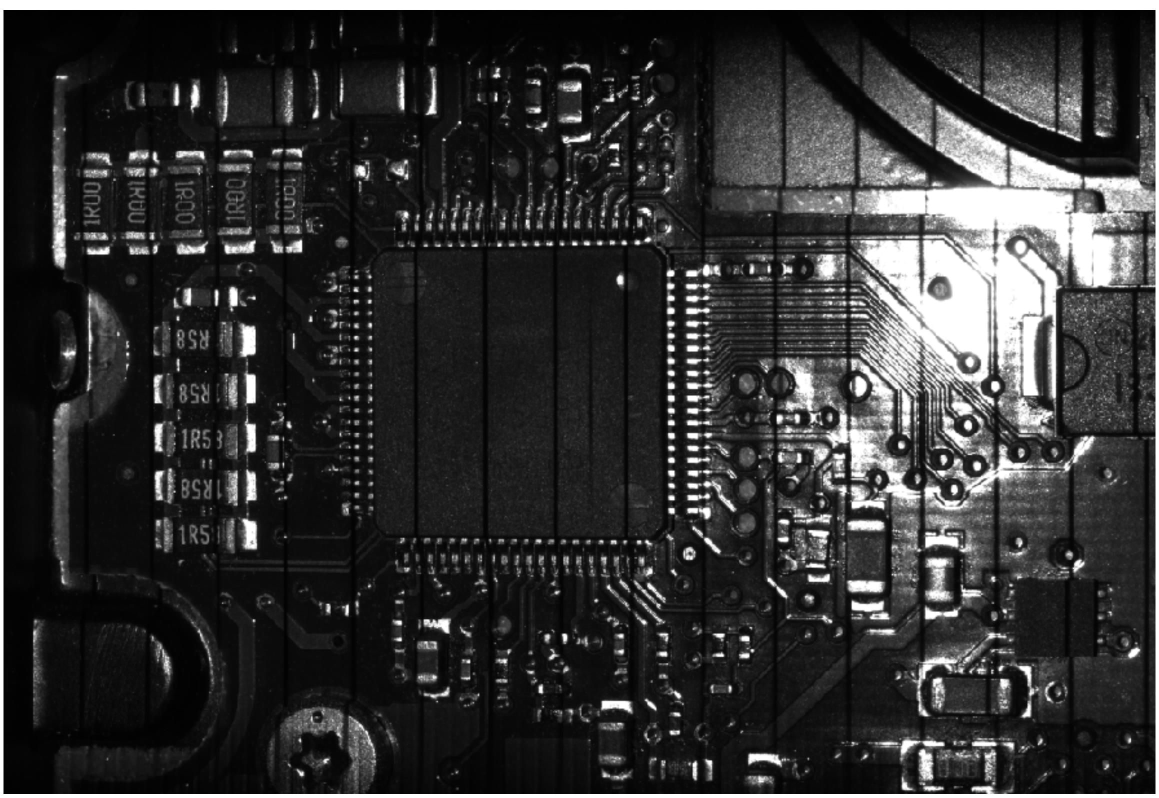
   A vertical line pattern is projected onto the surface (see next).

6. Move the sensor up and down until the dark lines are as sharp as possible.

   In the following image, the vertical lines are blurry:



After focusing, the lines are more sharply defined:

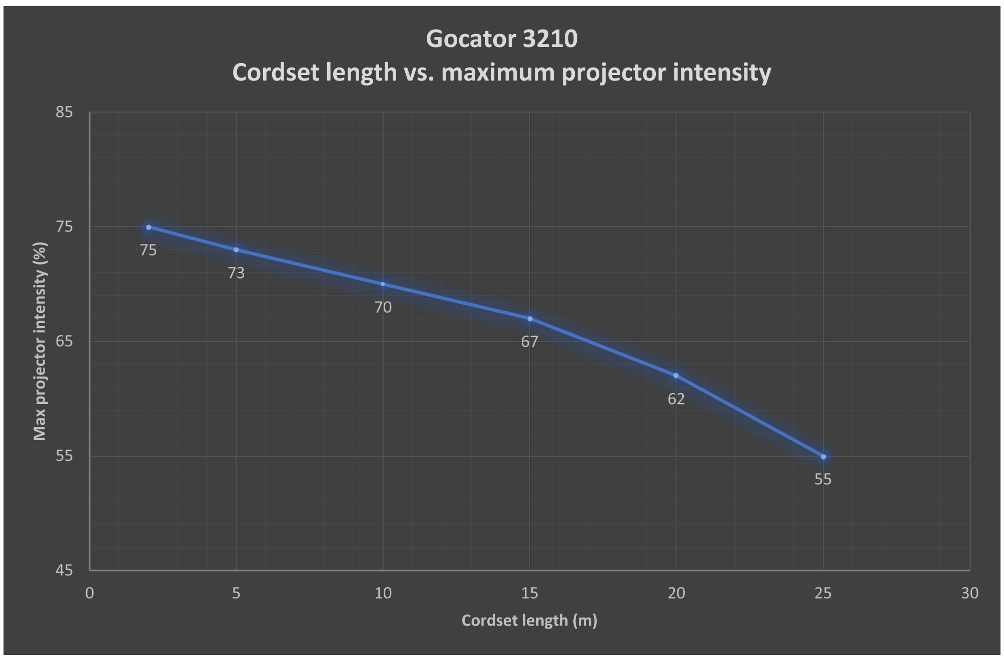


Running Gocator 3210 on 24 VDC

As of Gocator 4.7, Gocator 3210 can run at 24 VDC, for cases where this voltage is required, for example, by regulations. However, lowering the operating voltage of 3210 sensors from 48 VDC causes an increased current flow at constant power draw. The increased current will trip the sensor at 1.1 amps. At maximum projector intensity the power draw is the highest and will trip the sensor. Also, longer cordsets introduce additional resistance, leading to increased current, again at constant power draw.

To allow operation of the sensor at 24 VDC, the power draw must be lowered to accommodate the increased current flow, and because the LED projector is the highest consumer of power, the projector intensity must be lowered to limit the current flow.

The following chart plots maximum projector intensity versus cordset length:



You must consider the impact of this limitation on a case-by-case basis, but in general, scanning dark targets that already require longer exposure times will require additional exposure time. Shiny metal objects will generally not be affected, as the sensor does not run at full projector intensity.

The following table shows the exposure times *up to which* 24 VDC and 48 VDC operation is identical:

Cable Length	5 m	10 m	15 m	20 m	25 m	40 m
Exposure time (ms)	2.98	2.86	2.73	2.53	2.24	1.35

For longer exposure times, the sensor requires new exposure settings to accommodate for the different maximum projector intensity. For information on setting exposure, see the following sections.

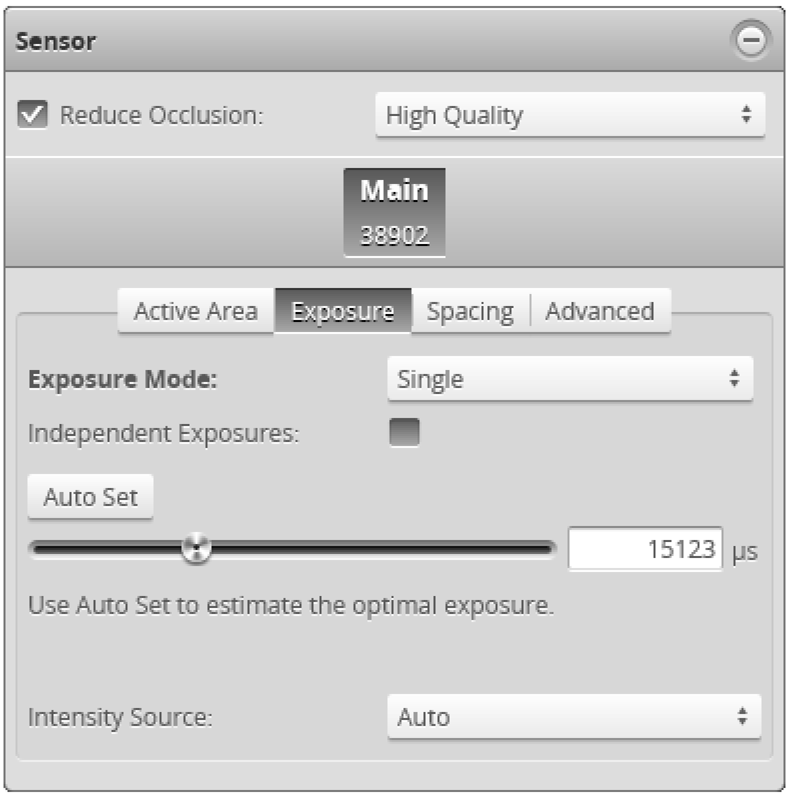
Single Exposure

The sensor uses a fixed exposure in every scan. Single exposure is used when the target surface is uniform and is the same for all targets.

See the note in *Exposure* on page 96 for important information on potential discrepancies between Max Frame Rate and the speed reported in the metrics area.

When the exposure mode is set to **Single**, you can optionally configure the sensor to allow setting exposure individually for each camera. This can be useful, for example, when reflections affect only camera: you could lower the exposure setting for that camera to improve scan data from that camera. (Automatic setting of exposure is not available when you this option is enabled.)

You can also choose which camera is used as the source for intensity data: if the front camera is typically affected by reflections off a part but not the back camera, you could choose the back camera as the source for intensity data.



To enable single exposure:

1. Place a representative target in view of the sensor.

   The target surface should be similar to the material that will normally be measured.

2. Go to the **Scan** page.
3. Expand the **Sensor** panel by clicking on the panel header or the button.
4. Click the **Exposure** tab.
5. Select **Single** from the **Exposure Mode** drop-down.
6. (Optional) Enable **Independent Exposure**.
7. Edit the exposure setting by using the slider or by manually entering a value.

You can automatically tune the exposure by pressing the **Auto Set** button, which causes the sensor to turn on and tune the exposure time. **Auto Set** is not available when **Independent Exposure** is enabled.

8. (Optional) Choose the source for intensity in **Intensity Source**.

   Auto: The default value. Video from both cameras is used to create intensity data.
   Front Camera: Video from the front camera is used to create intensity data.
   Back Camera: Video from the back camera is used to create intensity data.

9. Run the sensor and check that 3D data acquisition is satisfactory.

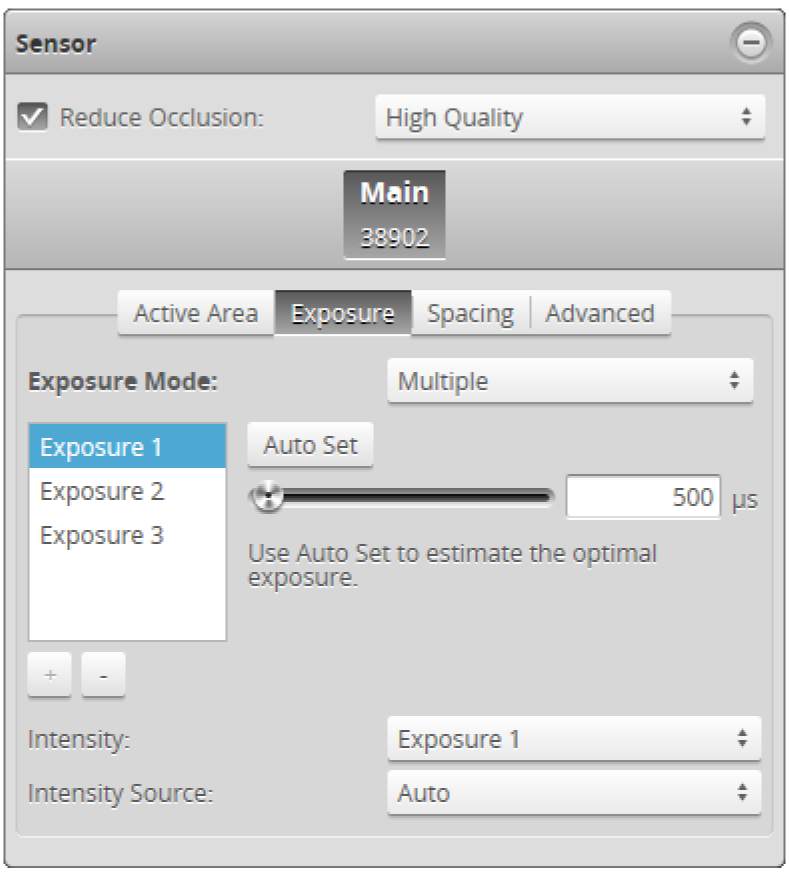
Multiple Exposure

The sensor combines data from multiple exposures to create a single 3D point cloud. Multiple exposures can be used to increase the ability to detect light and dark materials that are in the field of view simultaneously.

	See the note in *Exposure* on page 96 for important information on potential discrepancies between Max Frame Rate and the speed reported in the metrics area.

Up to three exposures can be defined with each set to a different exposure level. For each exposure, the sensor will perform a complete scan at the current frame rate making the effective frame rate slower. For example, if two exposures are selected, then the speed will be half of the single exposure frame rate. The sensor will perform a complete multi-exposure scan for each external input.

The resulting 3D point cloud is a composite created by combing data collected with different exposures.



If you have enabled intensity in the **Scan Mode** tab, you can use the **Intensity** setting to choose which of the exposures the sensor uses for acquiring intensity data. This lets you choose the exposure that produces the best image for intensity data.

You can also choose which camera is used as the source for intensity data using the **Intensity Source** setting: if the front camera is typically affected by reflections off a part but not the back camera, you could choose the back camera as the source for intensity data.

To enable multiple exposure:

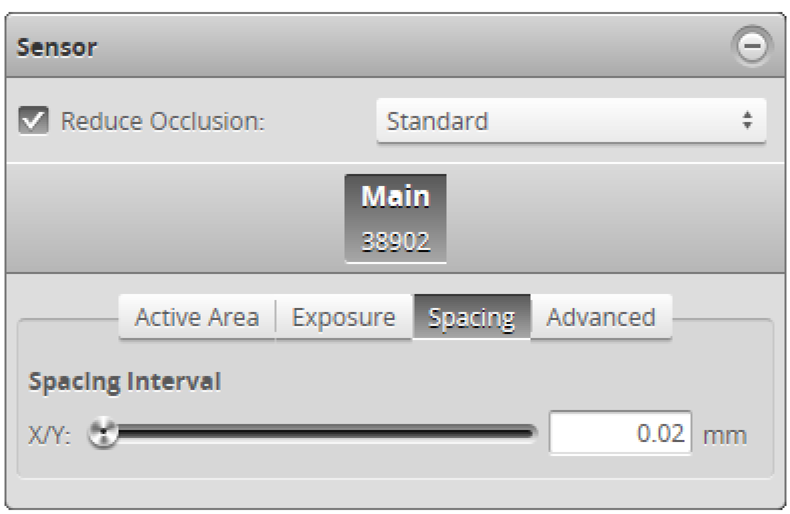
1. Go to the **Scan** page.
2. Expand the **Sensor** panel by clicking on the panel header or the ⊕ button.
3. Click the **Exposure** tab.
4. Select **Multiple** from the **Exposure Mode** drop-down.
5. Click the + button to add an exposure step.

   Up to a maximum of three exposure settings can be added.
   To remove an exposure, select it in the exposure list and click the - button
6. Set the exposure level for each exposure to make the sensor's camera less or more sensitive, as required.
7. If **Acquire Intensity** is enabled in **Scan Mode**, select the exposure that is used to capture the intensity output.
8. Run the sensor and check that 3D data acquisition is satisfactory.

   If 3D data acquisition is not satisfactory, adjust the exposure values manually. Switch to **Video** mode to use video to help tune the exposure; see *Exposure* on page 96 for details.

Spacing

The **Spacing** tab lets you configure the spacing interval.



Spacing Interval

Spacing interval is the spacing between data points in resampled data. A larger interval creates scans with lower X/Y resolution, reduces CPU usage, and potentially increases the maximum frame rate. A larger interval also reduces the data output rate. For more information on resampled data, see *Resampling* on page 47.

To configure the spacing interval:

1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.

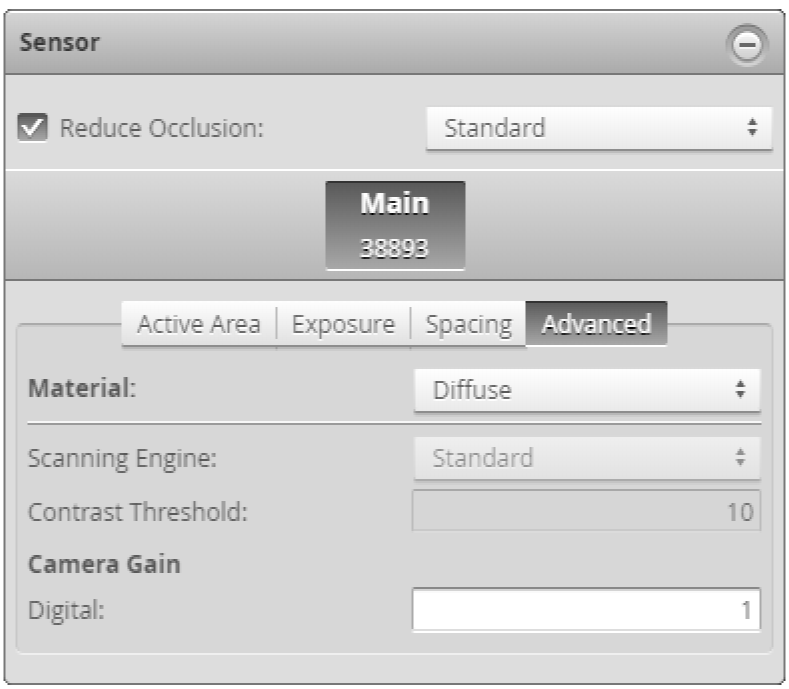
   If this mode is not selected, you will not be able to configure the spacing interval.
3. Expand the **Sensor** panel by clicking on the panel header or the ⊕ button.
4. Click the button corresponding to the sensor you want to configure.

   The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.

   Spacing is configured separately for each sensor.
5. Click the **Spacing** tab.
6. Do one of the following:
7. Select a spacing interval level.
8. Save the job in the **Toolbar** by clicking the **Save** button.

Advanced

The **Advanced** tab contains settings to configure camera gain and material characteristics.



To configure advanced settings:

1. Go to the **Scan** page.
2. Switch to Video mode.

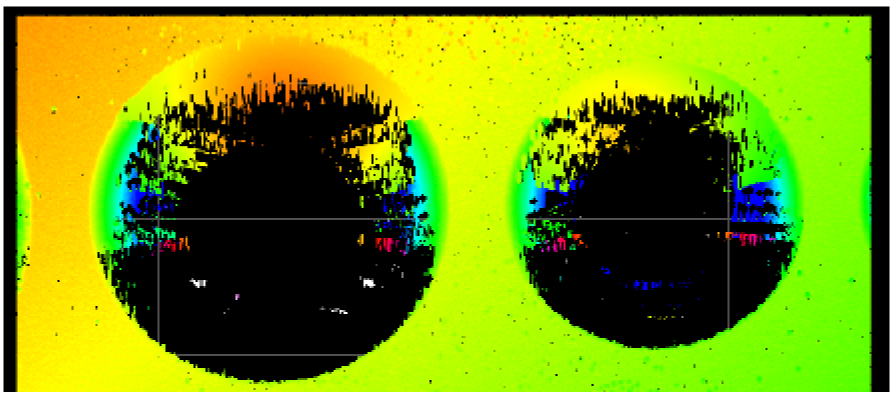
Using Video mode while configuring the settings lets you evaluate their impact.

3. Expand the **Sensor** panel by clicking on the panel header or the button.
4. Click on the **Advanced** tab.
5. Configure material characteristics and camera gain.
   For more information, see *Material* below and *Material Settings* on the next page.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that scan data is satisfactory.

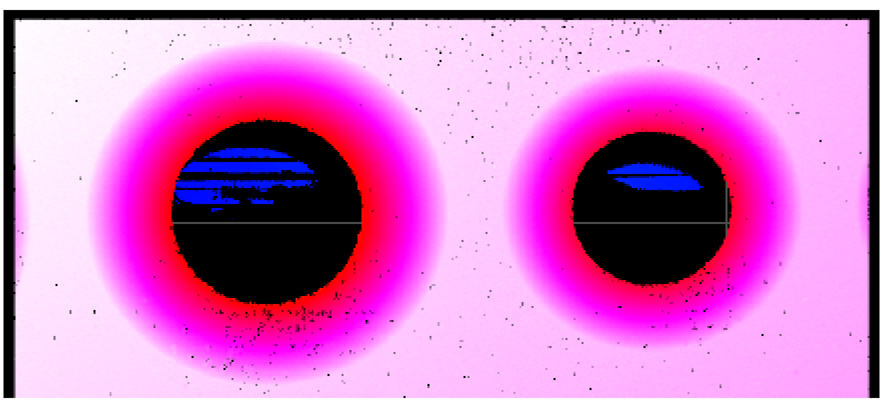
Material

Data acquisition can be configured to suit different types of target materials. For many targets, changing the setting is not necessary, but it can make a great difference with others.

You can select preset material types in the **Materials** setting under the **Advanced** tab. The **Diffuse** material option is suitable for most materials. The **Interreflective** material option is useful with targets whose surface may show reflections from other surfaces in the scanned area, for example inside a countersunk hole or on surfaces with concave features.



Countersunk holes with Material set to Diffuse



Countersunk holes with Material set to Interreflective

Currently, choosing **Custom** under the **Material** setting displays options identical to the

	Interreflective and **Diffuse** options under **Material**. Further customizations will be provided in the future.

🗋	A special G3 firmware that optimizes the on-sensor performance of the interreflective scan engine is available for download on LMI's Download center (https://downloads.lmi3d.com/). This firmware results in faster takt time for challenging targets. Note that the **Diffuse** material option (see above) will not be available: only Interreflective material will be available. In order to use the optimized engine, you must install the firmware on the sensor; for more information, see *Firmware Upgrade* on page 83.

Material Settings

You can set camera gain to improve data acquisition.

Setting	Description
Camera Gain	
	Digital camera gain can be used when the application is severely exposure limited, yet dynamic range is not a critical factor.
Scanning Engine	One of the following: **Standard** or **Interreflective**.
Contrast Threshold	Controls the contrast threshold. Allows tuning the detection of points based on the intensities observed in Video mode.

Alignment

Alignment procedures are required to compensate for sensor mounting inaccuracies or to set a Z (height) reference plane.

🗋	Sensors are pre-calibrated and ready to deliver data in engineering units (mm) out of the box. Alignment procedures do not affect sensor calibration.

A sensor can be in one of two alignment states: Unaligned and Aligned. An indicator on the **Alignment** panel display UNALIGNED or ALIGNED, depending on the sensor's state.

Alignment State

State	Explanation
Unaligned	The sensor or sensor system is not aligned. Data points are reported in sensor coordinates.
Aligned	The sensor is aligned using the alignment procedure (described below) or by manually modifying the values under **Transformation** in the **Sensor** tab on the **Scan** page (for more information, see *Transformations* on page 95). Data points are reported in system coordinates.

Once the alignment procedure has completed, the derived transformation values are displayed under **Transformations** in the **Sensor** panel.

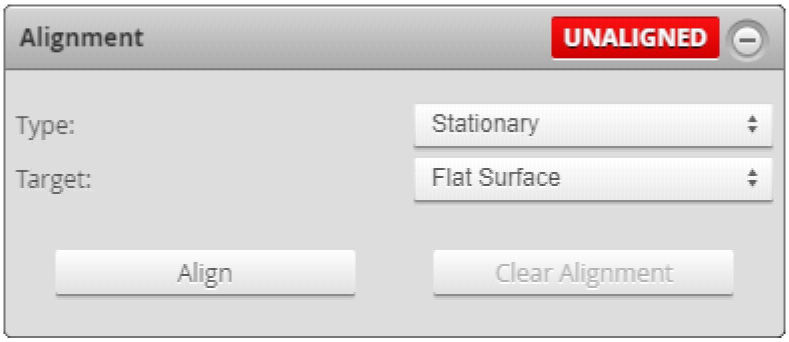
Alignment Types

Snapshot sensors only support Stationary alignment.

Type	Description
Stationary	**Stationary** is used when the alignment target does not move. This type of alignment compensates for Z offsets and X and Y angles, and optionally X and Y offsets and Z angle.

Aligning Sensors

Alignment is configured and performed using the **Alignment** panel.



To prepare for alignment:

1. Choose an alignment reference in the **Manage** page if you have not already done so.
   For more information, see *Alignment Reference* on page 78.
2. Go to the **Scan** page.
3. Choose a mode other than Video mode in the **Scan Mode** panel.
4. Expand the **Alignment** panel by clicking on the panel header or the button.
5. Ensure that all sensors have a clear view of the target surface.
   Remove any irregular objects from the sensor's field of view that might interfere with alignment.

To perform stationary alignment:

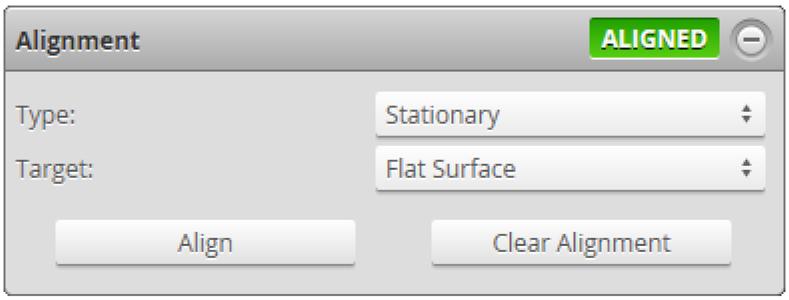
1. In the **Alignment** panel, select **Stationary** as the **Type**.
2. Clear the previous alignment if present.
   Press the **Clear Alignment** button to remove an existing alignment.
3. Select an alignment **Target**.
4. Click the **Align** button.
   The sensors will start, and the alignment process will take place. Alignment is performed simultaneously for all sensors. If the sensors do not align, check and adjust the exposure settings (page 96).

Alignment uses the exposure defined for single exposure mode, regardless of the current exposure mode.

5. Check the alignment results under **Transformation** in the **Active Area** tab in the **Sensor** panel.

Clearing Alignment

Alignment can be cleared to revert the sensor to sensor coordinates.



To clear alignment:

1. Go to the **Scan** page.
2. Expand the **Alignment** panel by clicking on the panel header or the button.
3. Click the **Clear Alignment** button.
   The alignment will be erased and sensors will revert to using sensor coordinates.

Filters

Filters are used to post-process scan data along the X or Y axis to remove noise or clean it up *before* it is output or is used by measurement tools.

In some situations, such as when **Uniform Spacing** is disabled or when a sensor does not support filters, the filters panel is not displayed.

The following types of filters are supported:

Filter	Description
Gap Filling	Fills in missing data caused by occlusions using information from the nearest neighbors. Gap filling also fills gaps where no data is detected, which can be due to the surface reflectivity, for example dark or specular surface areas, or to actual gaps in the surface.
Median	Substitutes the value of a data point with the median within a specified window around the data point.
Smoothing	Applies moving window averaging to reduce random noise.
Decimation	Reduces the number of data points.

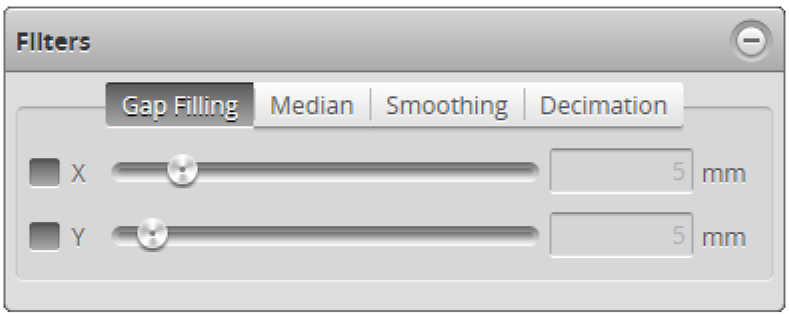
Filters are applied in the order displayed in the table above. The filters are configured in the **Filters** panel on the **Scan** page.

For more information, see the following sections.

Gap Filling

Gap filling works by filling in missing data points using either the lowest values from the nearest neighbors or linear interpolation between neighboring values (depending on the Z difference between neighboring values), in a specified X or Y window. The sensor can fill gaps along both the X axis and the Y axis.

If both X and Y gap filling are enabled, missing data is filled along the X and Y axes at the same time, using the available neighboring data.



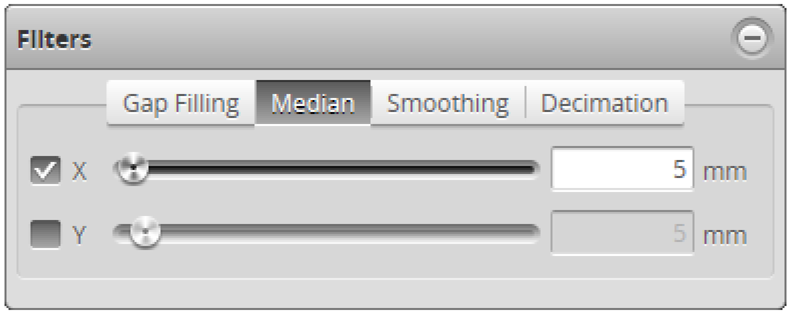
To configure X or Y gap filling:

1. Go to the **Scan** page.
2. Choose a mode other than Video in the **Scan Mode** panel.
   Otherwise, you will not be able to configure gap filling.
3. Expand the **Filters** panel by clicking on the panel header or the button.
4. Click on the **Gap Filling** tab.
5. Enable the **X** or **Y** setting and select the maximum width value.
   The value represents the maximum gap width that the sensor will fill. Gaps wider than the maximum width will not be filled.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the scan data is satisfactory.

Median

The Median filter substitutes the value of a data point with the median calculated within a specified window around the data point.

> Missing data points will not be filled with the median value calculated from data points in the neighbourhood.



To configure X or Y median:

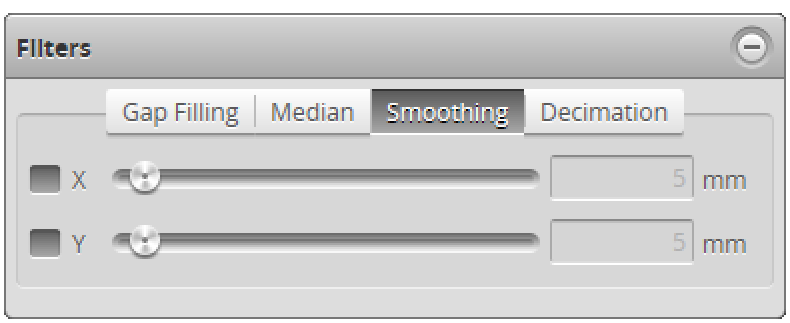
1. Go to the **Scan** page.
2. Choose a mode other than Video in the **Scan Mode** panel.
   Otherwise, you will not be able to configure the median filter.
3. Expand the **Filters** panel by clicking on the panel header or the ⊕ button.
4. Click on the **Median** tab.
5. Enable the **X** or **Y** setting and select the maximum width value.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the scan data is satisfactory.

Smoothing

Smoothing works by substituting a data point value with the average value of that data point and its nearest neighbors within a specified window. Smoothing can be applied along the X axis or the Y axis. X smoothing works by calculating a moving average across samples along the X axis. Y smoothing works by calculating a moving average along the X axis.

If both X and Y smoothing are enabled, the data is smoothed along X axis first, then along the Y axis.

	Missing data points will not be filled with the mean value calculated from data points in the neighbourhood.



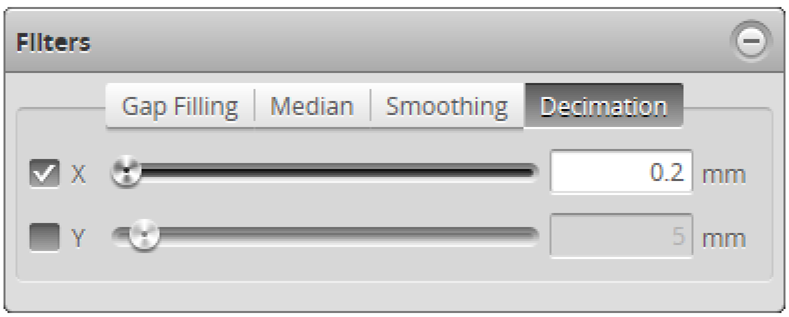
To configure X or Y smoothing:

1. Go to the **Scan** page.

2. Choose a mode other than Video in the **Scan Mode** panel.
   Otherwise, you will not be able to configure smoothing.
3. Expand the **Filters** panel by clicking on the panel header or the button.
4. Click on the **Smoothing** tab.
5. Enable the **X** or **Y** setting and select the averaging window value.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the scan data is satisfactory.

Decimation

Decimation reduces the number of data points along the X or Y axis by choosing data points at the end of a specified window around the data point. For example, by setting X to .2, only points every .2 millimeters will be used.



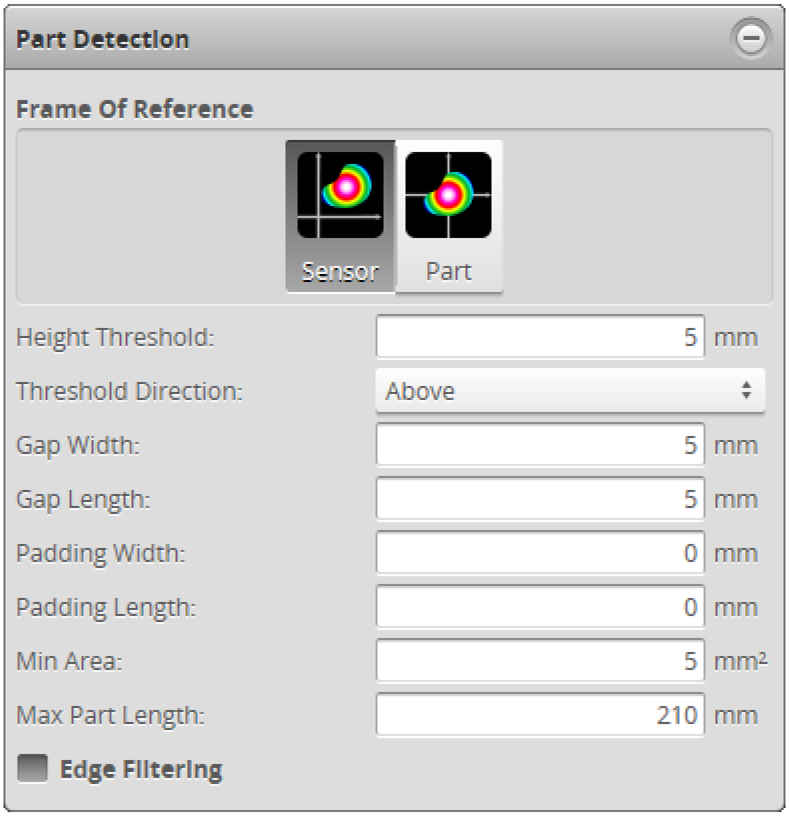
To configure X or Y decimation:

1. Go to the **Scan** page.
2. Choose a mode other than Video in the **Scan Mode** panel.
   Otherwise, you will not be able to configure the decimation filter.
3. Expand the **Filters** panel by clicking on the panel header or the button.
4. Click on the **Decimation** tab.
5. Enable the **X** or **Y** setting and select the decimation window value.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the scan data is satisfactory.

Part Detection

Multiple parts can be detected from a single surface and will be individually tracked.

> Gocator also lets you isolate and then measure using one of two Surface measurement tools (for more information on these tools, see *Blob* on page 289 and *Segmentation* on page 426). For a comparison of part detection and these tools, see *Isolating Parts from Surface Data* on page 277.



The following settings can be tuned to improve the accuracy and reliability of part detection.

Setting	Description
Height Threshold	Determines the height threshold for part detection. The setting for **Threshold Direction** determines if parts should be detected above or below the threshold. Above is typically used to prevent the belt surface from being detected as a part when scanning objects on a conveyor.
Threshold Direction	Determines if parts should be detected above or below the height threshold.
Gap Width	Determines the minimum separation between objects on the X axis. If parts are closer than the gap interval, they will be merged into a single part.
Gap Length	Determines the minimum separation between objects on the Y axis. If parts are closer than the gap interval, they will be merged into a single part.
Padding Width Padding Length	The amount of padding data added in the X and Y directions, respectively. The padding can contain data points that were outside the height threshold and excluded from the initial part detection. This is mostly useful when processing part data with third-party software such as HexSight, Halcon, etc.
Min Area	Determines the minimum area for a detected part. Set this value to a reasonable minimum in order to filter out small objects or noise.
Max Part Length	Determines the maximum length of the part object. When the object exceeds the maximum length, it is automatically separated into two parts. This is useful to break a long object into multiple sections and perform measurements on each section.

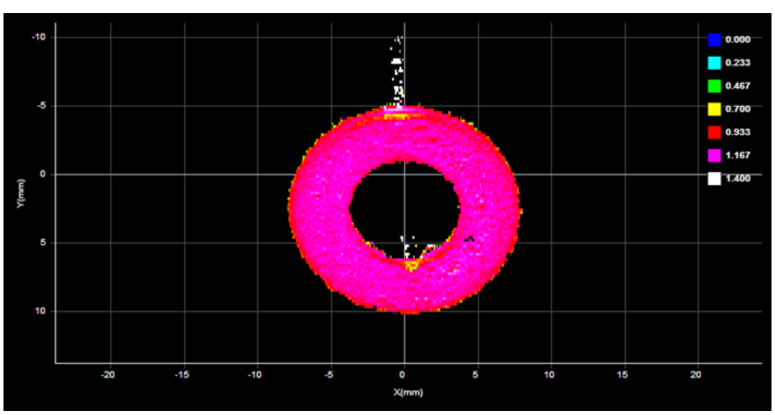
Setting	Description
Frame of Reference	Determines the coordinate reference for surface measurements. **Sensor** When **Frame of Reference** is set to **Sensor**, the sensor's frame of reference is used. When set to **Sensor**, all measurement values are relative to the sensor's field of view in X and relative to the encoder zero position in Y. **Part** When **Frame of Reference** is set to **Part**, all measurements except Bounding Box X and Y are relative to the center of the bounding box of the part. For Bounding Box X and Y, the measurement values are always relative to the sensor frame of reference (see *Bounding Box* on page 301).
Edge Filtering	See *Edge Filtering* below.

To set up part detection:

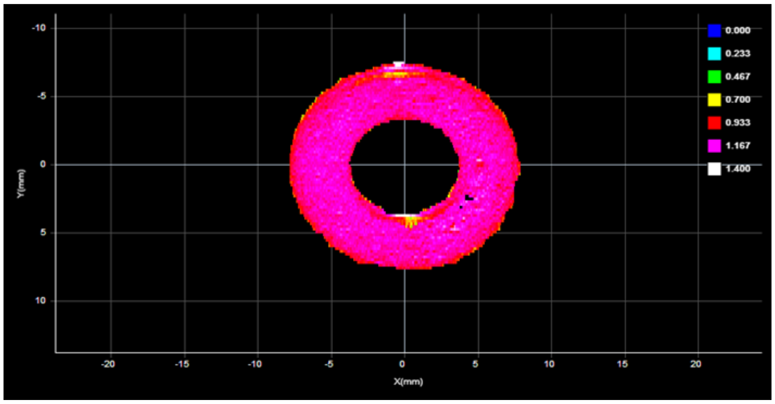
1. Go to the **Scan** page and choose **Surface** in the **Scan Mode** panel.
   If this mode is not selected, you will not be able to configure part detection.
2. Expand the **Part Detection** panel by clicking on the panel header or the button.
3. If necessary, check the **Enabled** option.
   When **Generation** is set to **Continuous**, part detection is always enabled.
4. Adjust the settings.
   See the part detection parameters above for more information.

Edge Filtering

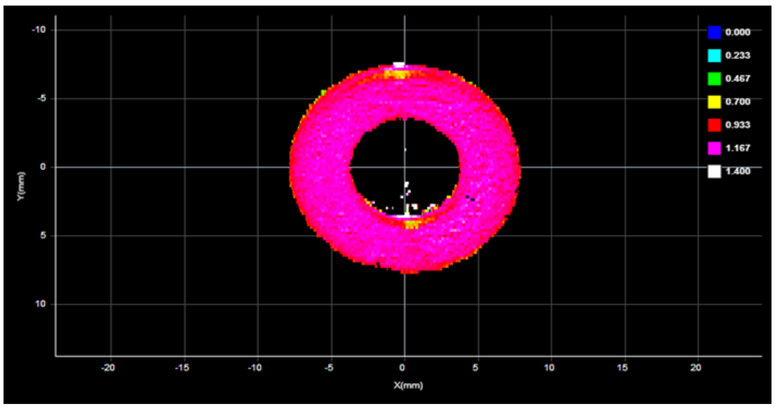
Part scans sometimes contain noise around the edges of the target. This noise is usually caused by the sensor's light being reflected off almost vertical sides, rounded corners, etc. Edge filtering helps reduce edge noise in order to produce more accurate and repeatable volume and area measurements, as well as to improve positioning of relative measurement regions. Optionally, the **Preserve Interior Feature** setting can be used to limit filtering to the outside edges of the target.



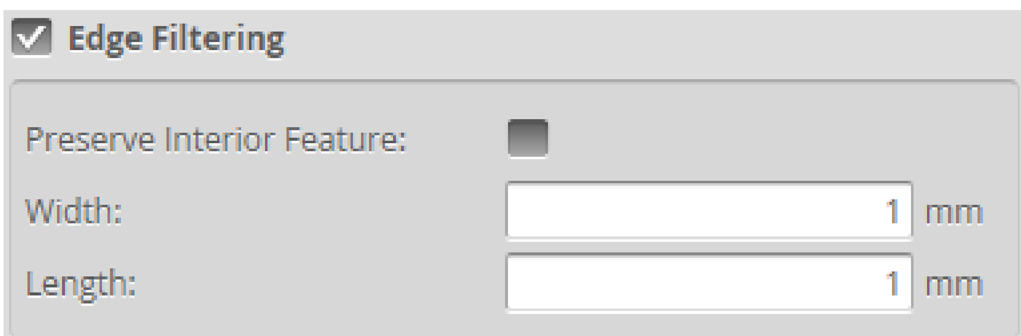
Edge Filtering disabled (scan shows reflection noise)



Edge Filtering enabled (reflection noise eliminated or reduced)



Edge Filtering enabled, Preserve Interior Feature enabled



To configure edge filtering:

1. Go to the **Scan** page and choose **Surface** in the **Scan Mode** panel.
   If this mode is not selected, you will not be able to configure part detection.
2. Expand the Part Detection panel by clicking on the panel header or the button and enable part detection if necessary.
3. Check the **Edge Filtering** checkbox to enable edge filtering.
4. Configure the **Width** and **Length** settings.
   The **Width** and **Length** settings represent the size of the filter on the X axis and the Y axis, respectively.

5. Set the **Preserve Interior Feature** setting if necessary.
   The **Preserve Interior Feature** setting limits filtering to the outside edges of the target.

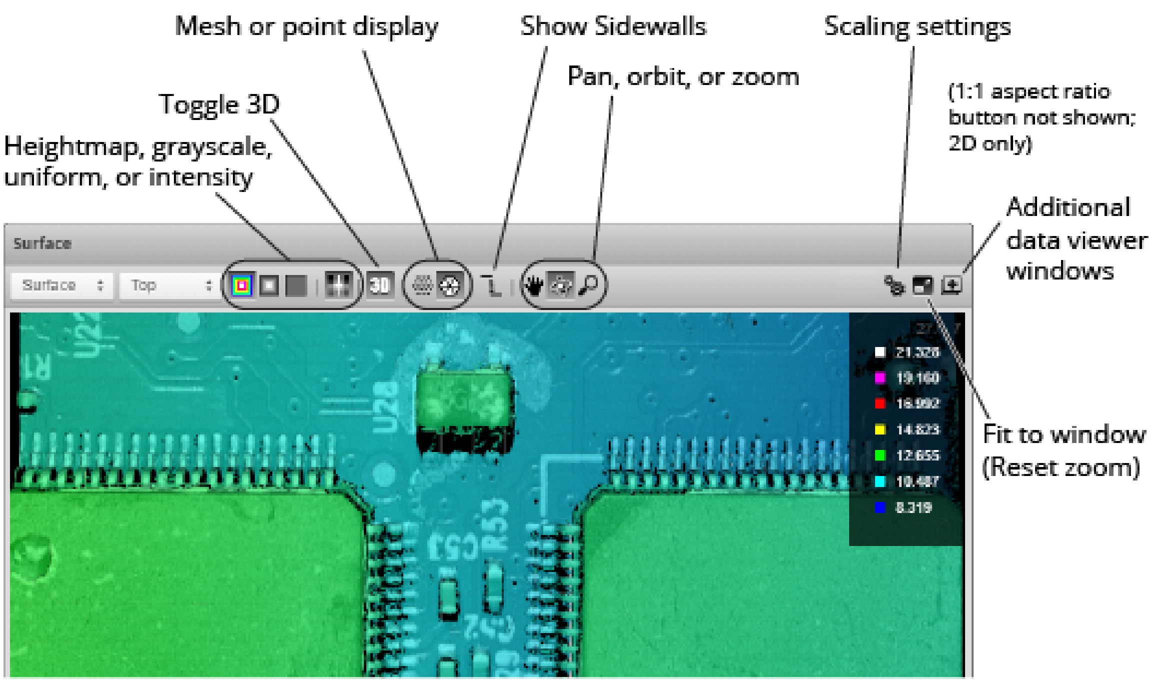
Data Viewer

The data viewer can display images in a 2D view and height map or intensity in 2D or 3D views, in addition to sections. The data viewer changes depending on the current operation mode and the panel that has been selected. Use the drop down list at the top-left corner to select the data source to view. The available data sources depend on the operation mode settings.

The data viewer lets you "pin" multiple outpus (measurements and geometric features) to the data viewer; for more information, see *Pinning Measurements and Features* on page 198.

Data Viewer Controls

The data viewer is controlled by mouse clicks and by the buttons on the display toolbar. The mouse wheel can also be used for zooming in and out.



For more information on the kinds of data displayed in Surface mode and how scan data is displayed, see *Surface Mode* on page 118.

For information on how to open and use additional data viewer windows, see *Using Multiple Data Viewer Windows* on page 147.

Video Mode

In Video mode, the data viewer displays images directly from the sensor's camera or cameras.

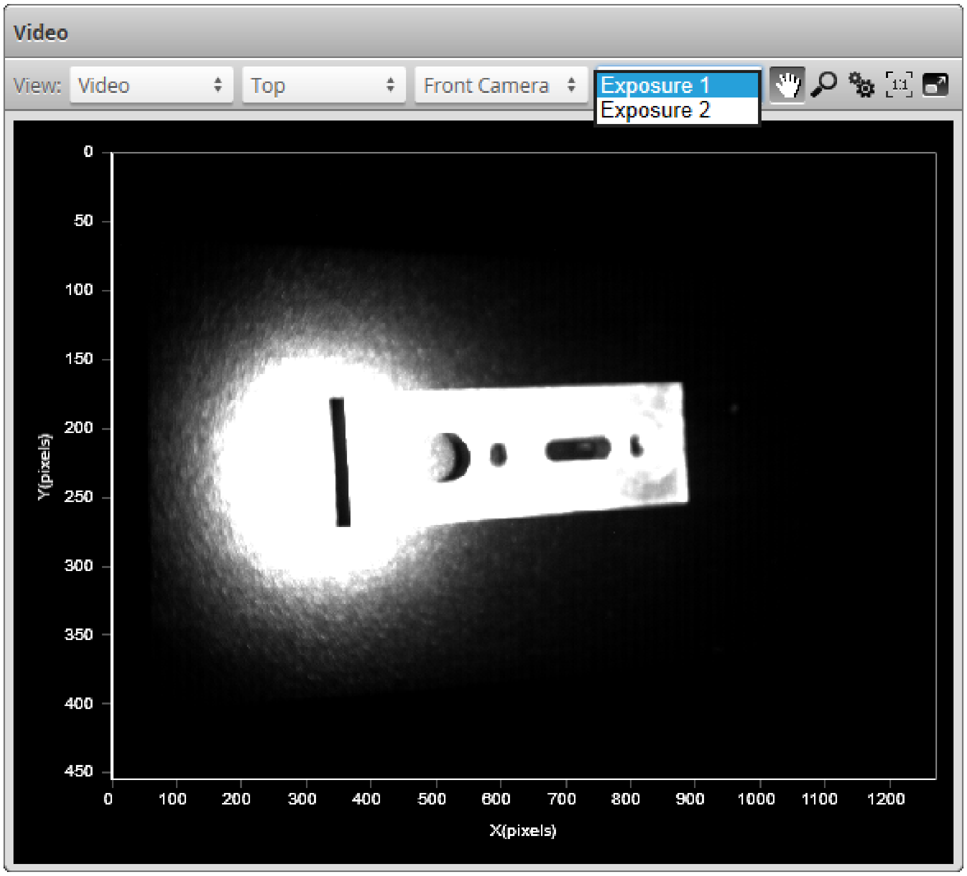
In this mode, you can configure the data viewer to display exposure, spot, and dropout information that can be useful in properly setting up the system for scanning.

Exposure Information

In Video mode, you can display exposure-related information. This information can help you correctly adjust the exposure settings.

Exposures

If you have set **Exposure Mode** to **Multiple**, and have set more than one exposure, a drop-down at the top of the data viewer lists the available exposures. Choosing an exposure changes the view of the data viewer to that exposure.



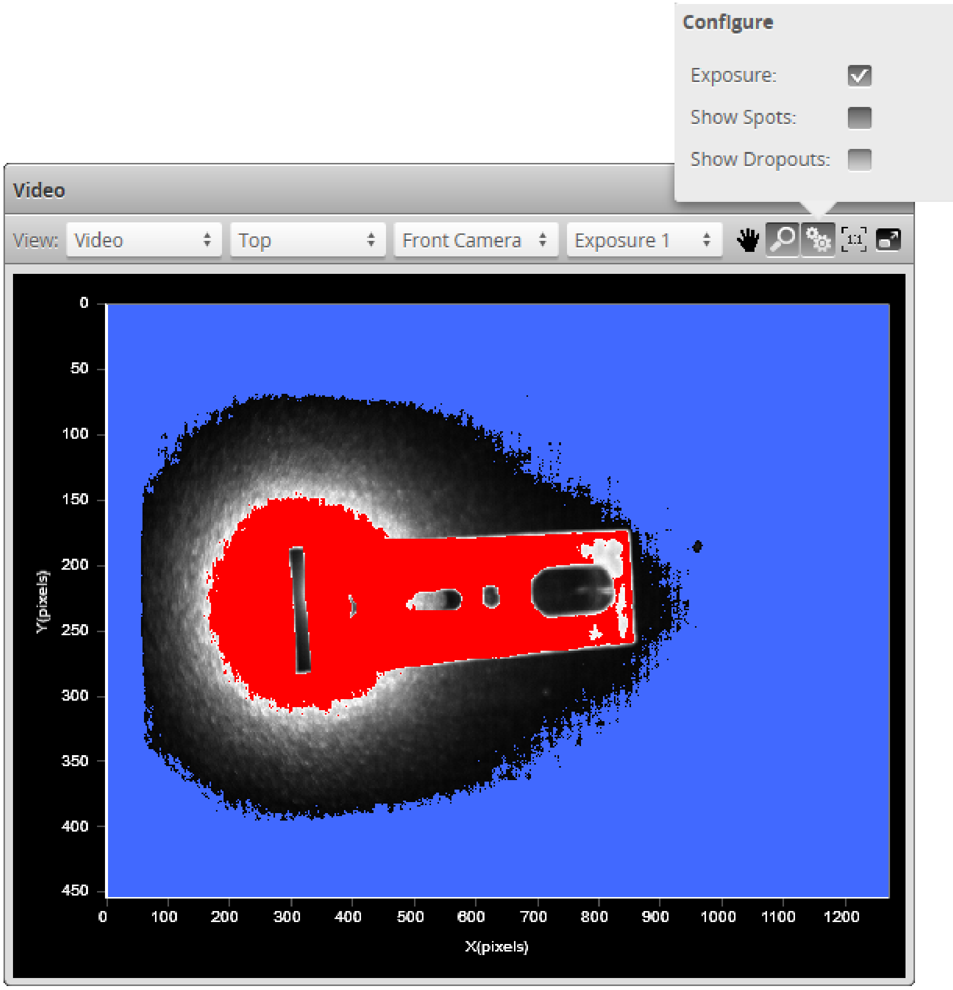
For details on setting exposure in the **Exposure** tab in the **Sensor** panel, see *Exposure* on page 96.

To select the exposure view of the display:

1. Go to the **Scan** page and choose **Video** mode in the **Scan Mode** panel.
2. Select the camera view in the data viewer.
   Use the first drop-down list next to **View** at the top of the data viewer to select **Front Camera** or **Back Camera**.
3. Select the exposure.
   Use the second drop-down list next to **View** at the top of the data viewer to select the exposure.

Overexposure and Underexposure

You can display a color exposure overlay on the video image to help set the correct exposure.



The **Exposure** setting uses the following colors:

- Blue: Indicates background pixels ignored by the sensor.
- Red: Indicates saturated pixels.

Correct tuning of exposure depends on the reflective properties of the target material and on the requirements of the application. Settings should be carefully evaluated for each application.

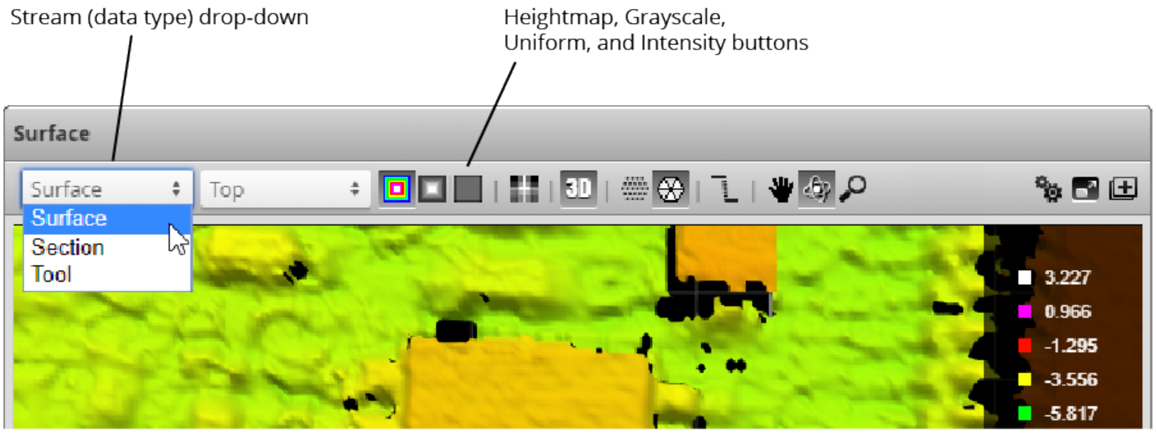
3x00 sensors cannot generate 3D points in over-saturated (areas indicated with red) or in under-exposed areas (indicated with blue). If it's not possible to set a single exposure to capture the entire object target without red areas appearing in the image, the **Multiple** exposure feature should be enabled. Use the drop-down selection box to view each exposure and tune one high exposure for dark areas on the target and one low exposure for bright areas on the target. Note that multiple exposures reduce the maximum speed the sensor can run at.

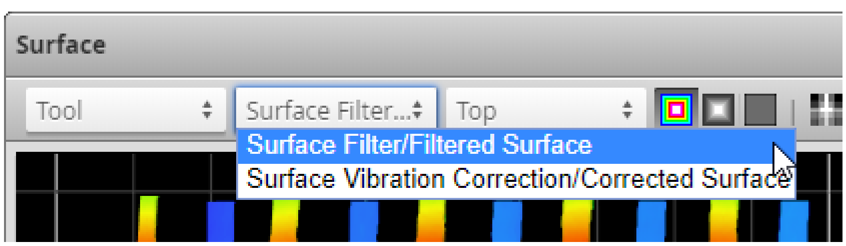
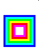
To display an overlay:

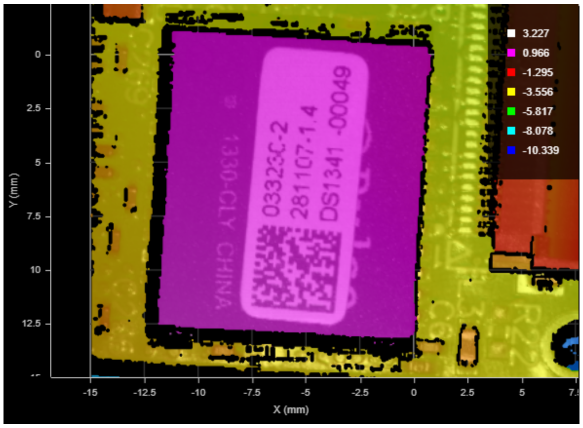
1. Go to the **Scan** page and choose **Video** mode in the **Scan Mode** panel.
2. Check **Exposure** at the top of the data viewer.

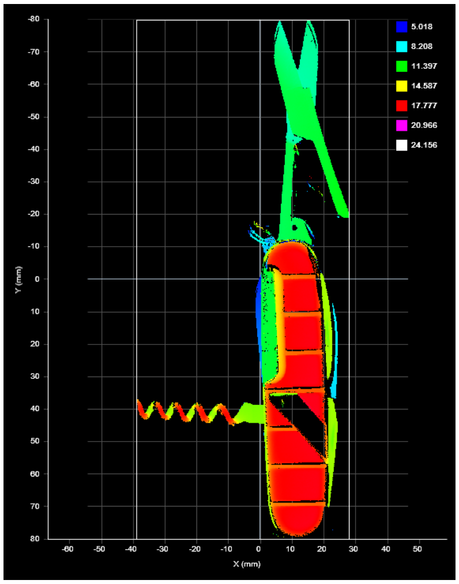
Surface Mode

When the sensor is in Surface scan mode, the data viewer can display height maps, sections, and intensity images. You can select the data to display from the first drop-down.



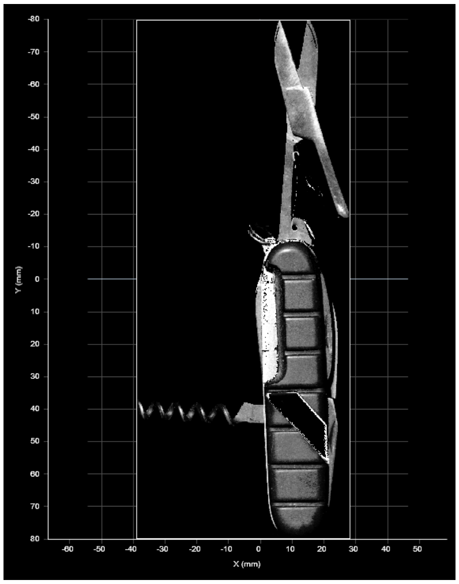
Data Type Option or Button	Description
Surface	Displays surface data received from the sensor's scan engine. If intensity data is available in the scan data, you can choose to display heightmap and intensity data at the same time to produce a more realistic part. For more information, see Heightmap button below.
Section	If any sections have been defined, displays the section selected in the Sections drop-down. (Only available in 2D view.)
Tool	Displays data from tools capable of producing "tool data" output (such as Surface Stitch or Surface Track). When you select **Tool**, a second drop-down is displayed next to the first, which lets you choose among the available data.  For more information on tool data output, see *Tool Data* on page 55.
Heightmap button 	Displays a pseudo-color height map over the scan data. If intensity data is available, you can use the Intensity button (see below) to display the combined heightmap and intensity data. This results in a more realistic-looking part in the data viewer and lets you use contrast-based information to help position tool regions. For more information on intensity data, see *Intensity Output* on page 125. By default, intensity is not enabled in the data viewer. For example, if you needed to measure the flatness of a CPU, this could help you avoid placing measurement regions on top of labels that are slightly raised compared to the surrounding area, which, if included in the flatness measurement, would result in inaccurate

Data Type Option or Button	Description
	measurements: 
Grayscale button 	If intensity data is available, when the Intensity button is toggled *off* (see below), this displays a grayscale height map. This is useful to better differentiate between scan data and the various elements of measurement tools that are displayed over the scan data. When the Intensity button is toggled *on*, displays intensity data only.
Uniform button 	Displays a uniformly shaded surface on the 3D model. (Only available in 3D view.) Mostly useful when you want to focus on shape or geometry. When this mode is selected, the Intensity button is hidden.
Intensity button 	Displays intensity data. See the descriptions of the Heightmap, Grayscale, and Uniform buttons above for an explanation of how this button interacts with those display modes. (The button is hidden if no intensity data is available in the scan data.) (**Acquire Intensity** must be checked in the **Scan Mode** panel for this button to be visible. For more information, see *Intensity Output* on page 125 and *Scan Modes* on page 88.)



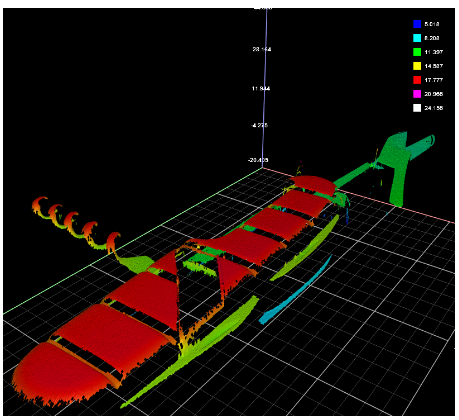
2D viewer with height map overlay

2D viewer with grayscale overlay

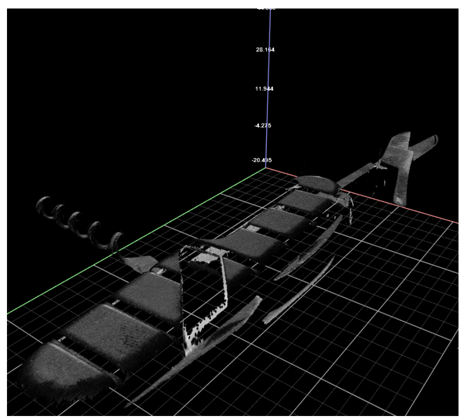


2D viewer with intensity overlay

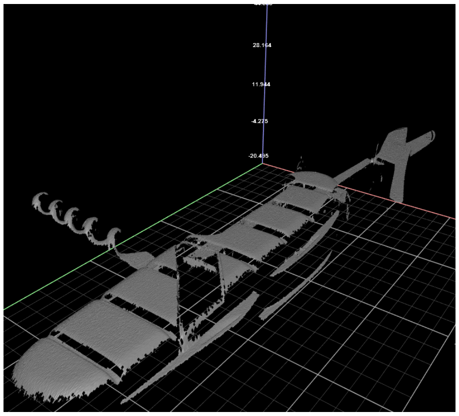
Clicking the **3D** button toggles between the 2D and 3D viewer. The 3D model is overlaid with the information that corresponds to the selected **View** option.



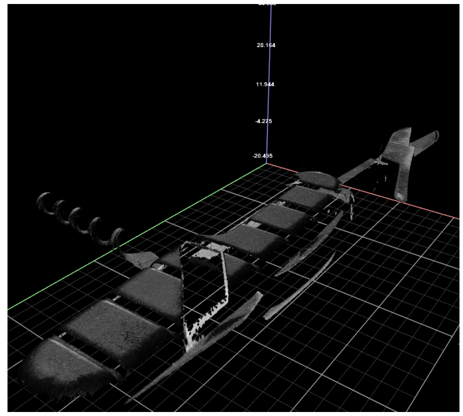
3D viewer with height map overlay



3D viewer with grayscale overlay

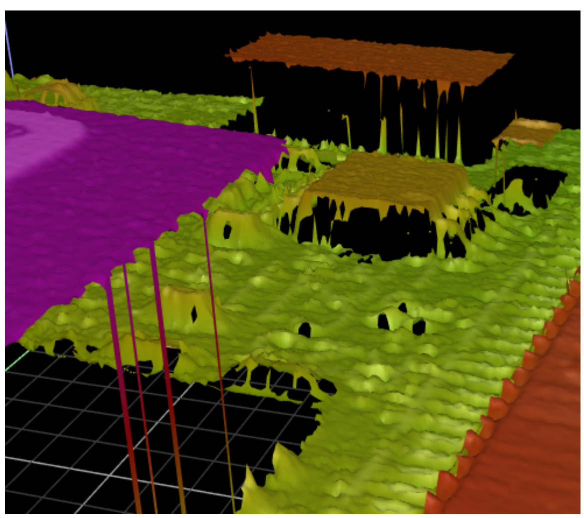


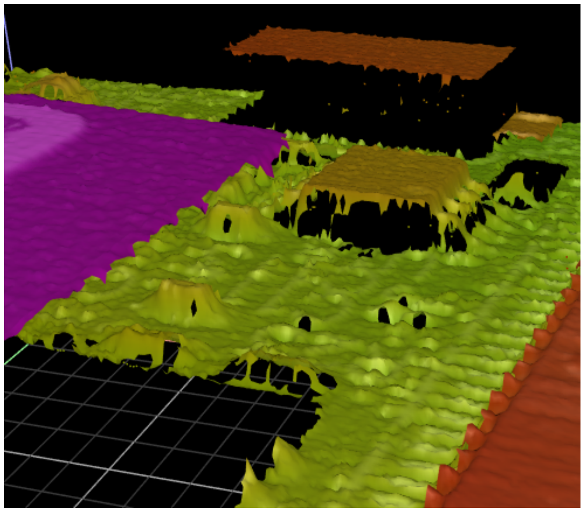
3D viewer with uniform overlay



3D viewer with uniform overlay

You can choose among the following options to change how the data viewer renders the scan data.

Rendering Mode	Description
Points 	Renders scan data using point. Useful in scan data that contains noise around edges, and can show hidden structure.
Mesh 	Renders scan by connecting points with polygons.
Show Sidewalls 	Toggles between hiding and showing polygons involving geometrically distant points. For example, in the following, the sidewalls are enabled: the long lines of scan data shown at the edges of the PCB components may be visually distracting.  These artifacts are hidden when "sidewalls" are disabled.

Rendering Mode	Description
	 Note that this setting only affects the *appearance* of scan data in the data viewer. It does not change the scan data and therefore does not affect measurements. In some situations, displaying long triangles may provide useful information. Try both modes in your application to determine the best choice.

Height Map Color Scale

Height maps are displayed in pseudo-color. The height axis (Z) is color-coded. The scaling of the height map can be adjusted.

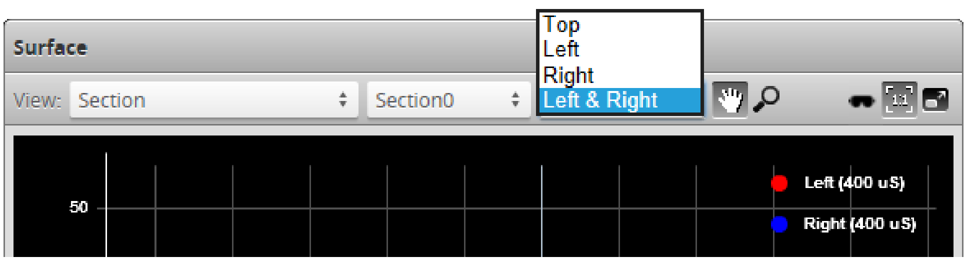
To change the scaling of the height map:

1. Select **Heightmap** from the **View** drop-down in the data viewer.
2. Click the **Scaling** button.
   - To automatically set the scale, choose **Auto** in the **Range** drop-down.
   - To automatically set the scale based on a user-selected sub-region of the heightmap, choose **Auto - Region** in the **Range** drop-down and adjust the yellow region box in the data viewer to the desired location and size.
   - To manually set the scale, choose the **Manual** in the **Range** drop-down and enter the minimum and maximum height to which the colors will be mapped.

Sections

When the sensor is in Surface scan mode, the data viewer can display sections (profiles extracted from surfaces).

When in the **Scan** page, selecting a panel (e.g., **Sensor** or **Alignment** panel) automatically sets the display to the most appropriate display view.



To manually select the display view in the Scan page:

1. Go to the **Scan** page.
2. Choose **Surface** mode in the **Scan Mode** panel.
3. Just above the data viewer, choose **Section** in the **View** drop-down.
   The view from an individual sensor or the combined view of two sensors can be selected from the drop-down list at the top of the data viewer.
   Top: View from a single sensor, from the top sensor in an opposite-layout dual-sensor system, or the combined view of sensors that have been aligned to use a common coordinate system.
   Bottom: View from the bottom sensor in an opposite-layout dual-sensor system.
   Left: View from the left sensor in a dual-sensor system.

Right: View from the right sensor in a dual-sensor system.
Left & Right: Views from both sensors, displayed at the same time in the data viewer, using the coordinate systems of each sensor.

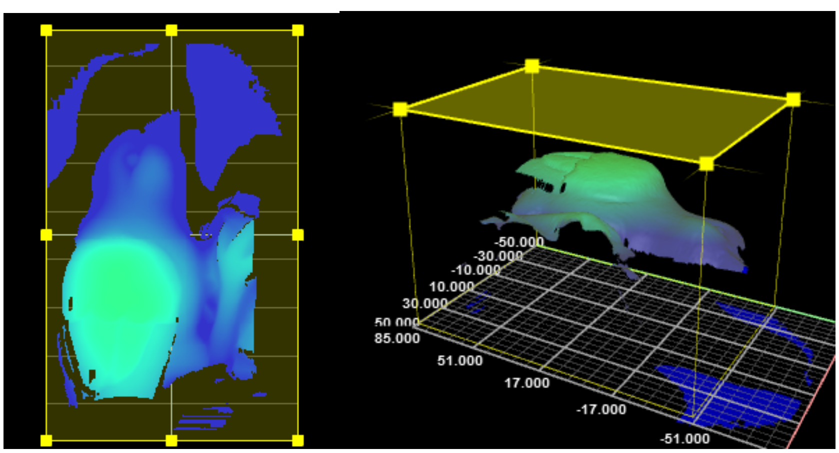
1. Go to the **Scan** page.
2. Choose **Surface** mode in the **Scan Mode** panel.
3. Just above the data viewer, choose **Section** in the **View** drop-down.
   The view from an individual sensor or the combined view of two sensors can be selected from the drop-down list at the top of the data viewer.
   Top: View from a single sensor, from the top sensor in an opposite-layout dual-sensor system, or the combined view of sensors that have been aligned to use a common coordinate system.
   Bottom: View from the bottom sensor in an opposite-layout dual-sensor system.
   Left: View from the left sensor in a dual-sensor system.
   Right: View from the right sensor in a dual-sensor system.
   Left & Right: Views from both sensors, displayed at the same time in the data viewer, using the coordinate systems of each sensor.

In the **Measure** page, the view of the display is set to the profile source of the selected measurement tool.

Region Definition

Regions, such as an active area or a measurement region, can be graphically set up using the data viewer in the 2D or in the 3D view.

When the **Scan** page is active, the data viewer can be used to graphically configure the active area. The **Active Area** setting can also be configured manually by entering values into its fields and is found in the **Sensor** panel (see *Sensor* on page 92).



To set up a region of interest:

1. Move the mouse cursor to the rectangle. In the 3D viewer, you must first select which side of the 3D rectangle to adjust by clicking on it.

The rectangle is automatically displayed when a setup or measurement requires an area to be specified.

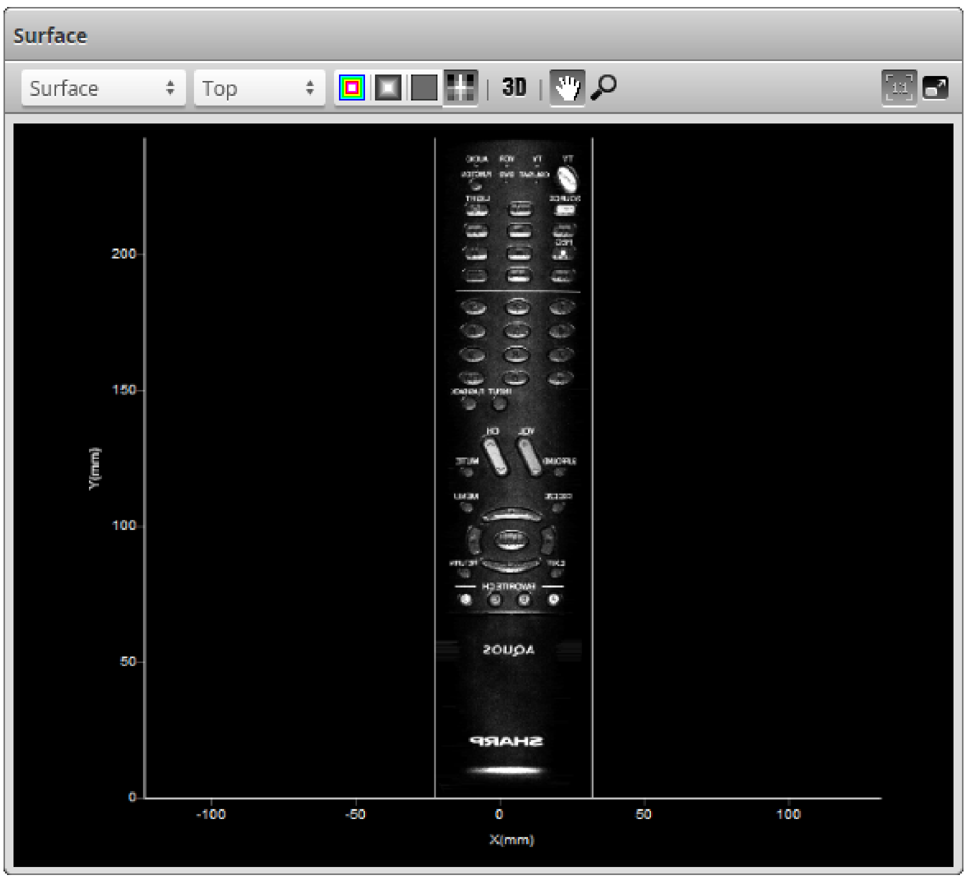
2. Drag the rectangle to move it, and use the handles on the rectangle's border to resize it.

Intensity Output

Sensors can produce intensity images that measure the amount of light reflected by an object. An 8-bit intensity value is output for each point in the 3D point cloud.

To display intensity data, click the Intensity button ( ).

> To be able to display intensity data, you must enabled **Acquire Intensity** in the **Scan Mode** panel.

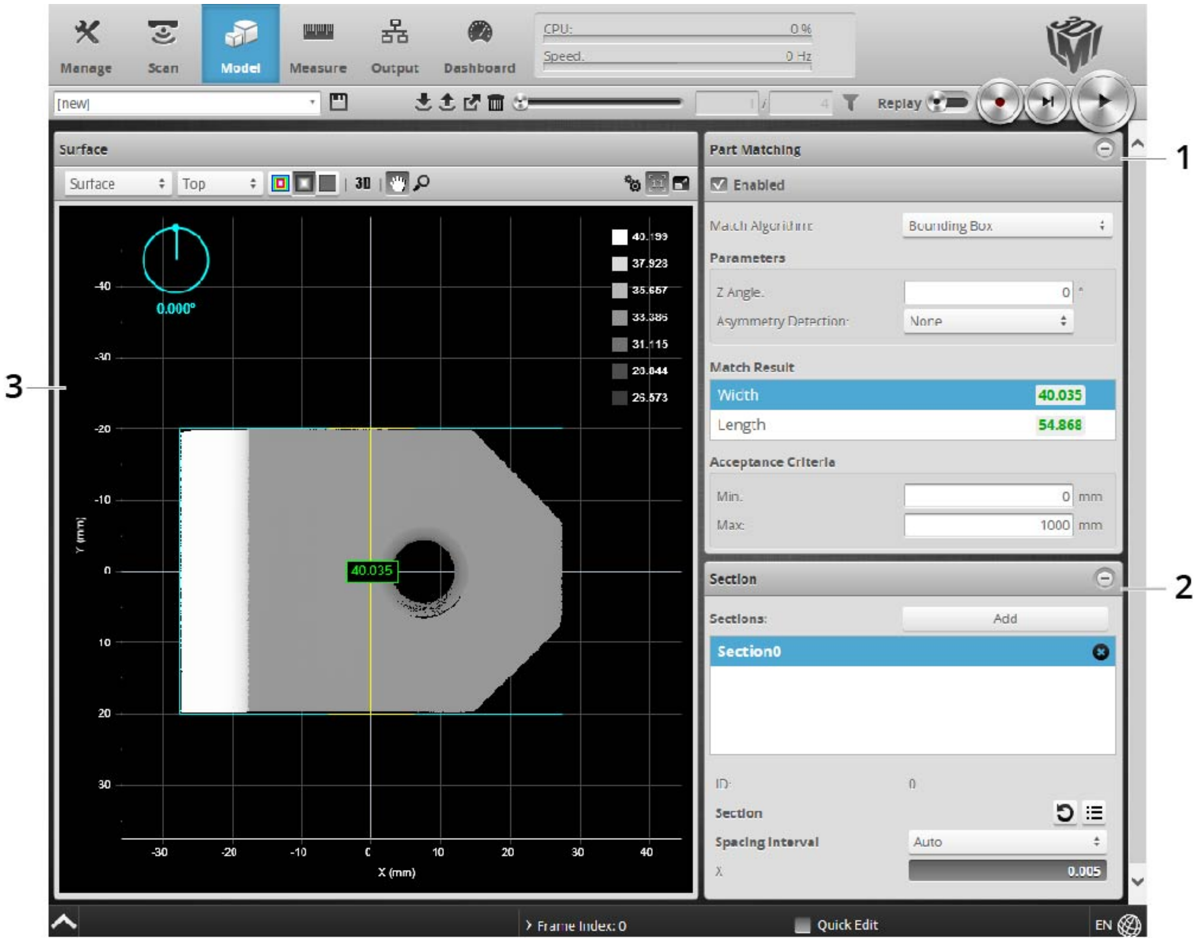


Models

The following sections describe how to set up part matching using a model, a bounding box, or an ellipse. It also describes how to configure sections.

Model Page Overview

The **Model** page lets you set up part matching and sections.



	Element	Description
1	Part Matching panel	Contains settings for configuring models and for part matching.
2	Sections panel	Contains settings for configuring sections, which let you extract profiles from surfaces.
3	Data Viewer	Displays sensor data and lets you add and remove model edge points.

Part Matching

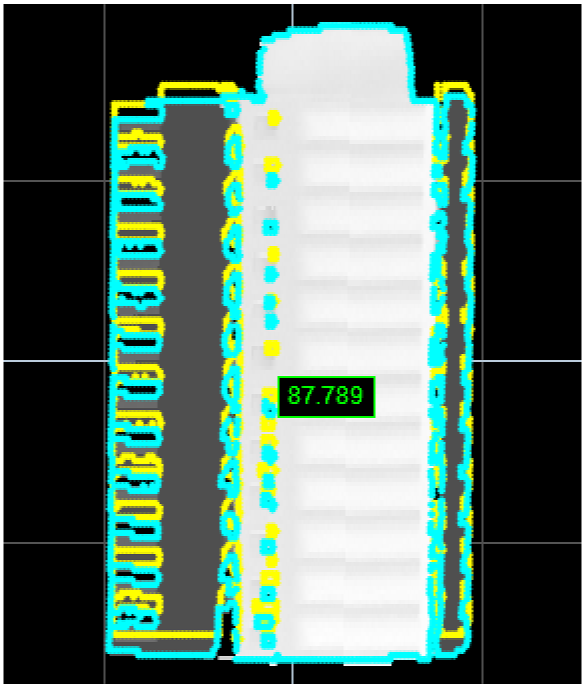
The sensor can match scanned parts to the edges of a model based on a previously scanned part (see *Using Edge Detection* on page 127) or to the dimensions of a fitted bounding box or ellipse that encapsulate the model (see *Using Bounding Box and Ellipse* on page 136). When parts match, the sensor can rotate scans so that they are all oriented in the same way. This allows measurement tools to be applied consistently to parts, regardless of the orientation of the part you are trying to match.

When the match quality between a model and a part reaches a minimum value (a percentage), or the bounding box or ellipse that encapsulates the part is between minimum and maximum dimension values, the part is "accepted" and any measurements that are added in the **Measure** page will return valid values, as long as the target is in range, etc. If the part is "rejected," any measurements added in the **Measure** page will return an Invalid value. For more information on measurements and decision values, *Measurement and Processing* on page 146.

Using Edge Detection

When using edge detection for part matching, the sensor compares a model that you must create from a previous scan to a "target" (one of the parts you want to match to the model).

In the data viewer, a model is represented as a yellow outline. The target is represented as a blue outline. If the part match quality above a minimum user-defined level, any measurements configured on the **Measure** page are applied.



Model (yellow outline) and target (blue outline).
Part match quality is 87.789%, which is greater
than the minimum set by the user, so the parts match.

When you create a model, the sensor runs an edge detection algorithm on either the heightmap or intensity image of a scanned part. The resulting model is made up of the detected edge points. The scan used to create the model should be of a reference (or "golden") part to which all other parts will be compared.

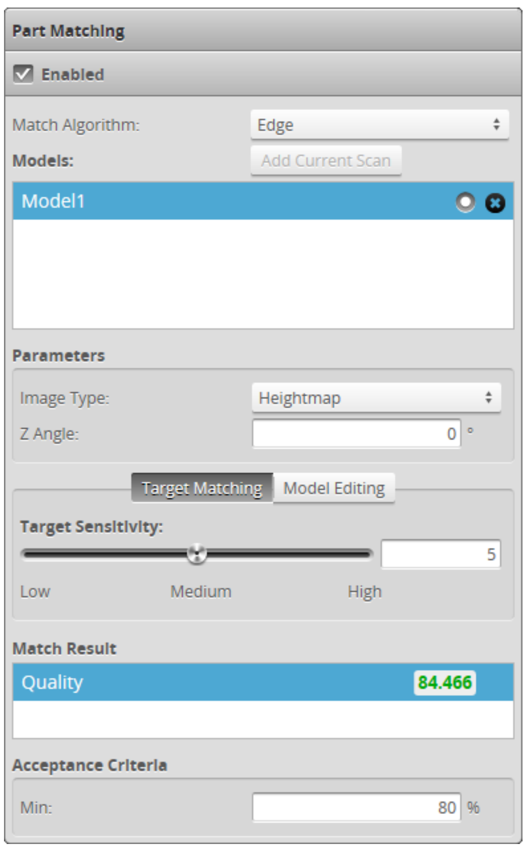
After the model has been created, you optionally modify the model by adjusting the sensitivity (how many edge points are detected), or selectively remove edge points from the model, to improve matching.

Models are saved as part of a job.

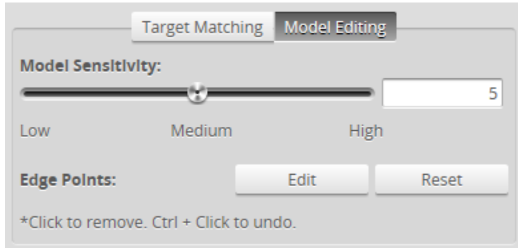
Once you have finished modifying the model, you can also modify target sensitivity, which controls how many edge points are detected on the subsequently scanned targets that will be compared to the model; the same edge detection algorithm used for creating models is used to compare a model to a part.

Typically, setting up edge detection to perform part matching involves the following steps:

1. Scan a reference part (you can also use replay data that you have previously saved).
2. Create a model based on the scan (using either heightmap or intensity data).
3. Adjust the model (edge detection algorithm sensitivity and selective removal of edge points).
4. Scan another part typical of the parts that would need to match the model.
5. Adjust the target sensitivity.
6. Set match acceptance level.



Part Matching panel showing Target Matching tab



Model Editing tab on Part Matching panel

The following settings are used to configure part matching using edge detection.

Setting	Description
Match Algorithm	Determines which algorithm the sensor will use to attempt a match. Set this to **Edge** for edge detection.
Image Type	Determines what kind of data the sensor will use to detect edges and therefore for part matching. Choose this setting based on the kinds of features that will be used for part matching: **Heightmap**: Surface elevation information of the scanned part will be used to determine edges. This setting is most commonly used. **Intensity**: Intensity data (how light or dark areas of a scanned part are) will be used to determine edges. Use this setting if the main distinguishing marks are printed text or patterns on the parts. The **Acquire Intensity** option must be checked in the **Scan Mode** panel on the **Scan** page for this option to be available.
Z Angle	Corrects the orientation of the model to accurately match typical orientation and simplify measurements.
Target Sensitivity (**Target Matching** tab)	Controls the threshold at which an edge point is detected on the target's heightmap or intensity image. (The "target" is any part that is matched to the model and which will subsequently be measured if the match is accepted.) Setting **Target Sensitivity** higher results in more edge points. Setting it lower results in fewer edge points and results in higher performance. Use this setting to exclude noise from the detected edges and to make sure distinguishing features are properly detected. The level of this setting should generally be similar to the level of **Model Sensitivity**.
Model Sensitivity (**Model Editing** tab)	Controls the threshold at which an edge point is detected on the heightmap or intensity image used to create the model. Setting **Model Sensitivity** higher results in more edge points. Setting it lower results in fewer edge points and results in higher performance. Use this setting to exclude noise from the detected edges and to make sure distinguishing features are properly detected. The level of this setting should generally be similar to the level of **Target Sensitivity**. Changing this setting causes the edge detection algorithm to run again at the

Setting	Description
	new threshold. If you have edited edge points manually (removing them selectively), those changes will be lost. See *Using Edge Detection* on page 127 for more information.
Edge Points (**Model Editing** tab)	The **Edit** button lets you selectively remove edge point that are detected by the edge detection algorithm at the current **Model Sensitivity** setting. See *Using Edge Detection* on page 127 for more information.
Acceptance Criteria	Determines the minimum quality level of the match as a percentage value.

Part rejected: Quality result is less than Min

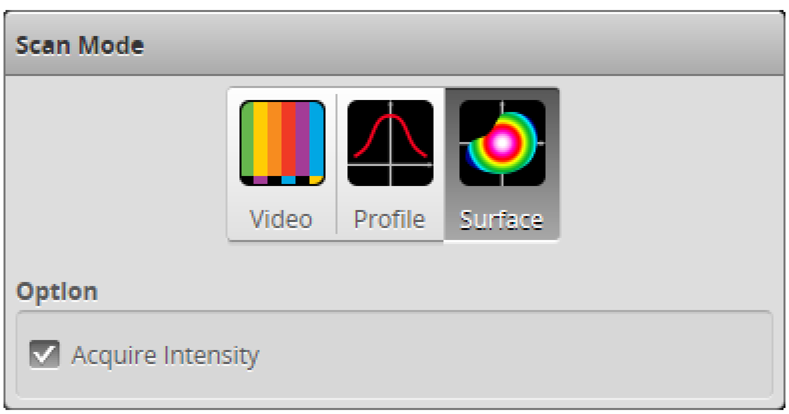
To run part matching, simply make sure that the **Enabled** option is checked on the **Part Matching** panel when the sensor is running. Any measurements that are added and configured on the **Measure** page will be applied to parts if a part match is accepted, regardless of the part's orientation (a successfully matched part is rotated to match orientation of the model), returning a value and decision (as long as the part is in range, etc.). If a part match is rejected, measurements will return an Invalid value.

Creating a Model

Gocator creates a model by running an edge detection algorithm on the heightmap or intensity image of a scan. The algorithm is run when a model is first created and whenever the **Model Sensitivity** setting is changed.

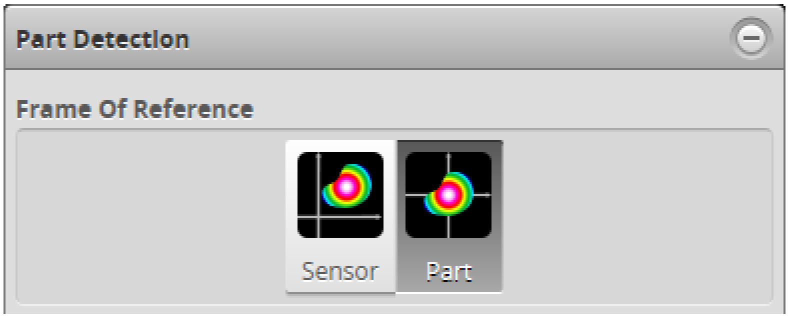
To create a model:

1. Go to the **Scan** page.
   a. In the **Scan Mode** panel, choose **Surface**.



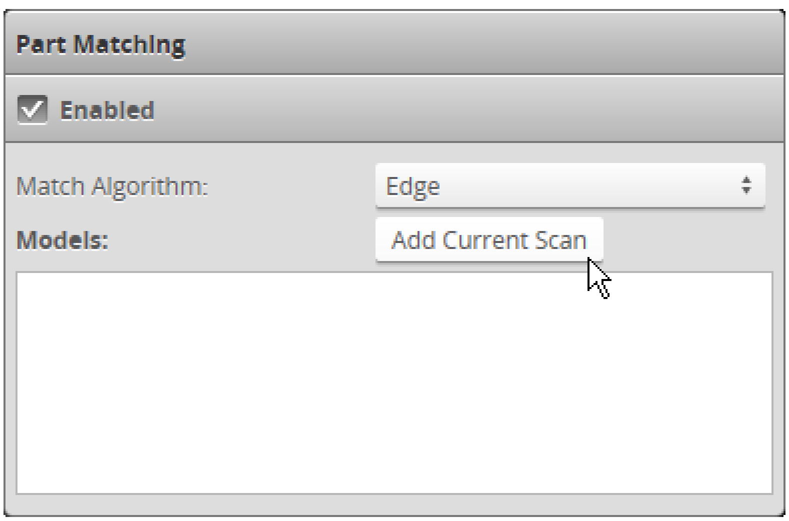
   You must choose **Surface** in order to scan a part. Furthermore, the **Model** page is only displayed in Surface mode.

   b. If you want to use intensity data to create the model, make sure **Acquire Intensity** is checked.
   c. In the **Part Detection** panel, choose **Part** for the **Frame of Reference**.



Part matching is only available when **Part** has been selected.

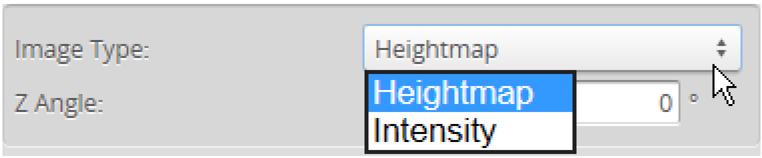
2. Do one of the following:
   - Scan a reference part. See *Scan Setup and Alignment* on page 87 for more information on setting up and aligning a sensor. See *Running a Standalone Sensor System* on page 37 for more information on running a system to scan a part.
   - Locate some previously recorded replay data and load it. See *Recording, Playback, and Measurement Simulation* on page 66 and *Downloading, Uploading, and Exporting Replay Data on page 69* for more information on replay data.
3. Go to the **Model** page.
   a. Make sure the **Enabled** option is checked in the **Part Matching** panel.
   b. In the **Match Algorithm** drop-down, choose **Edge**.



4. Click **Stop** on the toolbar if the sensor is running.
5. Click **Add Current Scan**.

> After adding the model, the sensor will show that the match quality is 100%, because it is in effect comparing the model to the scan that was used to create the model. This value can be ignored.

6. In the **Image Type** drop-down, choose **Heightmap** or **Intensity**.



7. If you need to correct the orientation of the model, provide a value in the **Z Angle** field.

   Correcting the Z angle is useful if the orientation of the model is not close to the typical angle of target parts on the production line.

8. Save the job by clicking the **Save** button.

   Models are saved in job files.

   See *Creating, Saving and Loading Jobs (Settings)* on page 64 for more information on saving jobs.

After you have created a model, you may wish to modify it to remove noise to improve its matching capabilities. You may also wish to modify a model to exclude certain areas. See *Creating a Model* on page 130 for more information.

Model names can be renamed.

To rename a model:

1. In the **Models** list, double-click on a model name.
2. Type a new name in the model name field.
3. Press Enter or click outside the model name field.
4. Save the job by clicking the **Save** button.

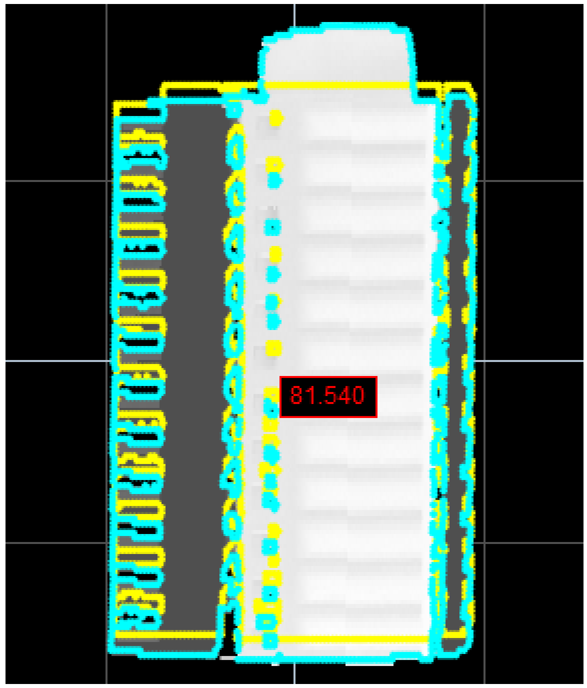
To delete a model, click the button.

Modifying a Model's Edge Points

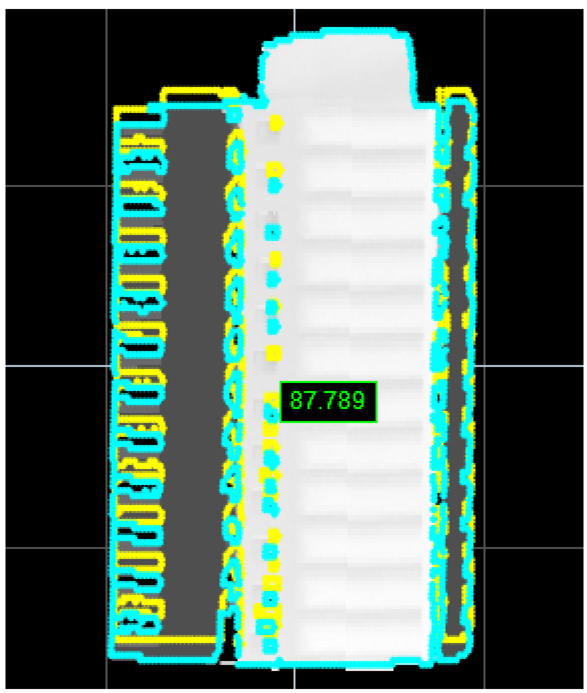
Modifying a model's edge points is useful to exclude noise in the detected edge points and to make sure distinguishing features are properly detected, which can improve matching. You can modify edge points in two ways.

First, you can control the overall number of edge points that are detected by the edge detection algorithm by raising and lowering the edge detection threshold (the **Model Sensitivity** setting). Modifying **Model Sensitivity** causes the edge detection algorithm to run again.

Second, you can fine-tune the model's edge points by selectively removing edge points that are detected by the edge detection algorithm. This could be useful, for example, if an edge on the target parts frequently presents minor variations such as flashing (excess material caused by leakage during molding): the edge points that make up the model can be edited to exclude that region. Editing the model can allow parts to match it more easily.



Edge points along top of model not removed.
Part is rejected. (Min set to 85%.)



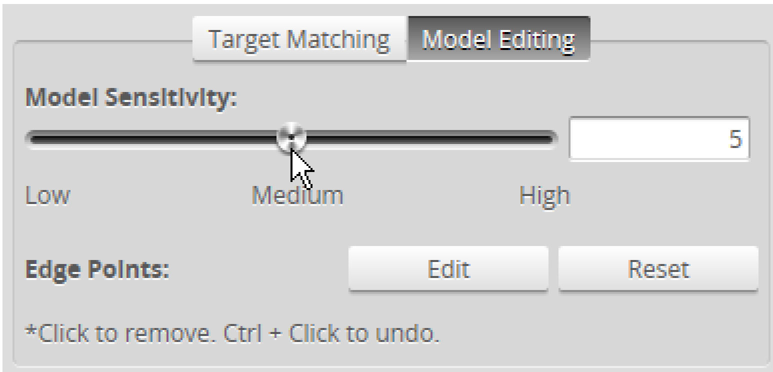
Edge points along top of model removed.
Part is accepted. (Min set to 85%.)

Removing edge points does not cause the edge detection algorithm to run again.

To change model senstivity:

1. In the **Models** list, select the model you want to configure by clicking on its selection control.

2. Click the **Model Editing** tab.
3. Adjust the **Model Sensitivity** slider to exclude noise and to properly detect the distinguishing features that will match parts.

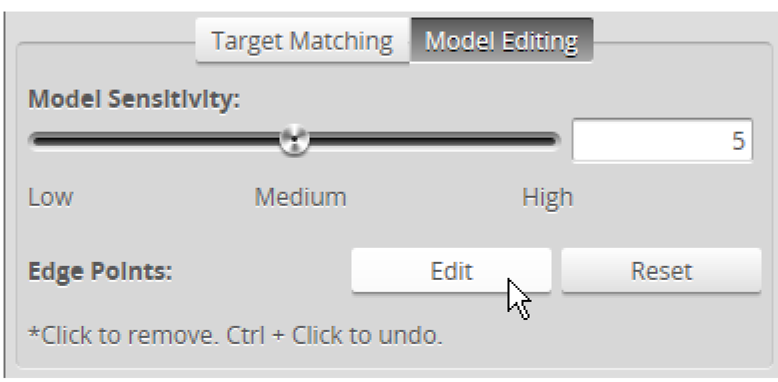


You can also set the sensitivity value manually in the provided text box.
4. Save the job by clicking the **Save** button.

To manually remove model edge points:

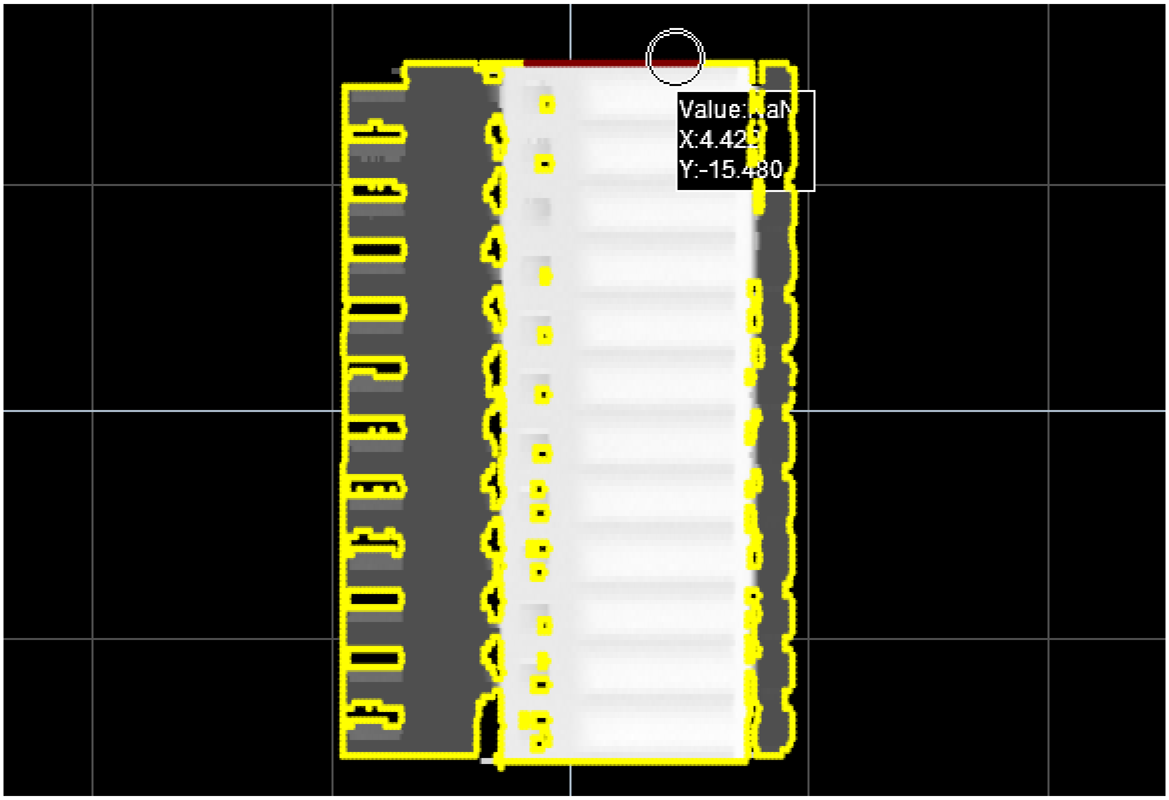
1. In the **Models** list, select the model you want to configure by clicking on its selection control.

2. In the **Model Editing** tab, click on the **Edit** button.



3. On the toolbar above the data viewer, make sure the **Select** tool is active.

4. Click in the data viewer and hold the mouse button while moving the pointer over the edge points you want to remove.



   Points within the circular **Select** tool are removed from the model. Removed edge points turn red in the data viewer.

   You can zoom in to see individual edge points by using the mouse wheel or by using the Zoom mode ( ).

5. If you have removed too many edge points, use Ctrl + Click in the data viewer to add the edge points back.
6. When you have finished editing the model, click **Save** in the **Model Editing** tab.
7. Save the job by clicking the **Save** button on the toolbar.

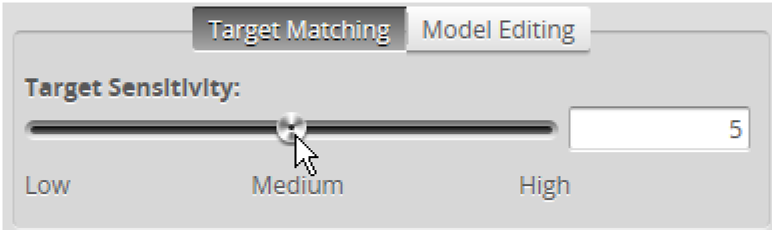
Adjusting Target Sensitivity

After you have added a model and optionally adjusted it, you must scan a different part, one that is typical of parts that must match the model.

Much in the same way that you can adjust a model's sensitivity, you can adjust the target sensitivity, that is, the threshold at which edge points are detected on the heightmaps or intensity images of parts that you want to match to the model. Adjusting the target sensitivity is useful to exclude noise, improving part matching.

To change target senstivity:

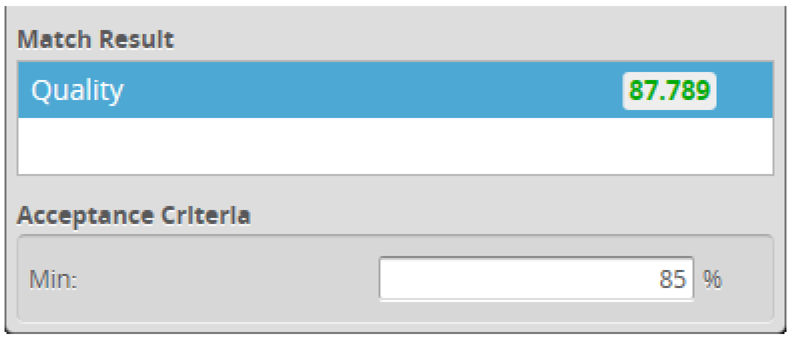
1. Click the **Target Matching** tab.
2. Adjust the **Target Sensitivity** setting to exclude noise in order to properly detect the distinguishing features that will allow parts to match.



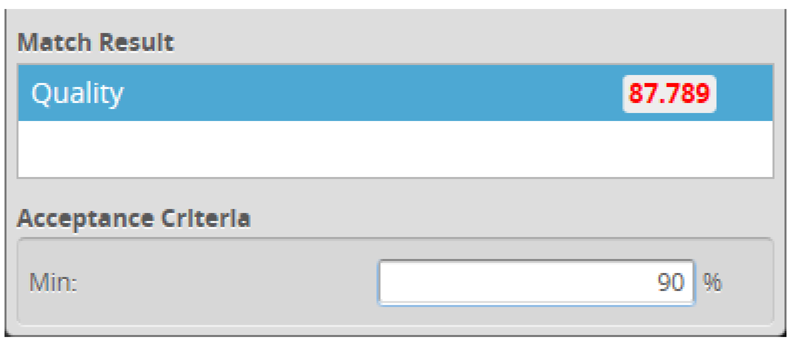
You can also set the sensitivity value manually in the provided text box.

Setting the Match Acceptance Criteria

In order for a part to match a model, the match quality must reach the minimum set in the **Min** field in **Acceptance Criteria** section of the **Part Matching** panel.



Part accepted: Quality result is greater than Min



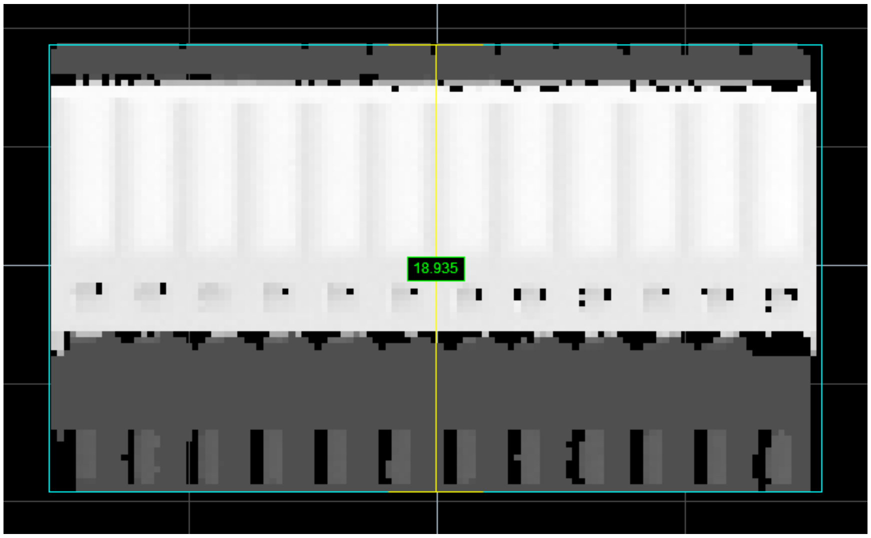
Running Part Matching

To run part matching, simply make sure that the **Enabled** option is checked on the **Part Matching** panel when the sensor is running. Any measurements that are added and configured on the **Measure** page will be applied to parts if a part match is accepted, regardless of the part's orientation (a successfully matched part is rotated to match orientation of the model), returning a value and decision (as long as the part is in range, etc.). If a part match is rejected, measurements will return an Invalid value.

Using Bounding Box and Ellipse

When using a bounding box or an ellipse to match parts, the sensor tests whether a part fits into a bounding box or ellipse that you define. A match will occur regardless of orientation.

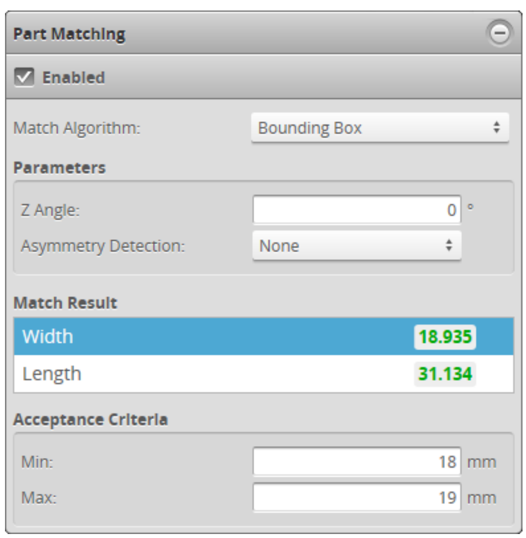
In the data viewer, a bounding box or ellipse is displayed with a blue outline. If a part fits in the bounding box or ellipse, any measurements configured on the **Measure** page are applied.



Blue bounding box around a part.
(Yellow lines show currently selected dimension in Part Matching panel.)

Typically, setting up a bounding box or an ellipse to perform part matching involves the following steps:

1. Scan a reference part (you can also use replay data that you have previously saved).
2. Set the characteristics of the bounding box (width and length) or ellipse (major and minor axes).



Part Matching panel (Bounding Box match algorithm)

The following settings are used to configure part matching using a bounding box or ellipse.

Setting	Description
Match Algorithm	Determines which algorithm the sensor will use to attempt a match. Set this to

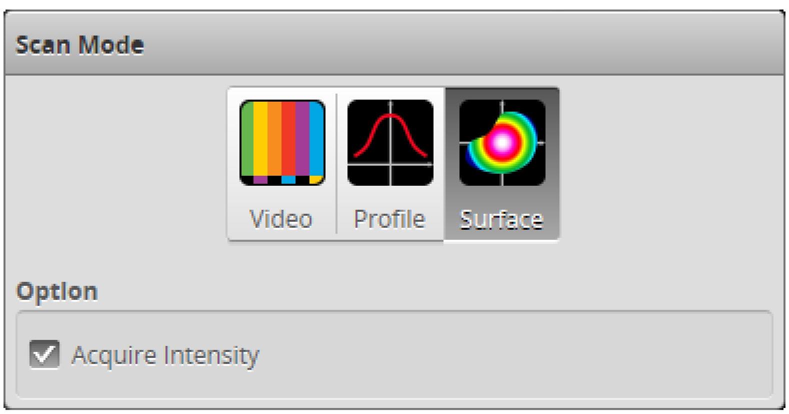
Setting	Description
	Bounding Box or **Ellipse**.
Z Angle	Corrects the orientation of the bounding box or ellipse to accurately match typical orientation and simplify measurements.
Asymmetry Detection	Rotates scans based on the asymmetry of the scanned part. The sensor calculates the number of points on each side of the part's centroid in the bounding box or ellipse. **Along Major Axis** – The scan is flipped so that the greater number of points is to the left. **Along Minor Axis** – The scan is flipped so that the greater number of points is on the bottom. **None** – The scan is not flipped.
Acceptance Criteria	Determines the minimum and maximum acceptable values of the selected dimension (Width and Length for bounding box, Major and Minor for ellipse) in **Match Result**.

Configuring a Bounding Box or an Ellipse

To use a bounding box or an ellipse to match a part, you must set its dimensions, taking into account expected acceptable variations when compared to a reference (or "golden") part.

To configure a bounding box or ellipse for part matching:

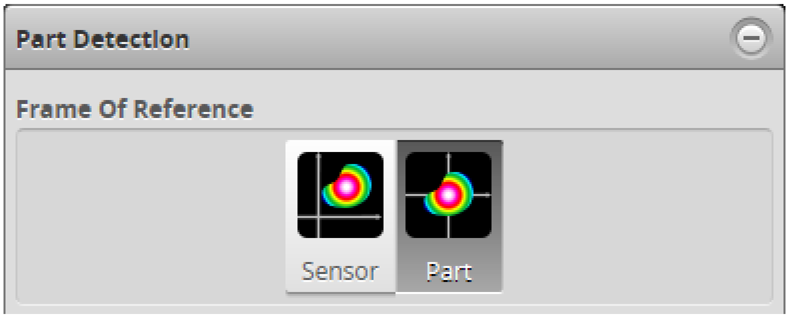
1. Go to the **Scan** page.
   a. In the **Scan Mode** panel, choose **Surface**.



   You must choose **Surface** in order to scan a part. Furthermore, the **Model** page is only displayed in Surface mode.

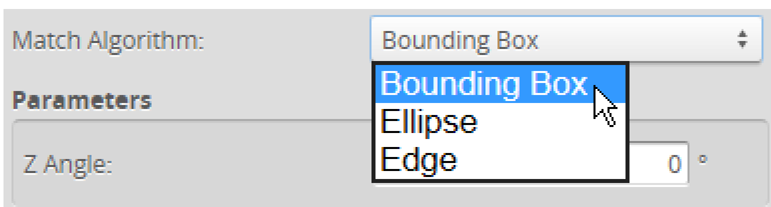
   Intensity data is not used when part matching using a bounding box or an ellipse, but you can enable the **Acquire Intensity** option if you need intensity data for other reasons.

   b. In the **Part Detection** panel, choose **Part** for the **Frame of Reference**.



Part matching is only available when **Part** has been selected.

2. Do one of the following:
   - Scan a reference part. See *Scan Setup and Alignment* on page 87 for more information on setting up and aligning a sensor. See *Running a Standalone Sensor System* on page 37 for more information on running a system to scan a part.
   - Locate some previously recorded replay data and load it. See *Recording, Playback, and Measurement Simulation* on page 66 and *Downloading, Uploading, and Exporting Replay Data on page 69* for more information on replay data.
3. Go to the **Model** page.
   a. Make sure the **Enabled** option is checked in the **Part Matching** panel.
   b. In the **Match Algorithm** drop-down, choose **Bounding Box** or **Ellipse**.



4. Set **Min** and **Max** of both of the dimensions of the selected match algorithm shape, taking into account expected acceptable variations.
   - If you chose **Bounding Box** for the match algorithm, select **Width** and then **Length** in **Match Result**, setting the minimum and maximum values acceptable for each dimension.
   - If you chose **Ellipse** for the match algorithm, select **Minor** and then **Major** in **Match Result**, setting the minimum and maximum values acceptable for each dimension.
5. Save the job by clicking the **Save** button.

   See *Creating, Saving and Loading Jobs (Settings)* on page 64 for more information on saving jobs.

Running Part Matching

To run part matching, simply make sure that the **Enabled** option is checked on the **Part Matching** panel when the sensor is running. Any measurements that are added and configured on the **Measure** page will be applied to parts if a part match is accepted, regardless of the part's orientation (a successfully matched part is rotated to match orientation of the bounding box or ellipse), returning a value and decision (as long as the part is in range, etc.). If a part match is rejected, measurements will return an Invalid value.

Using Part Matching to Accept or Reject a Part

Part matching results only determine whether a measurement is applied to a part. Whether the measurement returns a pass or fail value—its decision—depends on whether the measurement's value is between the **Min** and **Max** values set for the measurement. This decision, in addition to the actual value, can in turn be used to control a PLC for example. The part matching "decision" itself is not passed to the Gocator output, but you can simulate this by setting up a measurement that will always pass if it is applied.

For example, you could set up a Position Z measurement, choosing Max Z as the feature type, and setting the **Min** and **Max** values to the measurement range of the sensor. This way, as long as a part matches and the target is in range, etc., the measurement will pass. This measurement decision, which is passed to the Gocator's output, could in turn be used to control a PLC.

Sections

In Surface mode, the sensor can also extract a profile from a surface or part using a line you define on that surface or part. The resulting profile is called a “section.” A section can have any orientation on the surface, but its profile is parallel to the Z axis.

	Sections are not available on point cloud data, that is, when **Uniform Spacing** is disabled in the **Scan Mode** panel.

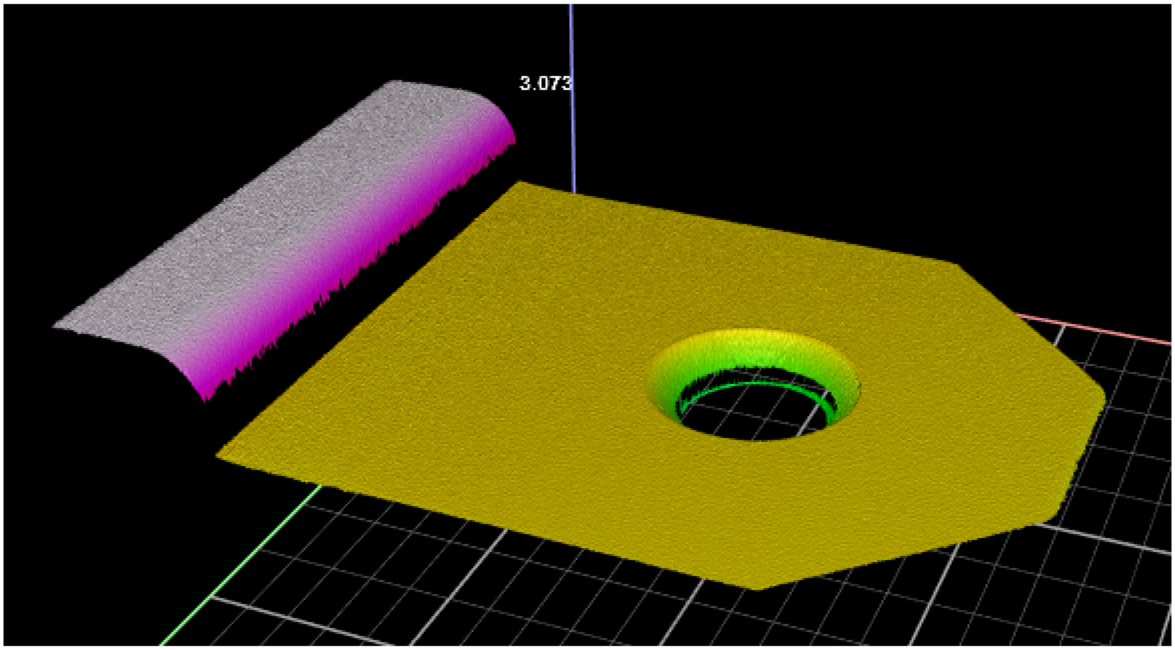
	You can't create sections from the **Models** page on surface data that is produced by other tools, such as Surface Stitch. You can however create sections on any kind of surface data using the Surface Section tool; for more information, see *Section* on page 415.

You can use most profile measurement tools on a section: you can't use tools that work with unresampled data. Using sections and the profile measurements, you can therefore use measurements that are not otherwise possible in Surface mode, for example:

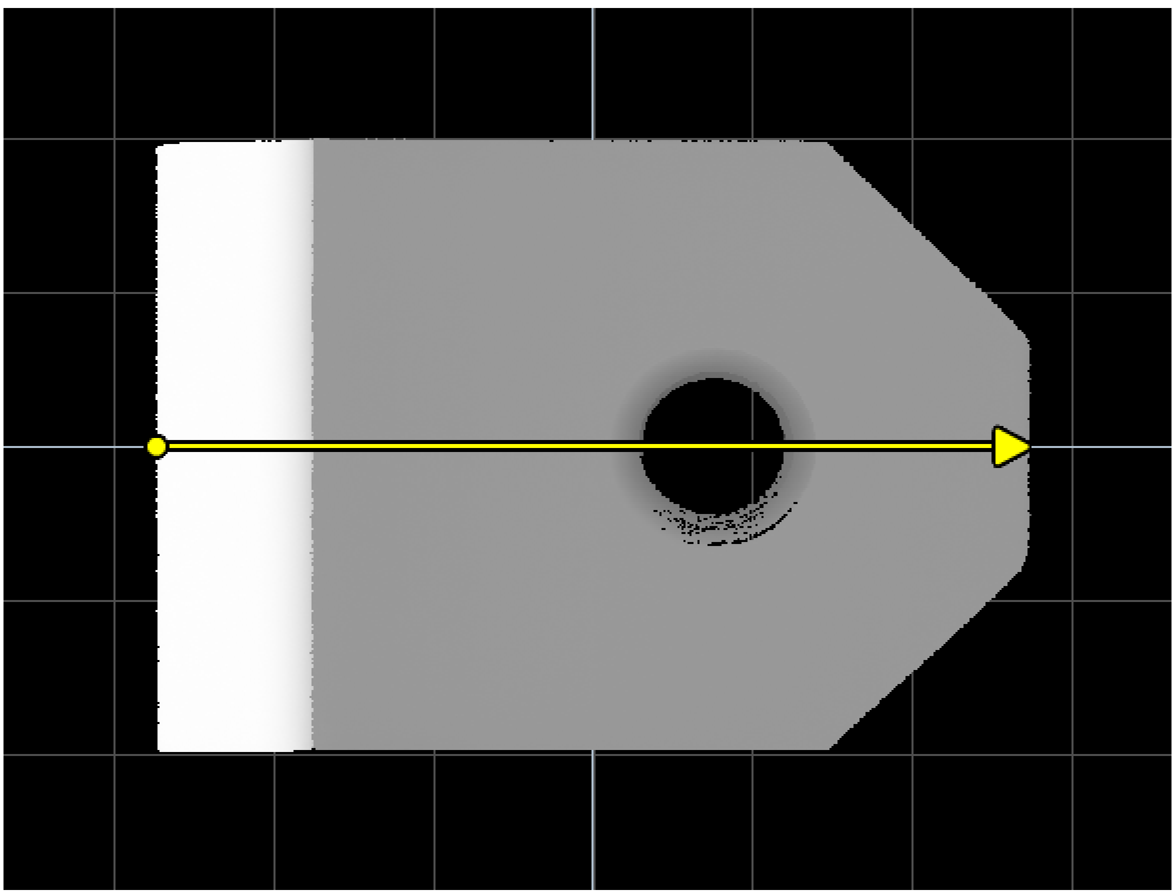
- Gap and flush measurements
- Surface radius measurements (for example, rounded edges or corners)
- Intersections
- Point-to-point dimension measurements between profile features

Gocator supports multiple sections, letting you take multiple measurements on the same object.

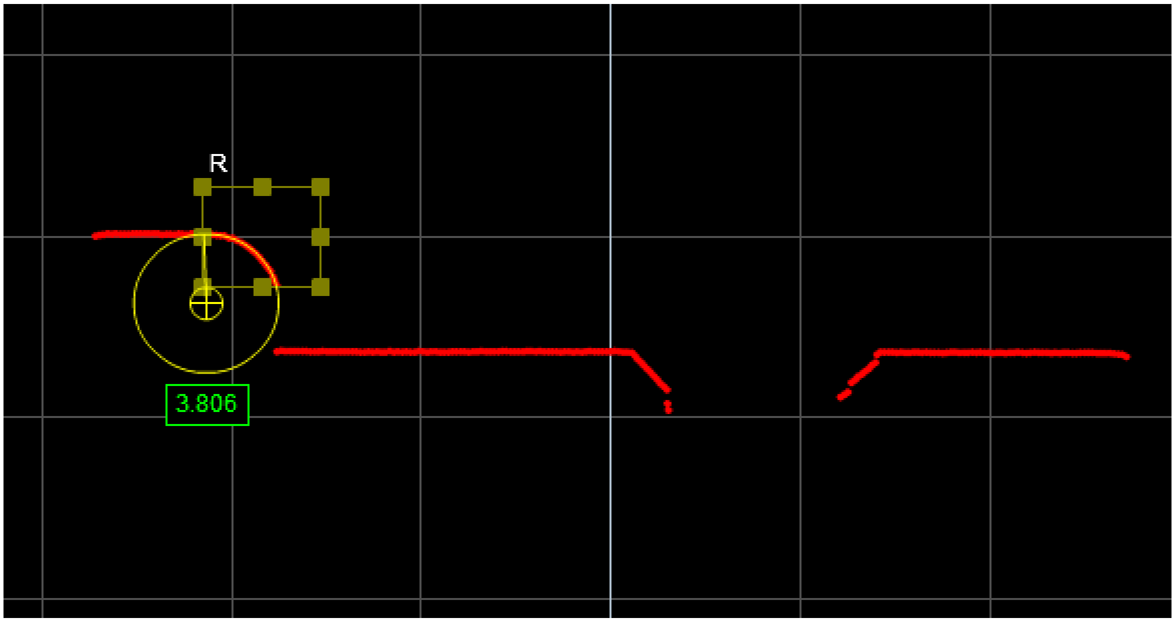
On the **Output** page, in Surface mode, you can output both surface measurements and section-based profile measurements at the same time. The sensor can also output the surfaces and section profiles themselves at the same time.



Part in data viewer (3D view)

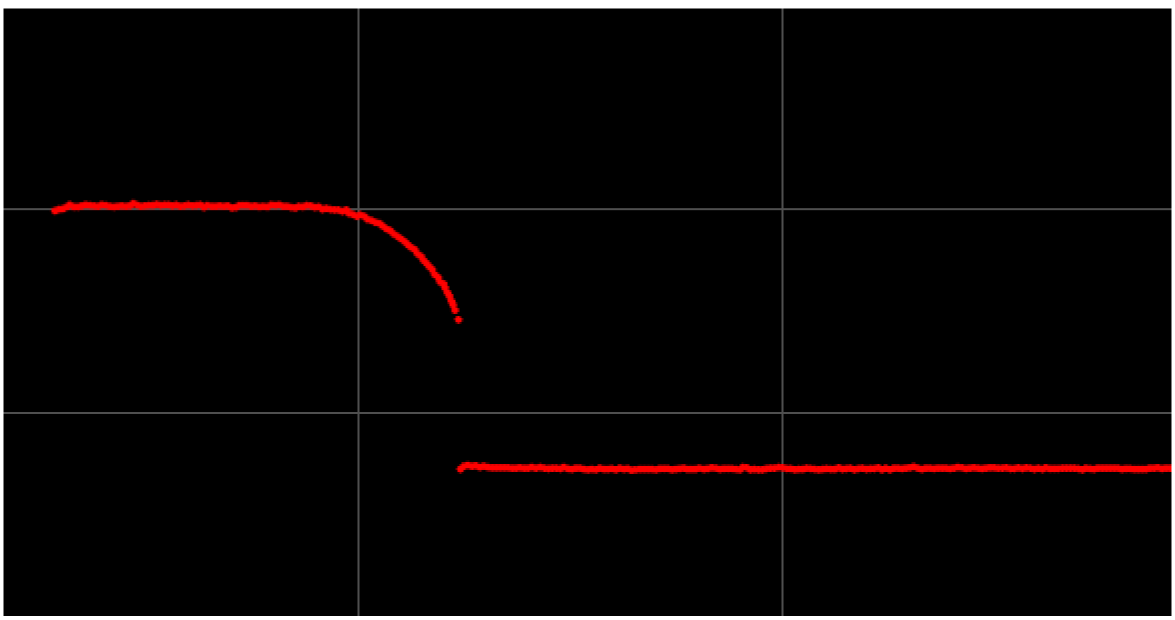


Section defined on top of part (2D view)

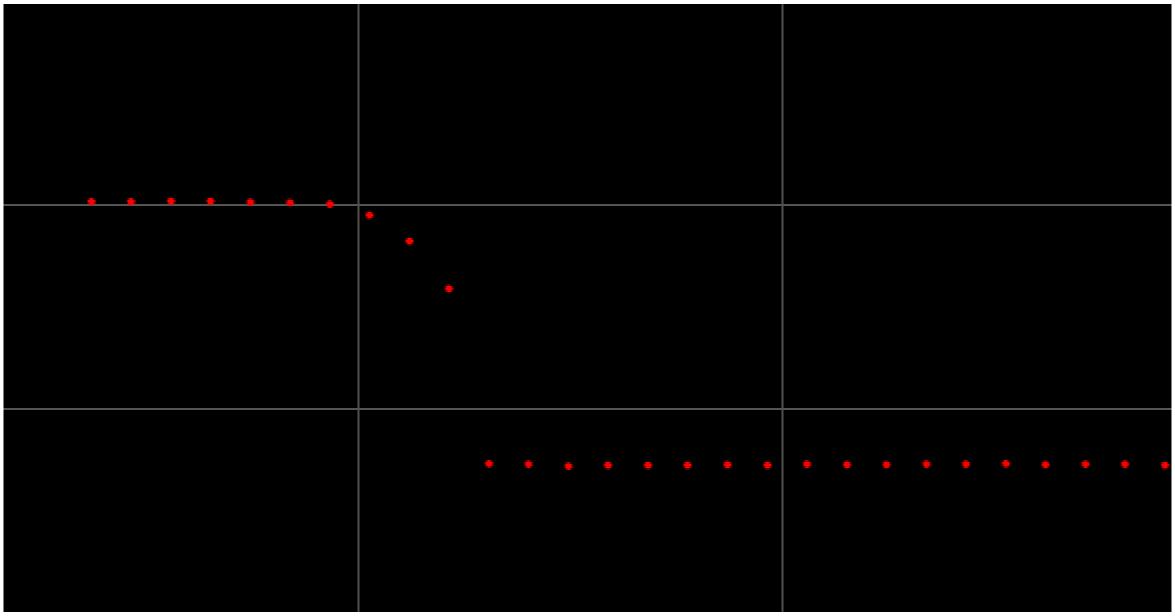


Circle Radius measurement running on profile extracted from surface using defined section

You can configure the sampling distance between points along the section. Reducing the sampling distance reduces the resolution of the profile, but increases the sensor's performance and results in less data being sent over the output.



Mininum spacing interval: highest profile resolution, greater sensor CPU usage and data output



Maximum spacing interval: lowest profile resolution,
lower sensor CPU usage and data output

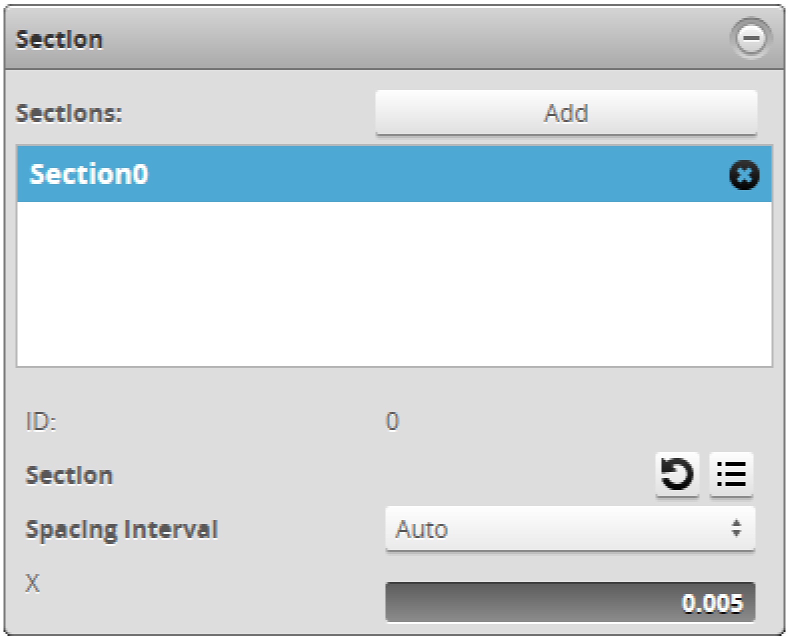
	Using a higher spacing interval can produce different measurement results compared to using a smaller spacing interval. You should therefore compare results using different spacing intervals before using sections in production.

The sections you add to a surface are directional, and their start and end points are defined using X and Y coordinates. The start point always corresponds to the leftmost point on the extracted profile, whereas the end point always corresponds to the rightmost point on the extracted profile, no matter the orientation of the section on the surface.

For more information on profile tools, see *Profile Measurement* on page 206.

Creating a Section

Before you create a section, you should first scan a target in Surface mode to create a surface on which you can create the section. You can use either live data or recorded data.



After creating a section, the following settings are available:

Setting	Description
Spacing Interval	Determines the space between the points of the extracted profile. **Auto**: The highest resolution, calculated using the X and Y resolution of the scan. **Custom**: Lets you set the spacing interval by using a slider or setting the value manually.
Section	Lets you manually set the X and Y coordinates of the start and end points of the section. Setting the coordinates manually is useful if you need to create a section that is perfectly horizontal or vertical. For example, to create a horizontal section, copy the Y value of either the start or end point to the other point's Y field. You can reverse the start and end points by clicking the button. To reset the start and end points to their initial values, click the button.

To create a section:

1. On the **Scan** page, in the **Scan Mode** panel, click **Surface**.
2. On the **Model** page, in the **Section** panel, click **Add**.

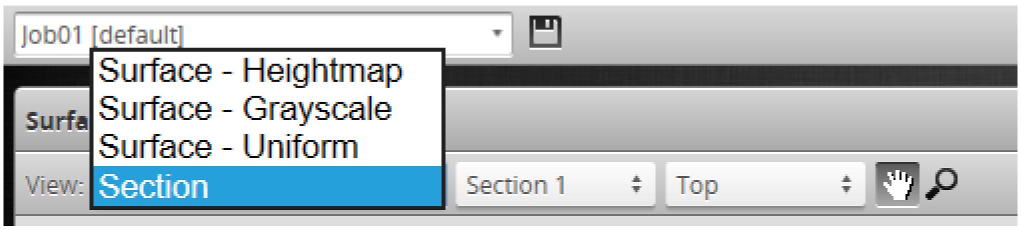
   You may need to click the button to expand the panel.

   The sensor creates a section on the surface.
3. Rename the section if you want.
4. Move the section and adjust the start and end points of the section to extract the desired profile.

   You can move or adjust the section graphically in the data viewer, or you can manually adjust the X and Y coordinates of the section.
5. (Optional) Adjust the **Spacing Interval**.

After you create a section, the profile measurement tools become available in the **Tools** panel on the **Measure** page. If you have created more than one section, you must select it in the tool. For more information on profile measurement tools, see *Profile Measurement* on page 206.

The sensor also adds a **Section** option to the **View** drop-down above the data viewer, which lets you view an extracted profile, as well as a section selector drop-down for cases where multiple sections are defined.



Sections are also added to the **Stream** drop-down in Profile and Feature tools.

If parts are not consistently oriented in the same way from scan to scan, you can use part matching to correct their rotation, if the entire part is visible in the scan. Parts will then be consistently oriented, and sections will fall on the same area on each part. You can also use anchoring to ensure that measurements are consistently placed on a part.

Deleting a Section

When you delete a section, the sensor removes any associated measurements. After you remove the last section, the sensor no longer displays profile measurement tools in the **Tools** panel.

To delete a section:

1. On the **Scan** page, in the **Scan Mode** panel, click **Surface**.
2. On the **Model** page, in the **Section** panel, click the button of the section you want to delete.

   You may need to click the button to expand the panel.

   If you have associated a measurement tool to the section by setting the tool's **Stream** setting to the section, the sensor asks if you want to delete all of the associated measurement tools.

   The sensor deletes the section on the surface.

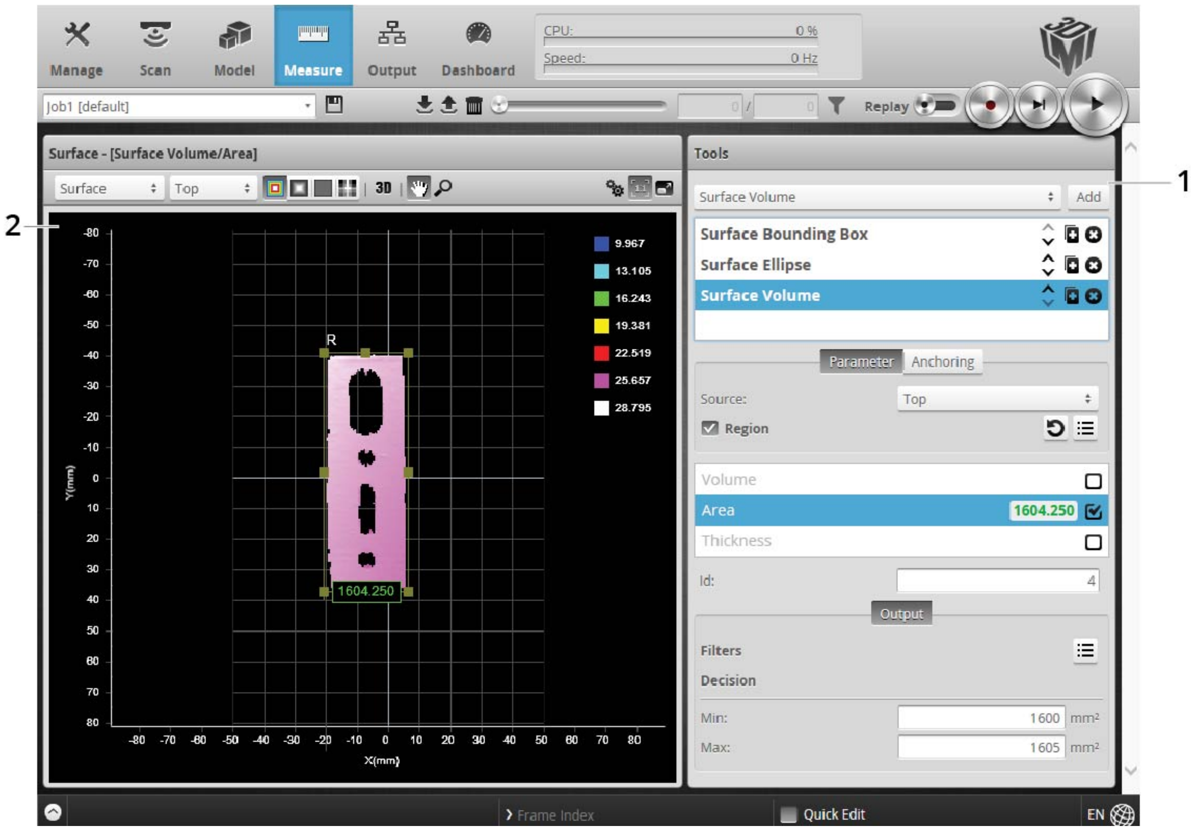
Measurement and Processing

The following sections describe Gocator's measurement and processing tools.

Measure Page Overview

Measurement tools are added and configured in the **Measure** page.

The content of the **Tools** panel in the **Measure** page depends on the current scan mode. In Surface mode, the **Measure** page displays tools for surface measurement. If you have defined a section in Surface mode, profile tools are also displayed. In Video mode, tools are not available.



	Element	Description
1	Tool configuration panel	Used to add, manage, and configure tools and measurements (see *Tools Panel* on page 149) and to choose anchors (*Measurement Anchoring* on page 169).
2	Data Viewer	Displays video and scan data, sets up tools, and displays result calipers related to the selected measurement. Parts are displayed using a height map, which is a top-down view of the XY plane, where color represents height. See *Data Viewer* on the next page.
3	Tools Diagram	Provides a visual representation of tools and the flow of data between them. For more information, see *Working with the Tools Diagram* on page 177.

	Element	Description
4	Feature Area	Configurable region of interest from which feature points are detected. These feature points are used to calculate the measurements. The number of feature areas displayed depends on which measurement tool is currently selected.
5	Displayed Outputs	Lists the measurements and geometric features currently displayed or pinned in the data viewer. For more information, see *Pinning Measurements and Features* on page 198.

Data Viewer

When the **Measure** page is active, the data viewer can be used to graphically configure measurement regions. Measurement regions can also be configured manually in measurements by entering values into the provided fields (see *Regions* on page 152).

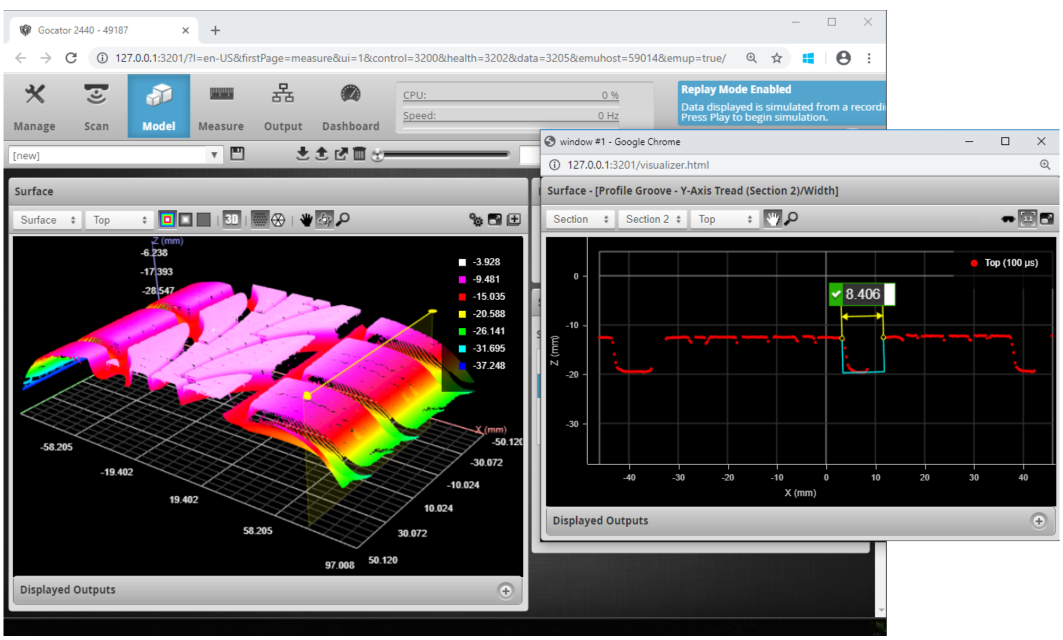
For information on controls in the data viewer, see *Data Viewer Controls* on page 115.

For information on setting up measurement regions graphically, see *Region Definition* on page 124.

For information on opening and using additional data viewer windows, see *Using Multiple Data Viewer Windows* below.

Using Multiple Data Viewer Windows

You can open multiple windows outside of the main browser window containing data viewers set to different views and different sets of pinned outputs. This lets you more easily monitor or set up complex applications, for example placing one or more data viewer window in one computer monitor, and others in a different monitor.

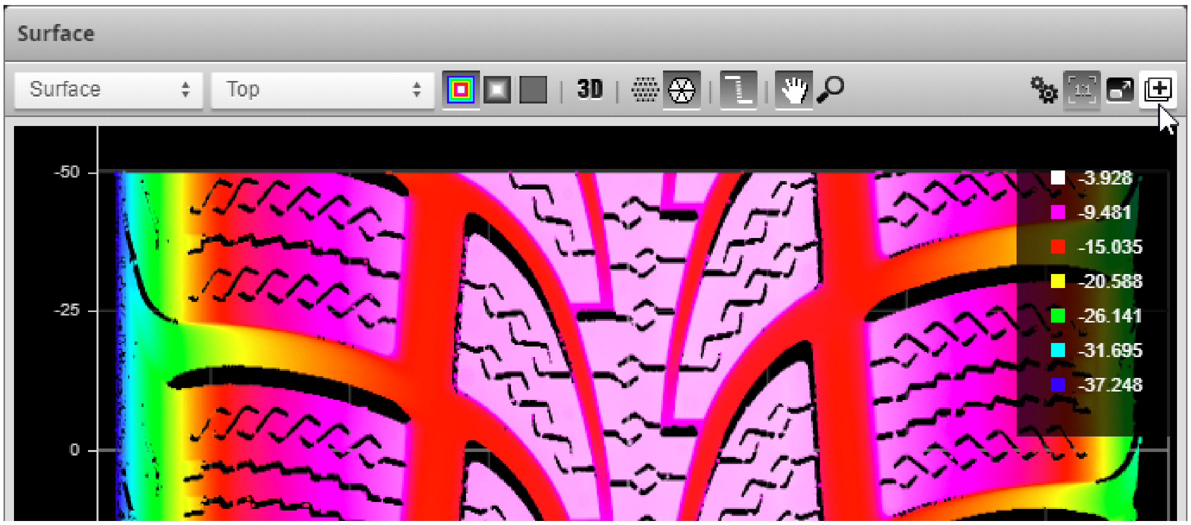


Main view in original browser window showing surface data and a defined section, and a second window showing a Profile tool running on the section

External data viewer windows provide the same functions as the Main View data viewer via the toolbar above the viewer (except for the ability to open a new window). External windows also include a Displayed Outputs panel at the bottom and support the pinning of outputs; pinning in external windows is independent from the Main View data viewer and other external windows. For more information on pinning outputs, see *Pinning Measurements and Features* on page 198.

To open a new external data viewer window:

1. In the toolbar of the Main View data viewer, click the New Data Viewer button ().



A new window opens containing a separate data viewer.

Use the tool bar at the top of the new data viewer to choose and modify the view (Surface vs. section data, color heightmap vs. intensity, 2D vs. 3D, etc.). For more information, see *Data Viewer* on page 115.

Pin outputs to the new data viewer as in the Main View data viewer. For more information, see *Pinning Measurements and Features* on page 198. Any outputs pinned in the Main View when you open a new data viewer window appear already pinned in the new window, but pinning in data viewers is otherwise independent.

Tools Panel

The **Tools** panel lets you add, configure, and manage measurement tools. Tools contain related measurements. For example, the Dimension tool provides Height, Width, and other measurements.

You can also add and remove tools, and connect tool and sensor ouputs to tool inputs from within the Tools Diagram panel. The Tools Diagram panel helps make working with complex applications much more easy, but you configure a tool's main parameters from within the Tools panel. For more information on the Tools Diagram panel, see *Working with the Tools Diagram* on page 177.

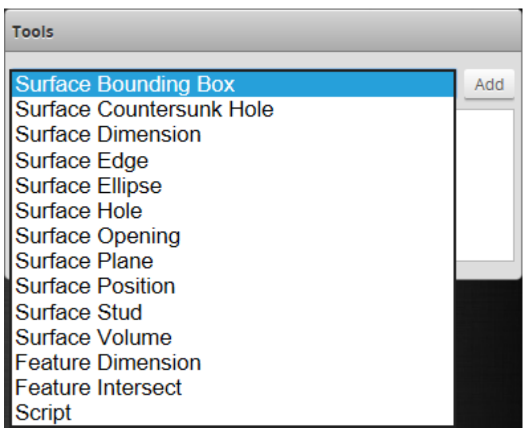
Some settings apply to tools, and therefore to all measurements; these settings are found in the **Parameters** tab below the list of tools. Other settings apply to specific measurements, and are found in a **Parameters** tab below the list of measurements; not all measurements have parameters.

See *Surface Measurement* on page 277 for information on the measurement tools and their settings.

Tool names in the user interface include the scan mode, but not in the manual. So for example, you will see "Surface Bounding Box" in the user interface, but simply "Bounding Box" in the manual.

Adding and Configuring a Measurement Tool

Adding a tool adds all of the tool's measurements to the **Tools** panel. You can then enable and configure the measurements selectively.



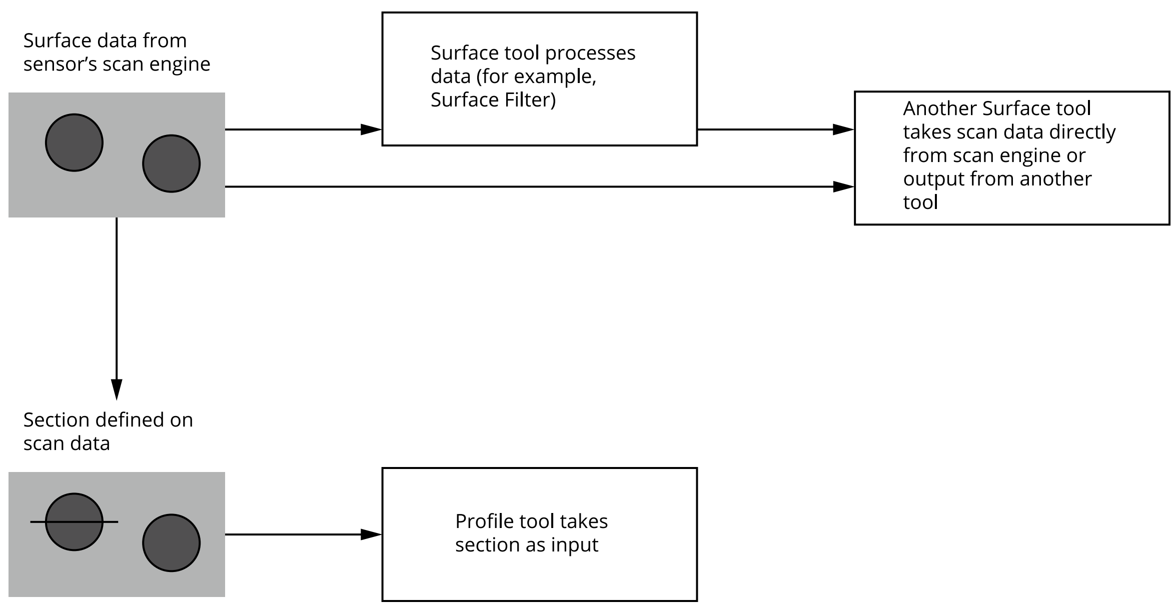
To add and configure a tool:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Profile or Surface mode in the **Scan Mode** panel.
   If one of these modes is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the Tools panel, select the tool you want to add from the drop-down list of tools.
5. Click on the **Add** button in the Tools panel.
   The tool and its available measurements are added to the tool list. The tool parameters are listed in the area below the tool list.

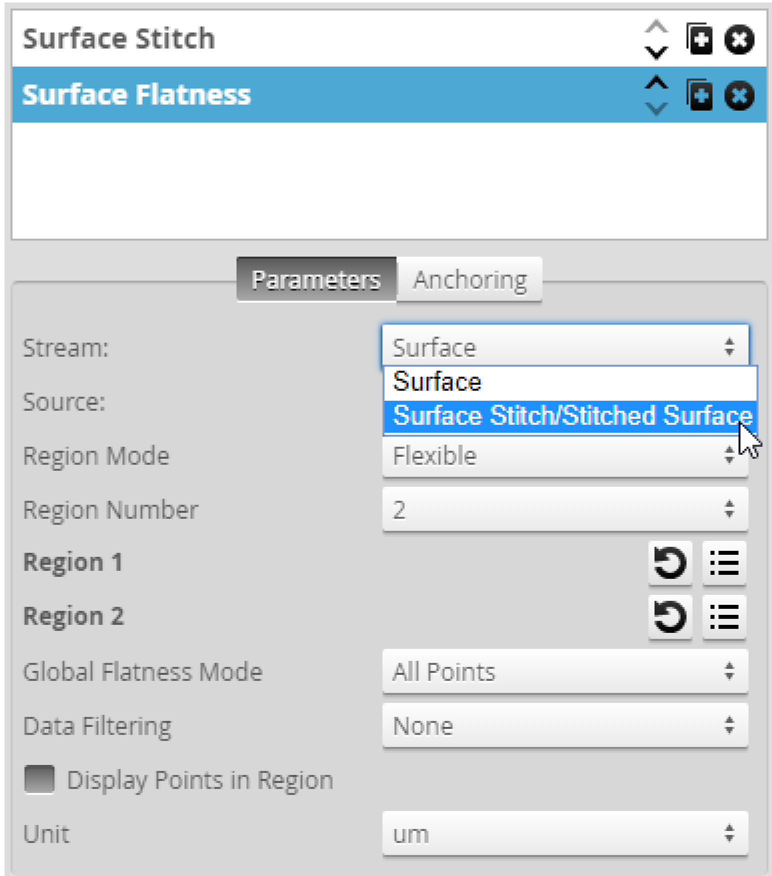
6. (Optional) If the measurement is a profile measurement running on a section, and you have created more than one section, choose the section that will provide data to the measurement in **Stream**.
   For more information on streams, see *Stream* below.
7. Select a measurement at the bottom of the tool panel.
8. Set any tool- or measurement-specific settings.
   For tool- and measurement-specific settings, see the topics for the individual profile or surface tools.
9. Set the **Min** and **Max** decision values.
   For more information on decisions, see *Decisions* on page 166.
10. (Optional) Set one or more filters.
    For more information on filters, see *Filters* on page 167.
11. (Optional) Set up anchoring.
    For more information on anchoring, see *Measurement Anchoring* on page 169.

Stream

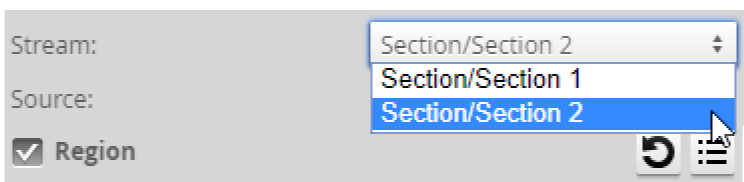
It's possible for more than one type of data to be available for a tool as input. You use the **Stream** drop-down in a tool to choose which type. If only one type of data is available for a tool, the **Stream** drop-down may not be displayed.



For example, many tools can produce processed surface data (such as the Stitched Surface output from the Surface Stitch tool, or the Corrected Surface output from the Surface Vibration Correction tool). When you have added one of these tools, the tool's data output is listed in the **Stream** drop-down, as well as the data that comes directly from the sensor's scanning engine. Surface data coming directly from the sensor's scan engine is always called "Surface" in the **Stream** drop-down. For data that comes from another tool, the convention is `{Tool name}/{Data output name}`:



[Sections](#) are also listed in the **Stream** setting.



To choose a stream:

1. Go to the **Measure** page by clicking on the **Measure** icon.

   > The [scan mode](#) must be set to the type of measurement you need to configure. Otherwise, the wrong tools, or no tools, will be listed on the **Measure** page.

2. In the **Tools** panel, click on a tool in the tool list.
3. If it is not already selected, click the **Parameter** tab in the tool configuration area.
4. Select the data in the **Stream** drop-down list.

Source

This setting is always **Top** with G3 sensors.

Regions

Many measurement tools use user-defined regions to limit the area in which measurements occur. Unlike reducing the active area, reducing the measurement region does not increase the maximum frame rate of the sensor.

> You can disable regions and force a tool to use the entire active area by unchecking the checkbox next to the **Regions** setting. For more information on active area, see *Active Area* on page 94.

All tools provide region settings under the upper, tool-level **Parameters** tab. This region applies to all of a tool's measurements. Region settings are sometimes found within expandable feature sections in a tool's panel.

Some of LMI's more recent tools provide "flexible" regions, which in addition to rectangular regions let you create circular and elliptical regions (which can optionally be annular) and polygon regions. These tools also let you use Surface and Surface Intensity data as masks. As of this writing, the following tools have flexible regions:

- Surface Direction Filter
- Surface Filter
- Surface Flatness
- Surface Mask
- Surface OCR
- Surface Segmentation

Other tools are currently limited to rectangular regions. However, you can get "flexible regions" in a tool that doesn't directly support them by using the Surface Mask tool, and using that tool's output as the other tool's input. For more information, see *Mask* on page 391.

For information on setting "flexible" regions, see *Flexible Regions* on the next page.

> In 2D mode, the tool region defaults to the center of the current data view, not the global field of view. In 3D mode, the region defaults to the global field of view.
>
> Use the region reset button () to set the size of a region to its default. This is useful after zooming in or out in the data viewer.

Standard Regions

The standard regions are limited to rectangles or boxes.

To configure standard regions:

1. Go to the **Measure** page by clicking on the **Measure** icon.

   > The scan mode must be set to the type of measurement you need to configure. Otherwise, the wrong tools, or no tools, will be listed on the **Measure** page.

2. In the **Tools** panel, click on a tool in the tool list.

3. Configure the region using the mouse in the data viewer.

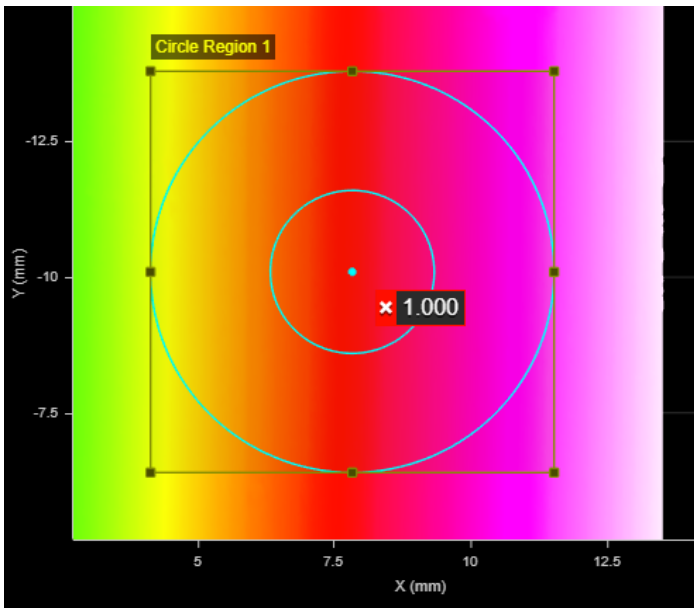
You can also configure regions manually by clicking the expand button (☰) and entering values in the fields. This is useful if you need to set precise values.

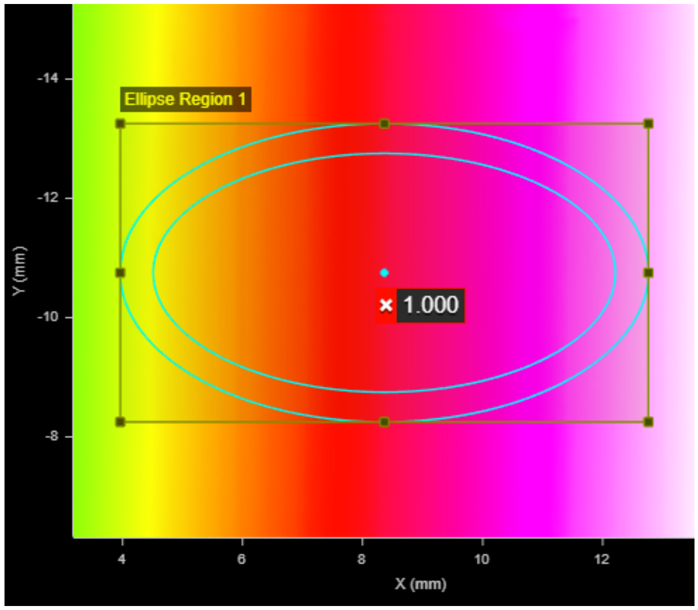
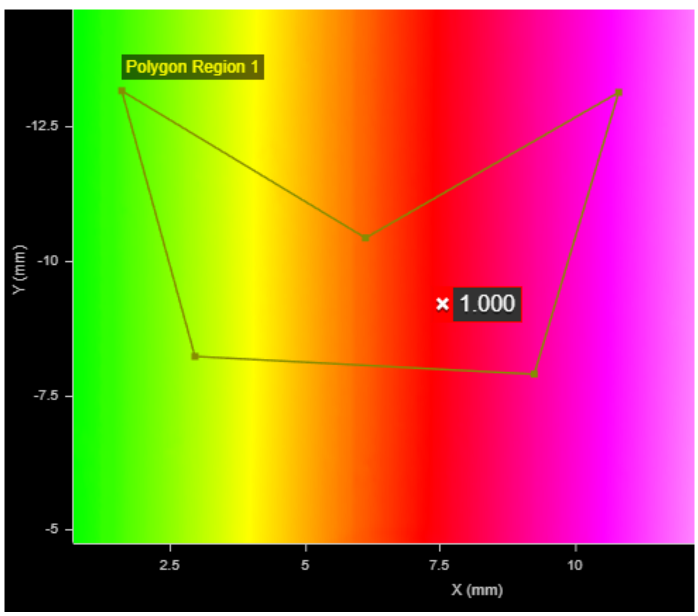
Flexible Regions

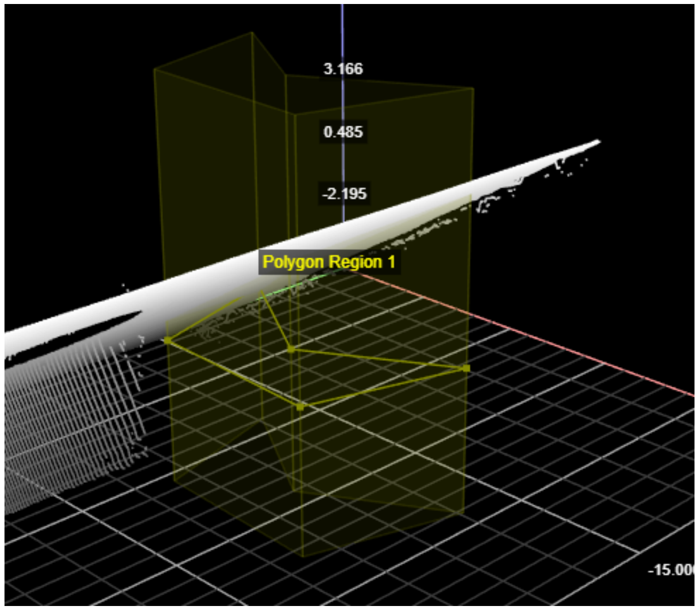
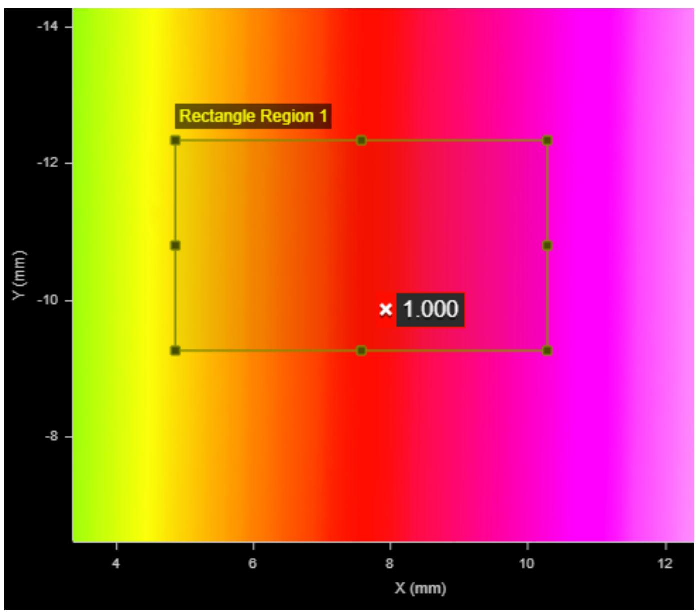
The following parameters are available in tools that support flexible regions

Flexible Region Parameters

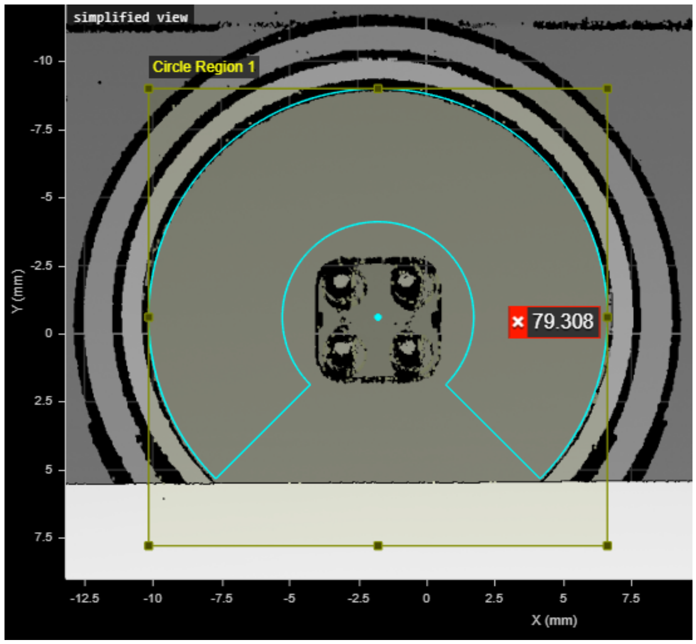
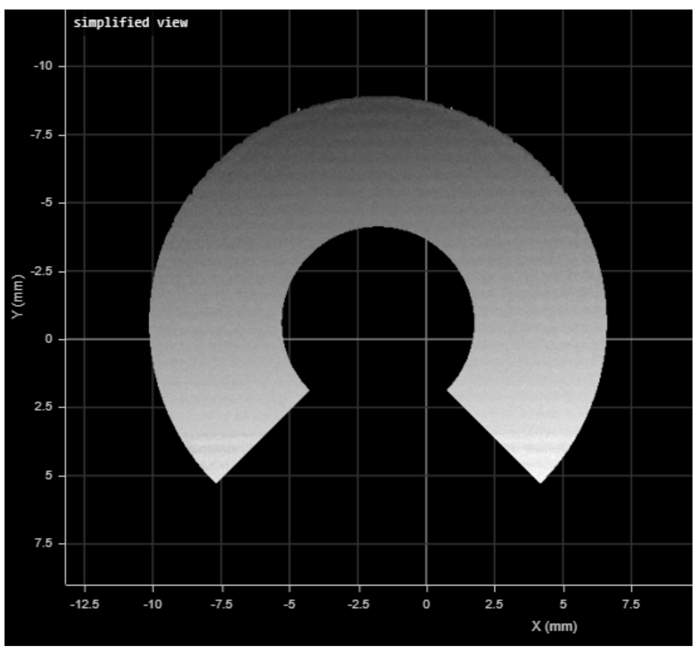
Parameter	Description
Number of Regions	The number of regions the tool uses to extract surface data. You can define up to 15 or 16 regions. This parameter is not available in some tools. When you specify more than one region, the regions are initially stacked on top of one another, in the same location.

Parameter	Description
Mask Type {n} Region Type {n}	For each mask (in the Surface Mask tool) or region, the type. Regions can overlap. One of the following. (For more information on the settings you use with the Circle and Ellipse types, see *Working with Circular and Elliptical Regions* on page 159. **Circle** Extracts a circular region from the surface data, constrained by a square region. Set the region's inner circle (inner cyan circle below) using the **Inner Circle Diameter** parameter to extract annular data. Use the **Sector Start Angle** and **Sector Angle Range** settings to extract a partial circular or elliptical region.  **Ellipse** Extracts an elliptical region from the surface data, constrained by a square or rectangular region. Set the region's inner ellipse (inner cyan ellipse below) using the **Inner Ellipse Major Axis** and **Inner Ellipse Minor Axis** parameters to extract annular data. Use the **Sector Start Angle** and **Sector Angle Range** settings to extract a partial circular or elliptical region.

Parameter	Description
	 **Polygon** Extracts a polygonal region with the number of vertices specified in **Vertex Count**. You can define the shape of the polygon using a mouse in the data viewer, dragging and dropping the vertex points.  Note that you can't adjust the height of a polygon region: it occupies the entire vertical space available:

Parameter	Description
	 **Rectangle** Extracts a rectangular region from the surface data.  **Surface** Uses the Surface data you select in **Mask Source** to create a mask. **Surface Intensity** Uses the intensity data you select in **Mask Source** to create a mask. Set the **Low Threshold** and **High Threshold** parameters as required.

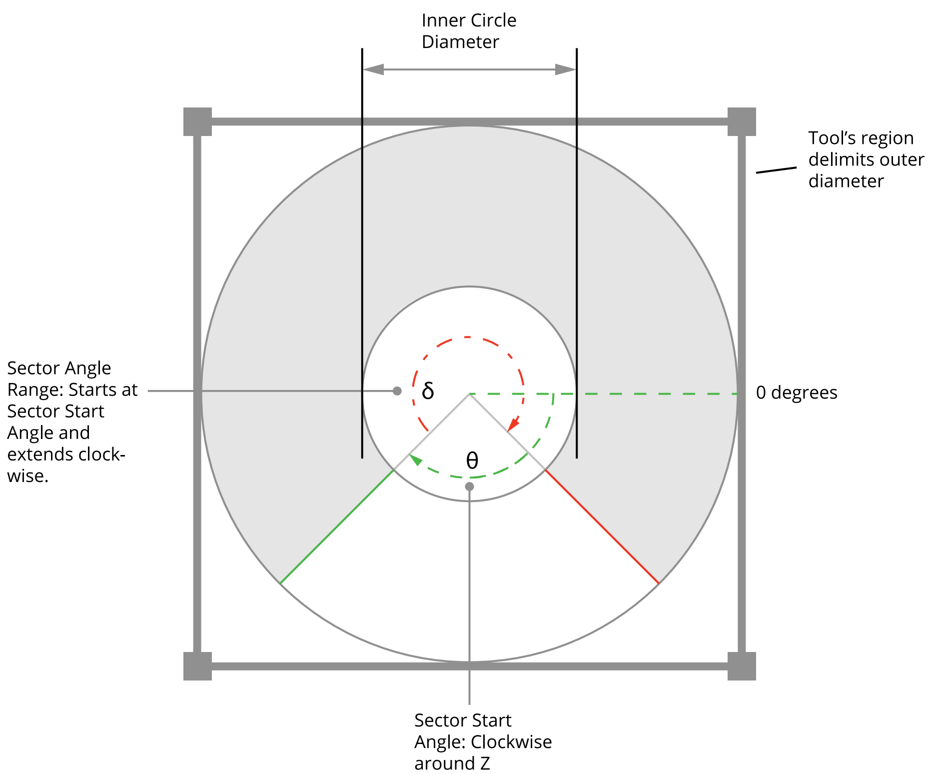
Parameter	Description
Inner Circle Diameter	Only available when **Region Type {n}** is set to **Circle**. Defines the diameter of the inner circle. Set this parameter to a value greater than 0 to extract a ring of data. Set this parameter to 0 to extract a circle of data.
Inner Ellipse Major Axis Inner Ellipse Minor Axis	Only available when **Region Type {n}** is set to **Ellipse**. These parameters define the major and minor axes of the inner ellipse, respectively Set this parameter to a value greater than 0 to extract a ring of data. Set this parameter to 0 to extract an elliptical disk of data.

Parameter	Description
Sector Start Angle Sector Angle Range	Only available when **Region Type {n}** is set to **Circle** or **Ellipse** Use these parameters together to extract a partial ring of data. **Sector Start Angle** controls the starting angle of the data, whereas **Sector Angle Range** controls the length of the arc. Note that the angles and ranges in these parameters are measured clockwise around Z, where 0 degrees is along the positive X axis. For example, in the first image below, **Sector Start Angle** is set to 135, and **Sector Angle Range** is set to 270. The resulting extracted partial ring (or annular data) is shown below that.  

Parameter	Description
	For more information on how these settings work together, see *Working with Circular and Elliptical Regions* below.
Mask Source	Only available when **Region Type {n}** is set to **Surface** or **Surface Intensity**. The Surface or Surface Intensity data the tool uses to create a mask.
Low Threshold High Threshold	Only available when **Region Type {n}** is set to **Surface Intensity**. The low and high thresholds the tool uses in combination with the intensity mask.

Working with Circular and Elliptical Regions

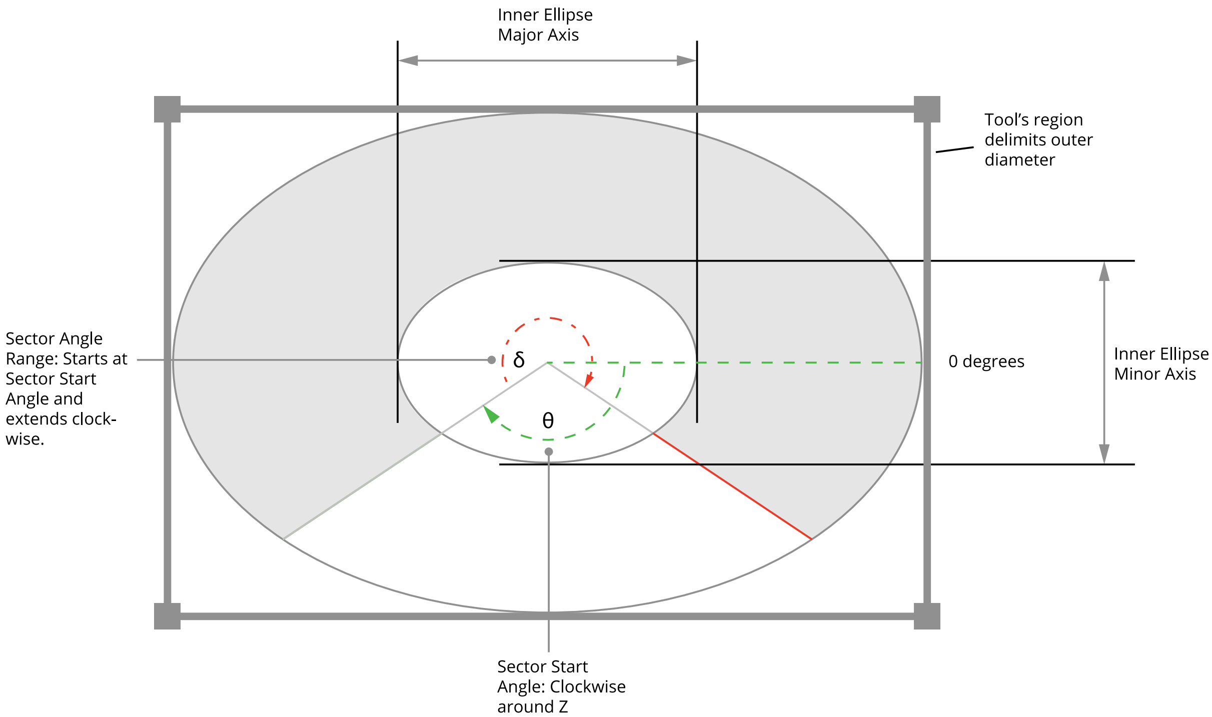
When you set a region's type to Circle or Ellipse, the tool displays several additional settings that work together to define the region. **Sector Start Angle** and **Sector Angle Range** work together to define the start and end of a partial circular/elliptical region (solid or annular). A region will be annular if **Inner Circle Diameter** is non-zero. Note that the "length" of the partial region extends *from* the start angle. In the following illustration, the start angle (θ) is 135 degrees relative to the 0-degree point indicated below, and the region extends 270 degrees (δ) from that, clockwise around Z.



Sector Start Angle starts at the 0-degree point around Z.

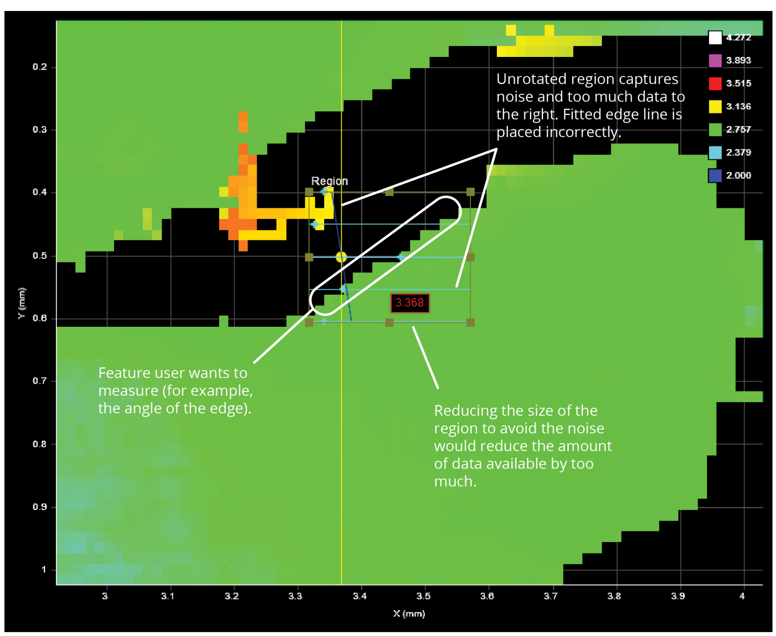
Note that the angles defining a partial circular/elliptical region are relative to the *region*, and not the sensor's coordinate system. So a region rotated 30 degrees using its **Z Angle** setting rotates the start angle and angle range by 30 degrees.

When you set a region type to Ellipse, instead of the inner circle diameter, you must set the major and minor axes of the inner ellipse.



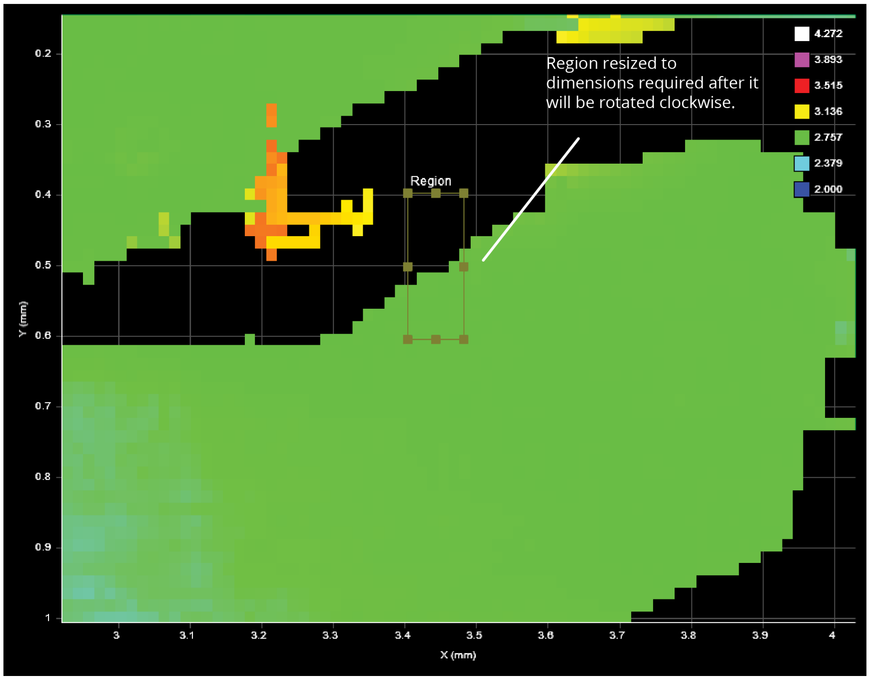
Region Rotation

The measurement region of some tools can be rotated by setting the region's **Z Angle** to better accommodate features that are on an angle on a target. By rotating the measurement region, data not related to the feature can often be excluded, improving accuracy of measurements.

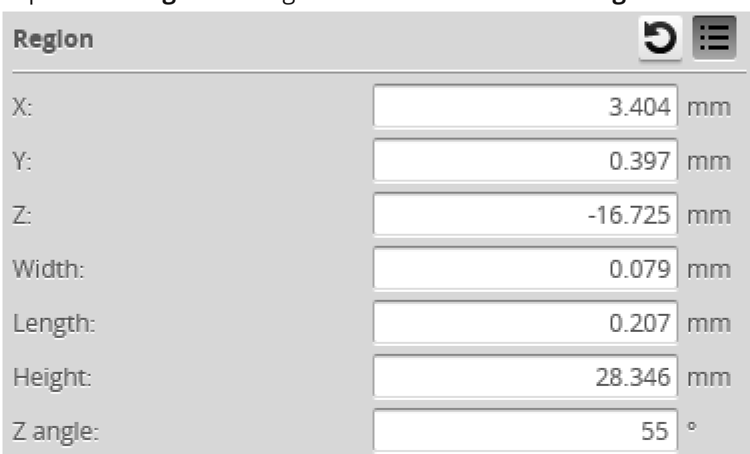


To rotate measurement regions:

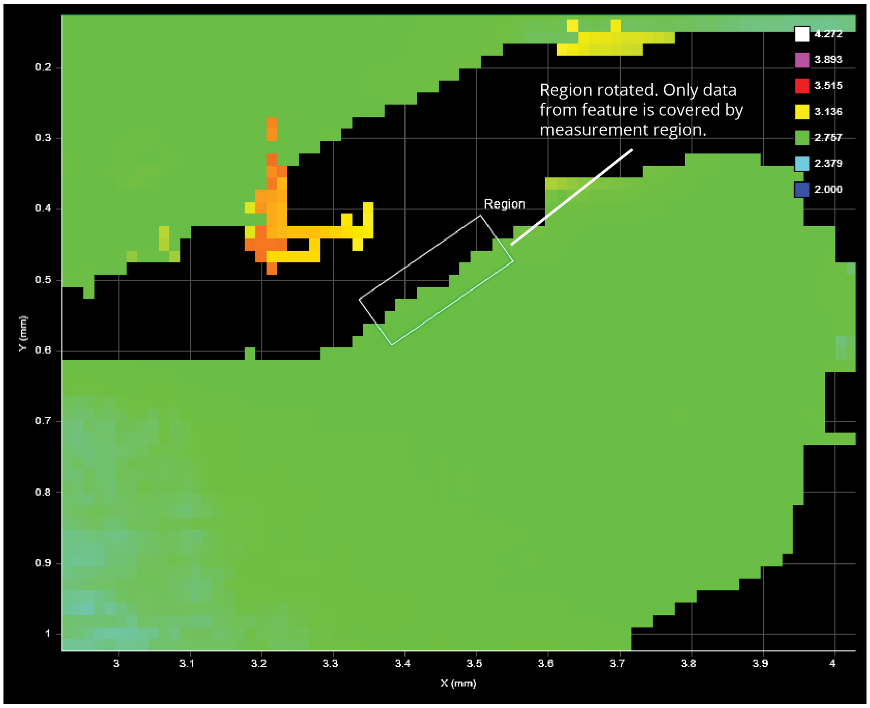
1. Determine the length and width of the region that will be required once it is rotated.



2. Expand the **Region** setting and then set a value in **Z Angle**.



The region rotates clockwise around the Z axis relative to the X axis.



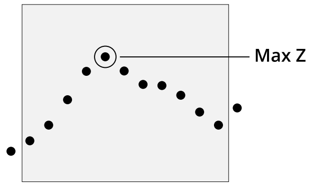
Once the region has been rotated, you can modify its size and location in the data viewer using the mouse. You can also modify its dimensions and its location manually by changing the region's values in the **Region** setting.

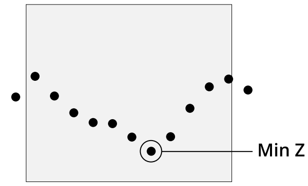
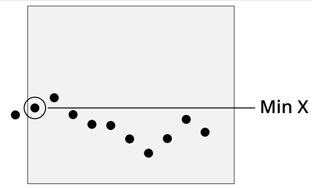
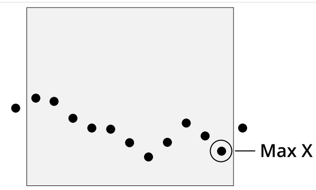
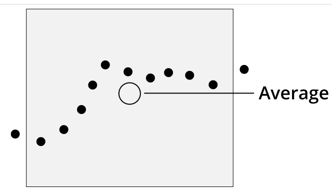
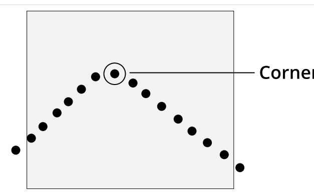
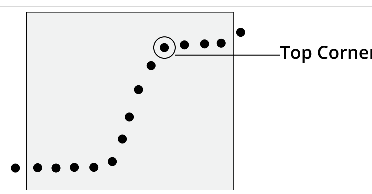
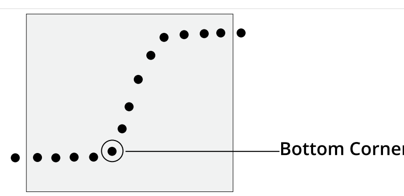
> Some tools let you disable regions entirely and force the measurement tool to use the entire active area by unchecking the checkbox next to the **Regions** setting. For more information on active area, see *Active Area* on page 94.

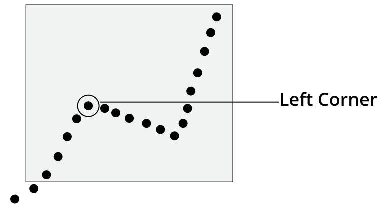
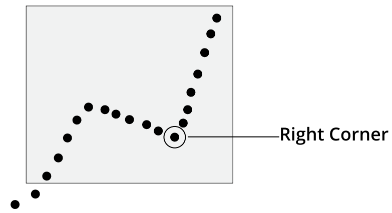
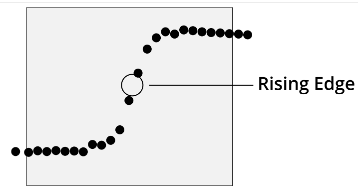
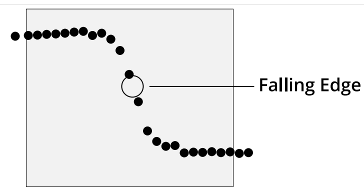
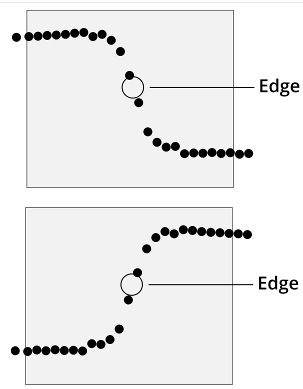
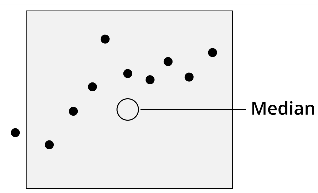
Feature Points

Dimensional and positional measurements detect *feature points* found within the defined measurement region and then compare measurement values taken at the selected point with minimum and maximum thresholds to produce a *decision*. Feature points are selected in one or more **Feature** dropdowns in a tool and are used for all of the tool's measurements.

The following types of points can be identified in a measurement region.

Point Type	Examples
Max Z Finds the point with the maximum Z value in the region of interest.	

Point Type	Examples
Min Z Finds the point with the minimum Z value in the region of interest.	
Min X Finds the point with the minimum X value in the region of interest.	
Max X Finds the point with the maximum X value in the region of interest.	
Average Determines the average location of points in the region of interest.	
Corner Finds a dominant corner in the region of interest, where corner is defined as a change in profile slope.	
Top Corner Finds the top-most corner in the region of interest, where corner is defined as a change in profile shape.	
Bottom Corner Finds the bottom-most corner in the region of interest, where corner is defined as a change in profile shape.	

Point Type	Examples
Left Corner Finds the left-most corner in the region of interest, where corner is defined as a change in profile shape.	
Right Corner Finds the right-most corner in the region of interest, where corner is defined as a change in profile shape.	
Rising Edge Finds a rising edge in the region of interest (moving from left to right).	
Falling Edge Finds a falling edge in the region of interest (moving from left to right).	
Any Edge Finds a rising or falling edge in the region of interest.	
Median Determines the median location of points in the region of interest.	

Geometric Features

Most Surface tools, and many Profile tools, can output features that Feature tools can take as input to produce measurements. These features are called *geometric features*. Feature tools use these entities to produce measurements based on more complex geometry. (For more information on Feature tools, see *Feature Measurement* on page 484.)

Gocator's measurement tools can currently generate the following kinds of geometric features:

Points: A 2D or 3D point. Can be used for point-to-point or point-to-line measurements.

Lines: A straight line that is infinitely long. Useful for locating the orientation of an enclosure or part, or to intersect with another line to form a reference point that can be consumed by a Feature tool.

Planes: A plane extracted from a surface. Can be used for point-to-plane distance or line-plane intersection measurements.

Circles: A circle extracted from a sphere.

The following tables list the tools that can generate geometric features. (Tools that can't generate geometric features are excluded.)

Geometric features generated by Surface tools

Tool	Point	Line	Plane	Circle
Bounding Box	X			
Countersunk Hole	X			
Edge	X	X		
Ellipse	X	X		
Hole	X			
Opening	X			
Plane			X	
Position	X			
Segmentation	X			
Sphere	X			X
Stud	X			
Volume				

The Circle geometric feature currently cannot be used by any of the built-in Feature tools.

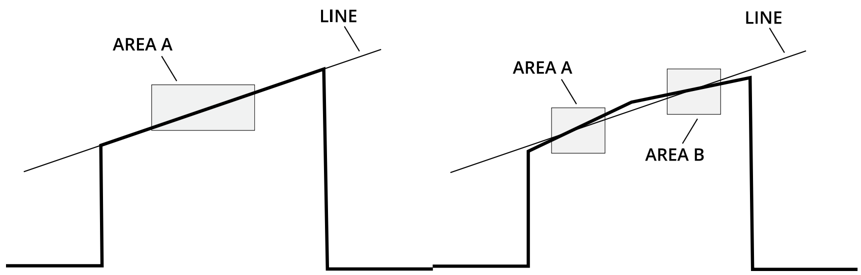
Geometric features generated by Profile tools

Tool	Point	Line
Area	X	
Bounding Box	X	
Circle	X	
Intersect	X	X
Line	X	X
Position	X	

The [Feature Intersect] tool can also produce an intersect point. [Script tools] do not currently take geometric features as input.

Fit Lines

Some measurements involve estimating lines in order to measure angles or intersection points. A fit line can be calculated using data from either one or two fit areas.

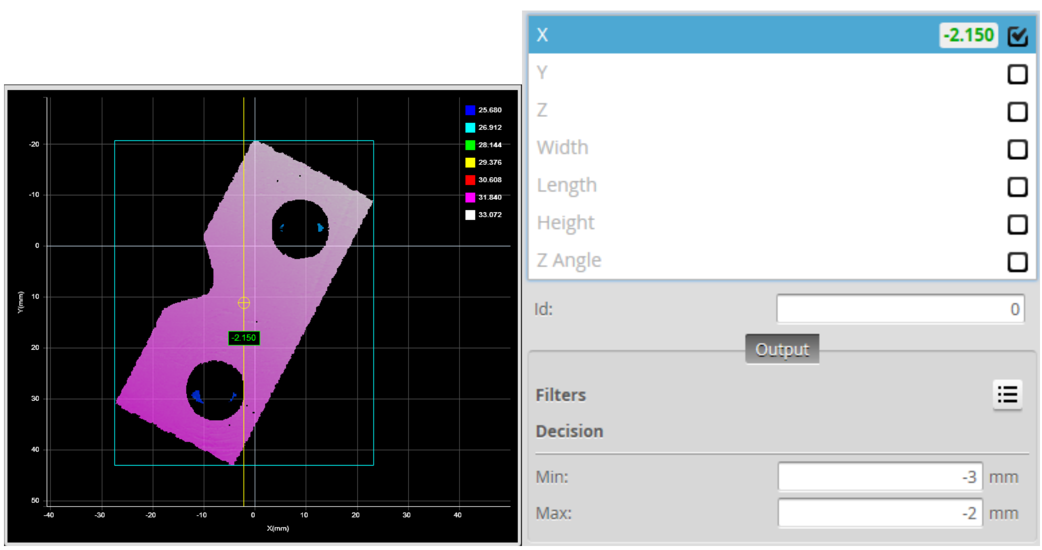


A line can be defined using one or two areas. Two areas can be used to bypass discontinuity in a line segment.

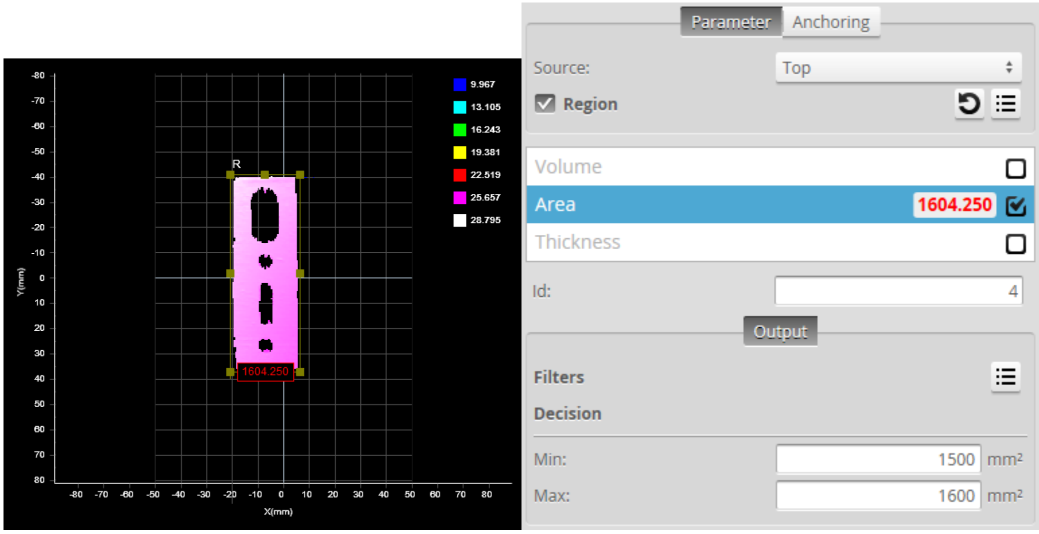
Decisions

Results from a measurement can be compared against minimum and maximum thresholds to generate *pass* / *fail* decisions. The decision state is *pass* if a measurement value is between the minimum and maximum threshold. In the data viewer and next to the measurement, these values are displayed in green. Otherwise, the decision state is *fail*. In the user interface, these values are displayed in red.

All measurements provide decision settings under the **Output** tab.



Value (-2.150) within decision thresholds (Min: -3, Max: -2). Decision: Pass



Value (1604.250) outside decision thresholds (Min: 1500, Max: 1600). Decision: Fail

Along with measurement values, decisions can be sent to external programs and devices. In particular, decisions are often used with digital outputs to trigger an external event in response to a measurement. See *Output* on page 511 for more information on transmitting values and decisions.

To configure decisions:

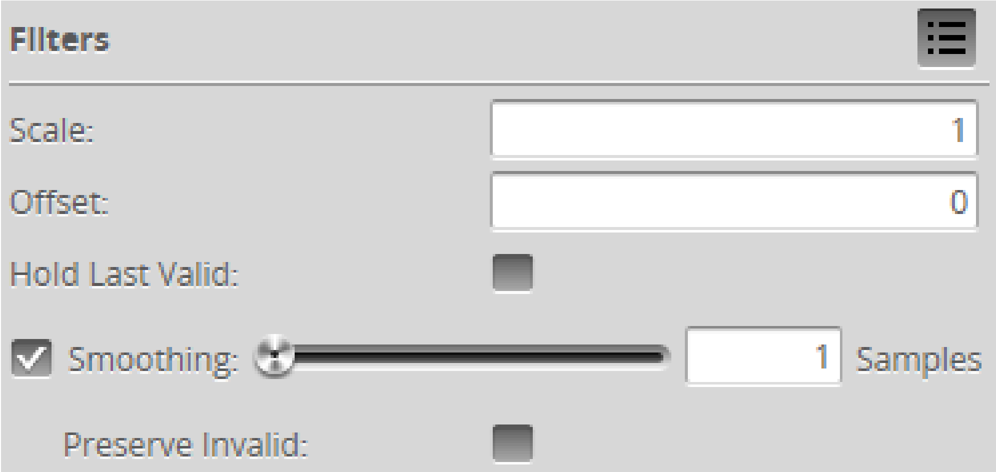
1. Go to the **Measure** page by clicking on the **Measure** icon.

   > The scan mode must be set to the type of measurement you need to configure. Otherwise, the wrong tools, or no tools, will be listed on the **Measure** page.

2. In the **Tools** panel, click on a tool in the tool list.
3. In the measurement list, select a measurement.
   To select a measurement, it must be enabled. See *Enabling and Disabling Measurements* on page 174 for instructions on how to enable a measurement.
4. Click on the **Output** tab.
   For some measurements, only the **Output** tab is displayed.
5. Enter values in the **Min** and **Max** fields.

Filters

Filters can be applied to measurement values before they are output from the Gocator sensors.



All measurements provide filter settings under the **Output** tab. The following settings are available.

Filter	Description
Scale and Offset	The **Scale** and **Offset** settings are applied to a measurement value according to the following formula: Scale * Value + Offset **Scale** and **Offset** can be used to transform the output without the need to write a script. For example, to convert the measurement value from millimeters to thousands of an inch, set **Scale** to 39.37. To convert from radius to diameter, set **Scale** to 2. For more information on scripts, see *Scripts* on page 505.
Hold Last Valid	Holds the last valid value when the measurement is invalid.
Smoothing	Averages the *valid* measurements in the number of preceding frames specified in **Samples**. Use this to reduce the impact of random noise on a measurement's output. If **Hold Last Valid** is enabled, the smoothing filter uses the last valid measurement value until a valid value is encountered.
Preserve Invalid	When enabled, smoothing is only applied to valid measurements and not to invalid results: invalid results are not modified and are sent to output as is. When disabled, smoothing is applied to both valid and invalid results. (This setting is only visible when **Smoothing** is enabled.) If **Hold Last Valid** is enabled, results will always be valid, in which case this setting does nothing.

To configure the filters:

1. Go to the **Measure** page by clicking on the **Measure** icon.

🗋	The scan mode must be set to the type of measurement you need to configure. Otherwise, the wrong tools, or no tools, will be listed on the **Measure** page.

2. In the **Tools** panel, click on a tool in the tool list.
3. In the measurement list, select a measurement.
   To select a measurement, it must be enabled. See *Enabling and Disabling Measurements* on page 174 for instructions on how to enable a measurement.
4. Click on the **Output** tab.
   For some measurements, only the **Output** tab is displayed.
5. Expand the **Filters** panel by clicking on the panel header or the ⊕ button.
6. Configure the filters.
   Refer to the table above for a list of the filters.

Measurement Anchoring

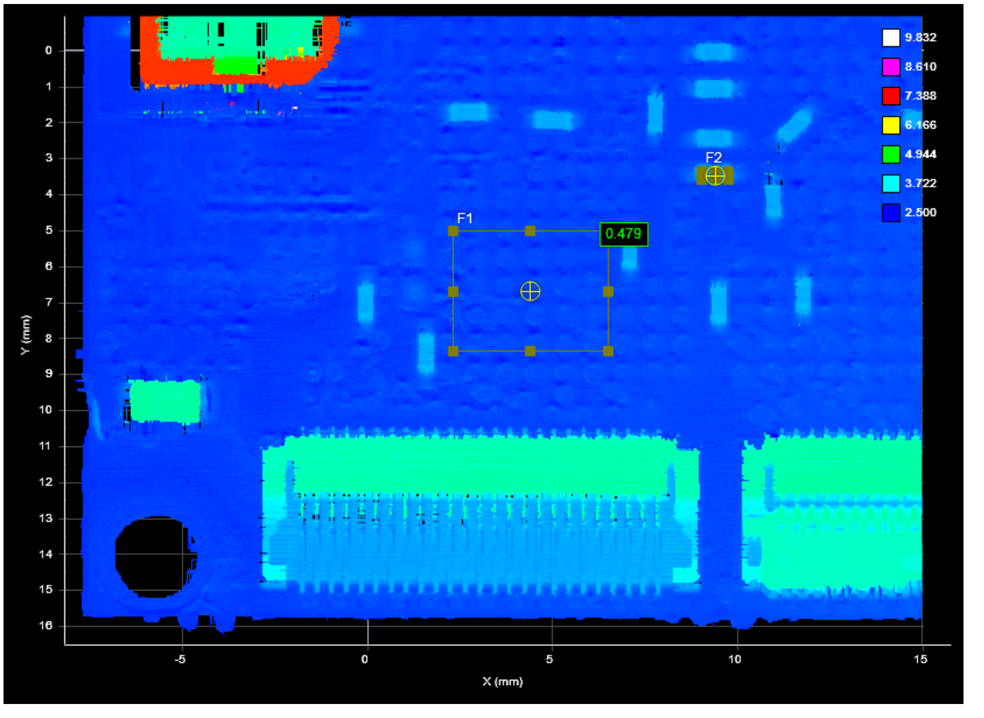
When parts that a sensor is scanning move on a transport mechanism such as a conveyor, their position typically changes from part to part in one or both of the following ways:

- along the X, Y, and Z axes (basically, horizontally and vertically)
- around the Z axis (orientation angle)

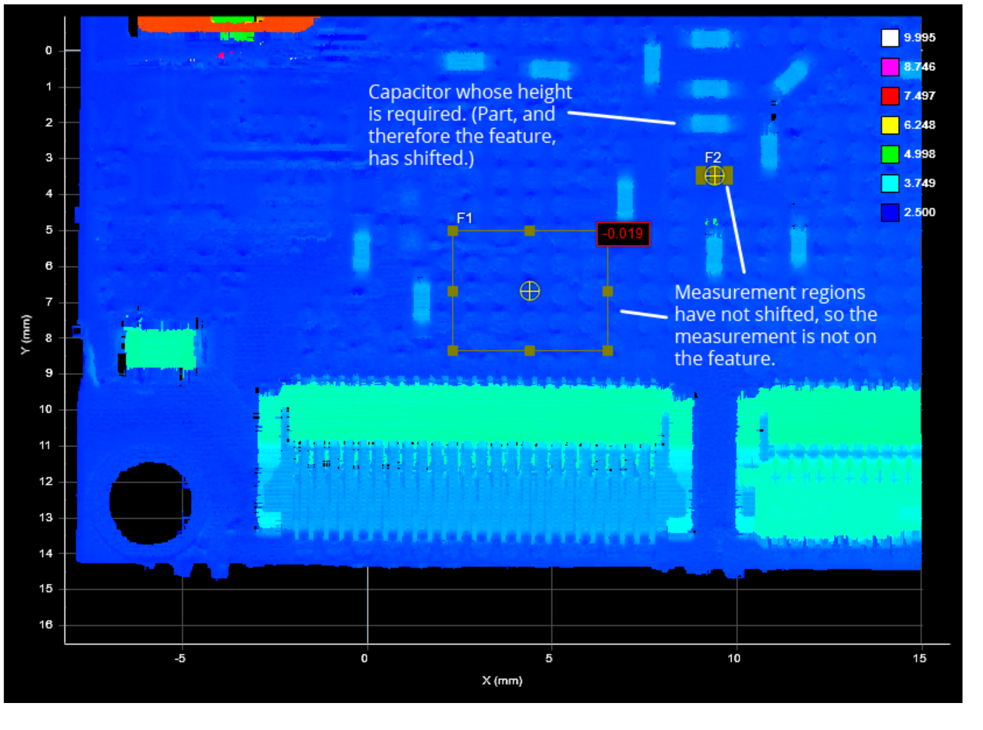
When the position and angle variation between parts is minor—for example, when scanning electronic parts in trays—you can anchor one tool to one or more measurements from another tool to compensate for these minor shifts. As a result, Gocator can correctly place the anchored tool's measurement regions on each part. This increases the repeatability and accuracy of measurements.

> For cases where movement from part to part is more drastic, you can use part matching to compensate. However, in order for part matching to work properly, the entire part typically must be visible in the field of view.

For example, the following image shows a surface scan of a PCB. A Surface Dimension height measurement returns the height of a surface-mount capacitor relative to a nearby surface (the F1 region).

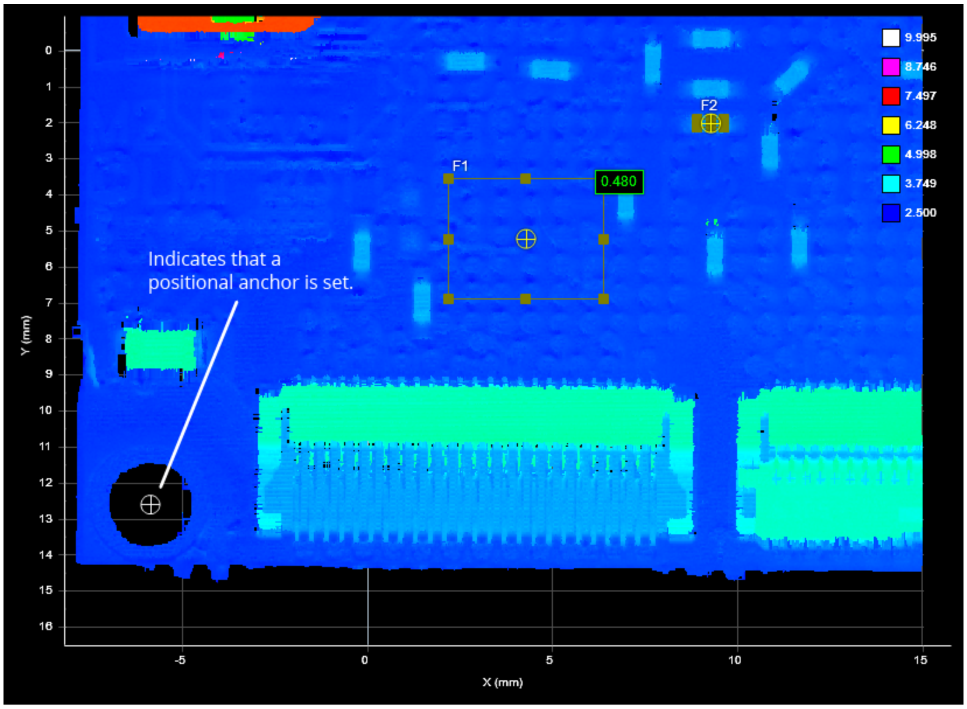


In the following scan, the part has shifted, but the measurement regions remain where they were originally configured, in relation to the sensor or system coordinate system, so the measurement returned is incorrect:

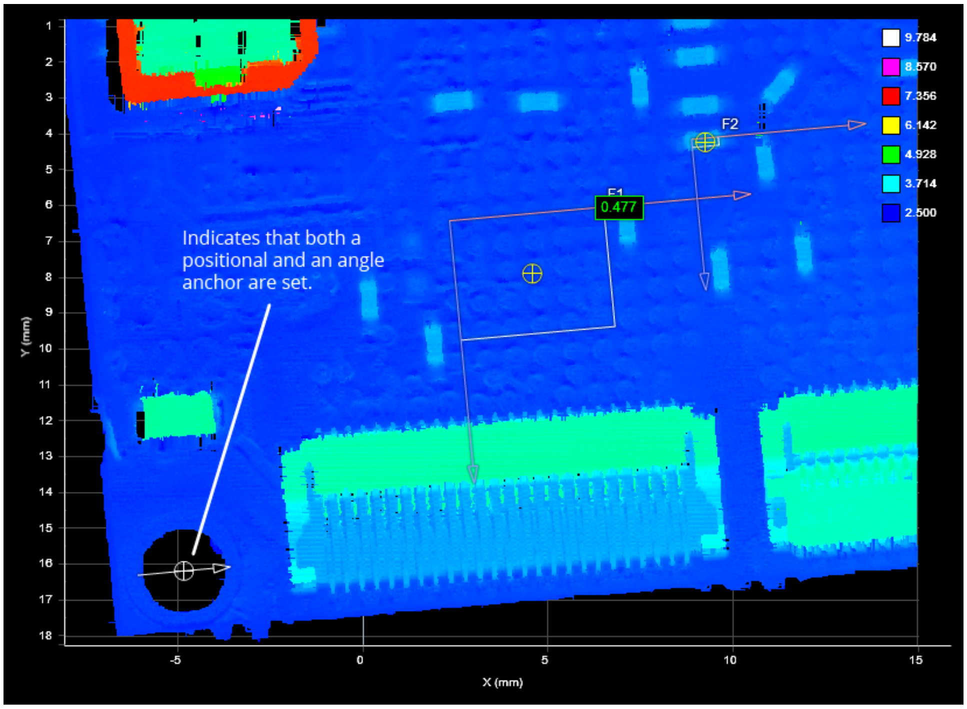


When you set a tool's anchor source, an offset is calculated between the anchored tool and the anchor source. This offset is used for each frame of scanned data: the anchored tool's measurement region is placed in relation to the anchor source, at the calculated offset.

In the following image, after the Surface Dimension tool is anchored to the X and Y measurements from a Surface Hole tool (placed over the hole to the lower left), Gocator compensates for the shift—mostly along the Y axis in this case—and returns a correct measurement, despite the shift.



You can combine the positional anchors (X, Y, or Z measurements) with an angle anchor (a Z Angle measurement) for optimum measurement placement. For example, in the following scan, the part has not only shifted on the XY plane but also rotated around the Z axis. Anchoring the Surface Dimension tool to the Z Angle measurement of a Surface Edge tool (placed on the lower edge in this case) compensates for the rotation, and the anchored tool returns a correct measurement.



🗋	If Z Angle anchoring is used with both X and Y anchoring, the X and Y anchors should come from the same tool.

🗋	If Z Angle anchoring is used without X or Y anchoring, the tool's measurement region rotates around its *center*. If only one of X or Y is used ,the region is rotated around its center and then shifted by the X or Y offset.

Several anchors can be created to run in parallel. For example, you could anchor the measurements of one tool relative to the left edge of a target, and anchor the measurements of another tool relative to the right edge of a target.

You can combine positional anchors (X, Y, or Z) with angle anchors (Z Angle) for optimum measurement placement.

To anchor a profile or surface tool to a measurement:

1. Place a representative target object in the field of view.
   In Profile mode
   a. Use the **Start** or **Snapshot** button to view live profile data to help position the target.

   In Surface mode
   a. Adjust Part Detection settings (see *Part Detection* on page 111) if applicable.
   b. Start the sensor, scan the target, and then stop the sensor.
2. On the **Measure** page, add a suitable tool to act as an anchor.
   A suitable tool is one that returns an X, Y, or Z position or Z Angle as a measurement value.

3. Adjust the anchoring tool's settings and measurement region, and choose a feature type (if applicable).
   You can adjust the measurement region graphically in the data viewer or manually by expanding the **Regions** area.
   The position and size of the anchoring tool’s measurement regions define the zone within which movement will be tracked.

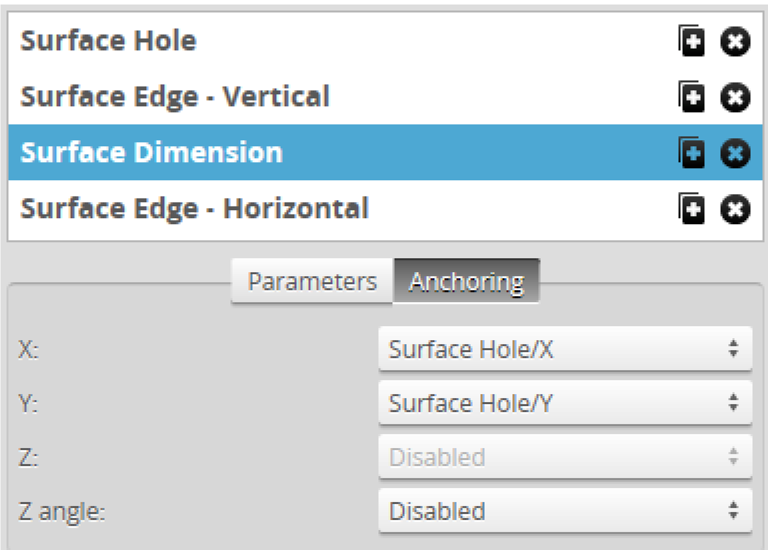
   > If you intend to use angle anchoring and the part in the initial scan is rotated too much, you may need to rotate the anchoring tool's region to accomodate this rotation. For more information on region rotation, see *Regions* on page 152.

4. Add the tool that you want to anchor.
   Any tool can be anchored.

5. Adjust the tool and measurement settings, as well as the measurement regions, on a scan of the representative target.

6. Click on the tool's **Anchoring** tab.

7. Choose an anchor from one of the drop-down boxes.



   If the sensor is running, the anchored tool’s measurement regions are shown in white to indicate the regions are locked to the anchor. The measurement regions of anchored tools cannot be adjusted.
   The anchored tool’s measurement regions are now tracked and will move with the target’s position and angle under the sensor, as long as the anchoring measurement produces a valid measurement value.
   If the anchoring measurement is invalid, for example, if part moves outside its measurement region, the anchored tool will not show the measurement regions at all and an “Invalid-Anchor” message will be displayed in the tool panel.

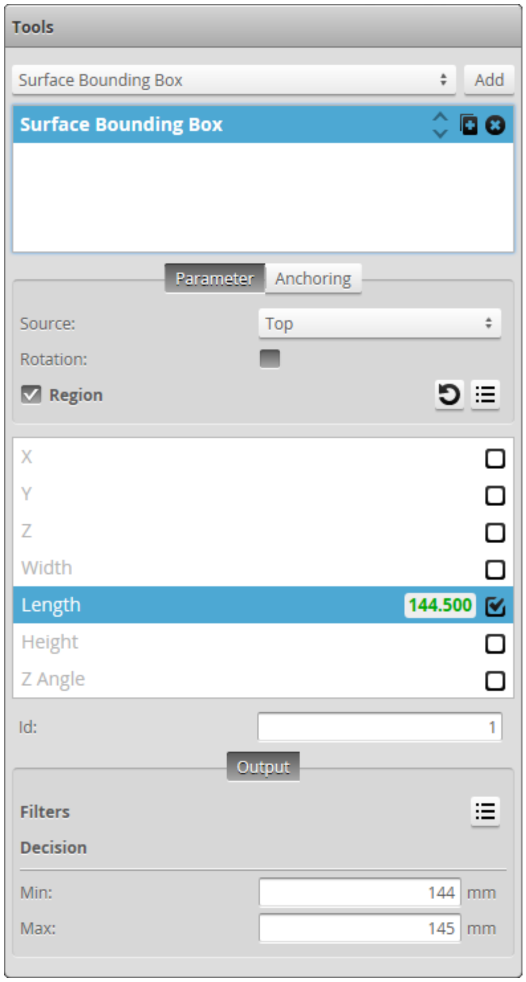
8. Verify that the anchored tool works correctly on other scans of targets in which the part has moved slightly.

To remove an anchor from a tool:

1. Click on the anchored tool's Anchoring tab.
   Select **Disabled** in the X, Y, or Z drop-down.

Enabling and Disabling Measurements

All of the measurements available in a tool are listed in the measurement list in the **Tools** panel after a tool has been added. To configure a measurement, you must enable it.



To enable a measurement:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the measurements list, check the box of the measurement you want to enable.
   The measurement will be enabled and selected. The **Output** tab, which contains output settings will be displayed below the measurements list. For some measurements, a **Parameters** tab, which contains measurement-specific parameters, will also be displayed.

To disable a measurement:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Surface mode in the **Scan Mode** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the measurement list, uncheck the box of the measurement you want to disable.
   The measurement will be disabled and the **Output** tab (and the **Parameters** tab if it was available) will be hidden.

Editing Tool, Input, or Output Names

You can change the names of tools you add in Gocator. You can also change the names of their measurements. This allows multiple instances of tools and measurements of the same type to be more easily distinguished in the Gocator web interface. The measurement name is also referenced by the Script tool.

To change a tool or measurement name:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. Do one of the following:
   - **Tool**: In the tool list, double-click the tool name you want to change
   - **Measurement**: In a tool's measurement list, double-click the measurement name you want to change.
5. Type a new name.
6. Press the Tab or Enter key, or click outside the field.
   The name will be changed.

Changing a Measurement ID

The measurement ID is used to uniquely identify a measurement in the Gocator protocol or in the SDK. The value **must** be unique among all measurements.

To edit a measurement ID:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the measurement list, select a measurement.

To select a measurement, it must be enabled. See *Enabling and Disabling Measurements* on page 174 for instructions on how to enable a measurement.

5. Click in the ID field.

6. Type a new ID number.
   The value must be unique among all measurements.

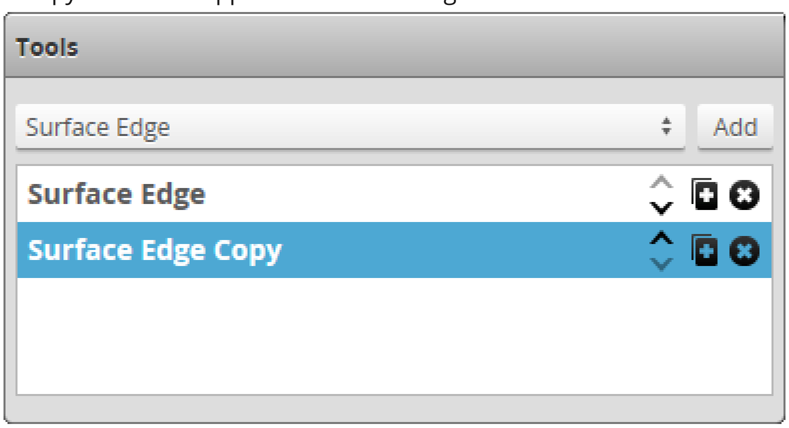
7. Press the Tab or Enter key, or click outside the ID field.
   The measurement ID will be changed.

Duplicating a Tool

You can quickly create a copy of a previously added tool in Gocator. All settings of the original are copied. This is useful, for example, when you need almost identical tools with only minor variations, such as different Min and Max values.

To duplicate a tool:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Profile or Surface mode in the **Scan Mode** panel.
   If one of these modes is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the tool list, click the Duplicate button () of the tool you want to duplicate.
   A copy of the tool appears below the original.



5. Configure the copy as desired and rename it if necessary.
   For information on renaming a tool, see *Editing Tool, Input, or Output Names* on the previous page.

Removing a Tool

Removing a tool removes all of its associated measurements.

To remove a tool:

1. Go to the **Scan** page by clicking on the **Scan** icon.

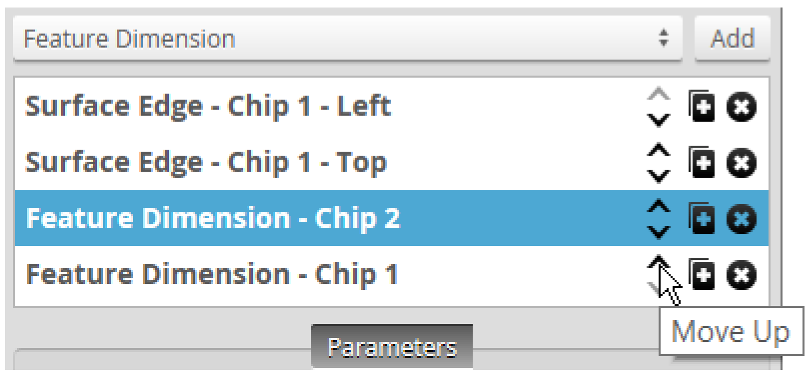
2. Choose Profile or Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.

3. Go to the **Measure** page by clicking on the **Measure** icon.

4. In the tool list, click on the Duplicate button ( ) of the tool you want to duplicate.
   A copy of the tool appears below the original.

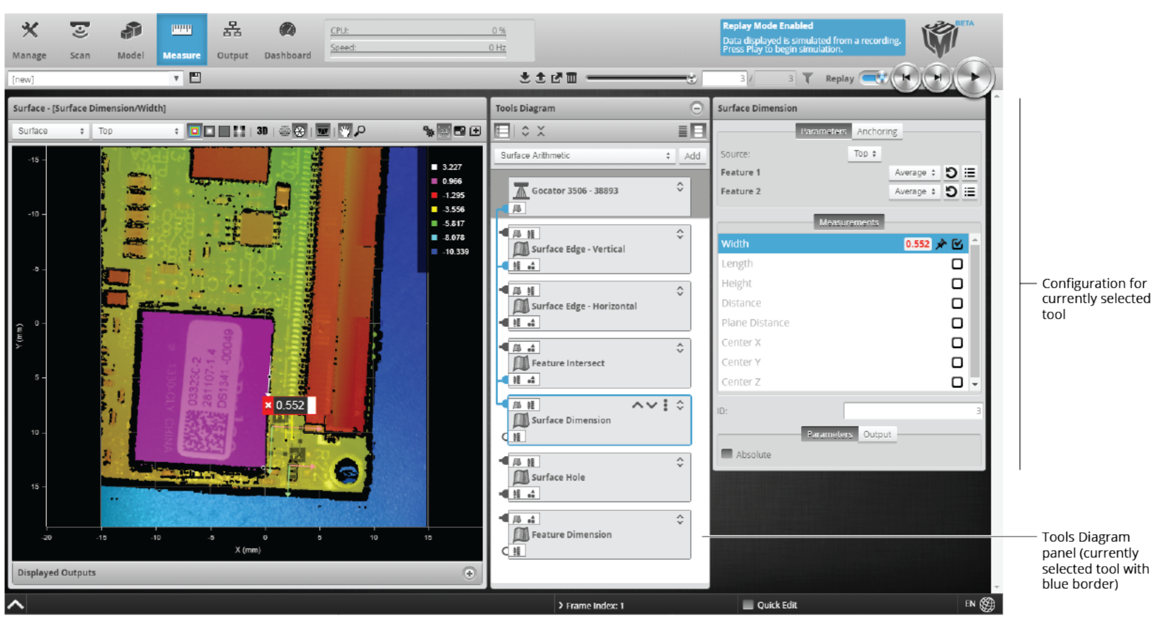
Reordering Tools

When you add or duplicate a tool, the tool is added to the bottom of the list in the **Tools** panel. You can reorder tools in the web interface to organize tools more logically. For example, you could group tools that output geometric features with the tools that use them. Or you could group tools you use as anchors with the tools that use those anchors.



Working with the Tools Diagram

The Tools Diagram provides a visual representation of the data flow in a sensor system (the output from a sensor, and the input and output of tools). It lets you create and view complex tool chains with drag-and-drop and other mouse operations, letting you implement and maintain applications demanding multiple, interconnected tools, quickly and easily.



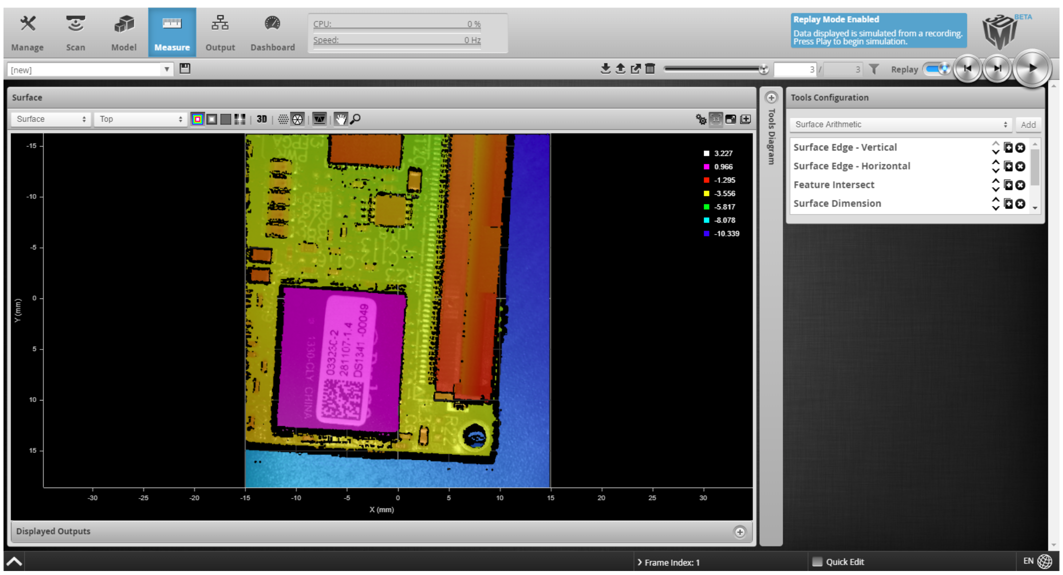
All data types and their relationships between tools are displayed:

- Profile data (either directly from a sensor's output or from tool output)
- Surface data (either directly from a sensor's output or from tool output)
- Measurements (for use as anchors)
- Geometric features
- Tool data output (some data outputs are intended to be consumed only by SDK applications and can't be used as part of a tool chain withing Gocator)

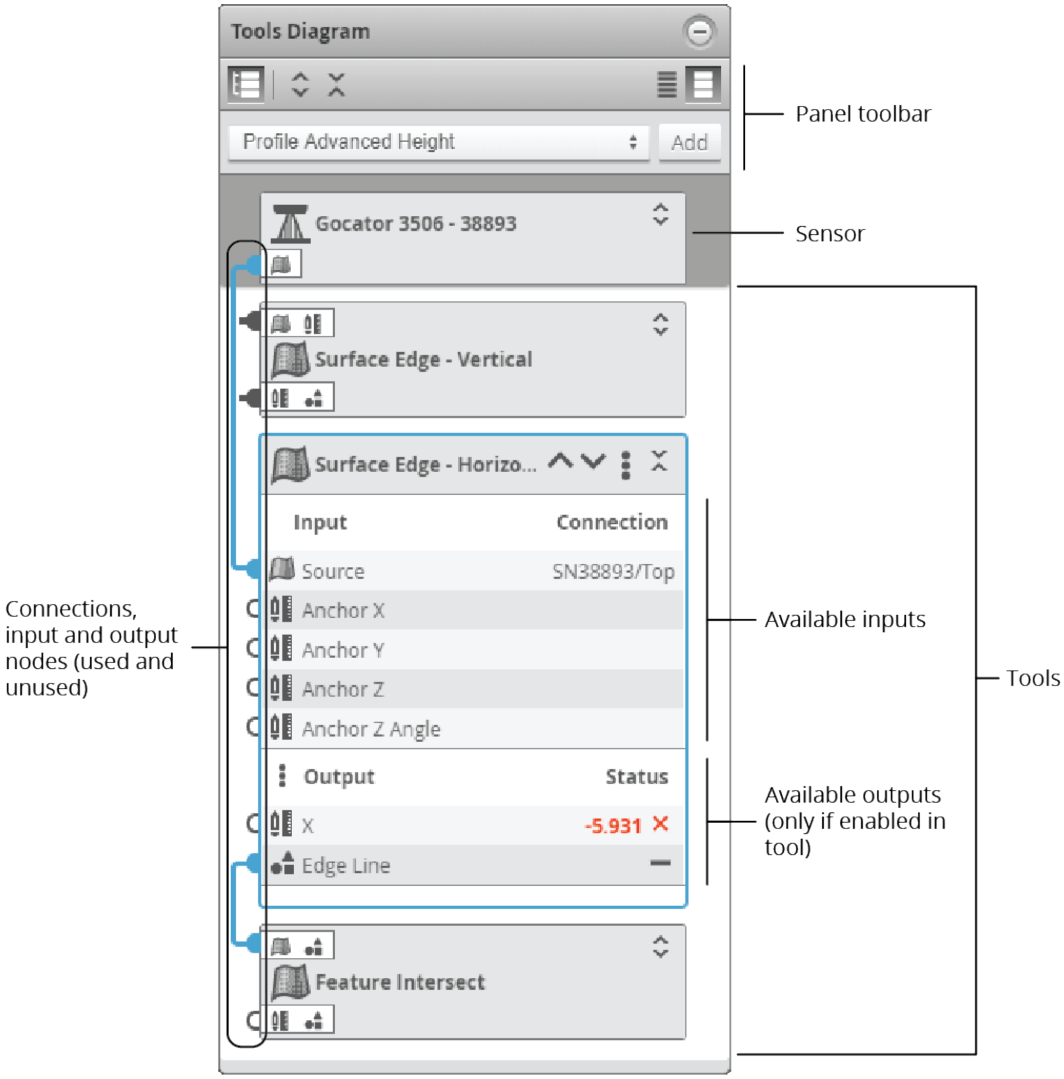
For details on how the Tools Diagram panel displays information, see *Working with the Tools Diagram* on the previous page.

For details on how to connect and disconnect, see *Working with the Tools Diagram* on the previous page and *Disconnecting Tools* on page 197.

The Tools Diagram panel is open by default. When the panel is open, the parameters of the tool selected in the panel are to the right of the Tools Diagram panel. You can close the Tools Diagram panel by clicking the  button at the top of the panel. When you close it, the tool drop-down list and button used to add tools moves to the Tools Configuration panel.

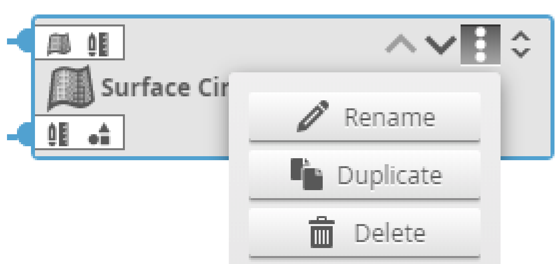


The following illustrates the main aspects of the Tools Diagram panel.



Tools Diagram panel showing sensor, tools, outputs/inputs, and data flow connections.

At the top of a tool, a drop-down menu provides functions to rename, duplicate, and delete the current tool. For more information, see the topics below.



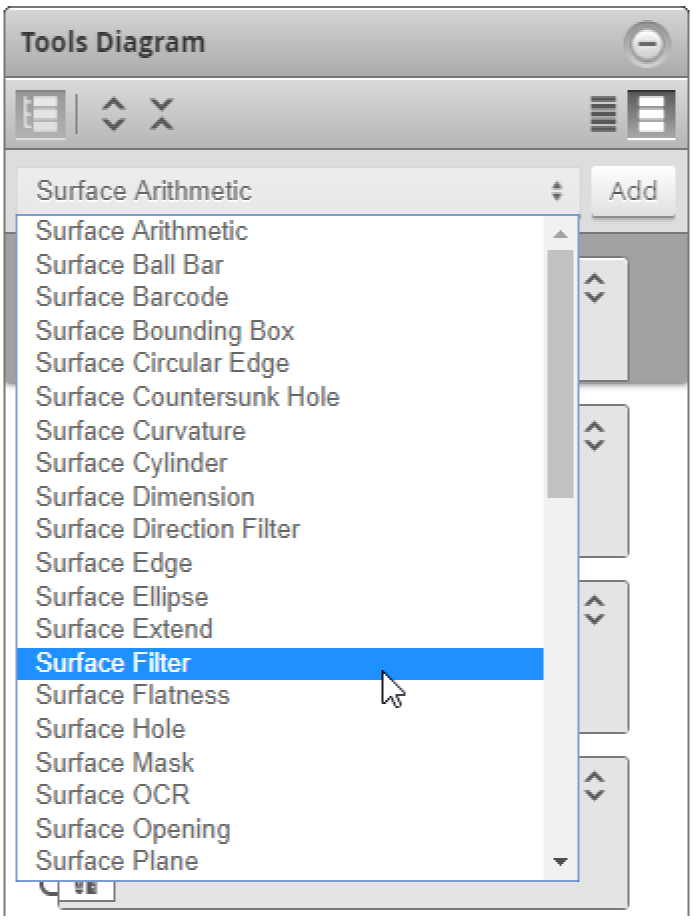
Action menu (collapsed tool)

Adding a Tool

In the Tools Diagram panel, you add a tool using the drop-down and the **Add** button below the panel's toolbar.

To add a tool in the Tools Diagram panel

1. In the drop-down at the top of the panel, choose a tool to add.



2. Click Add.

The tool appears at the bottom of the Tools Diagram panel.

After you have added a tool, you must configure it. For information on configuring tools, see .

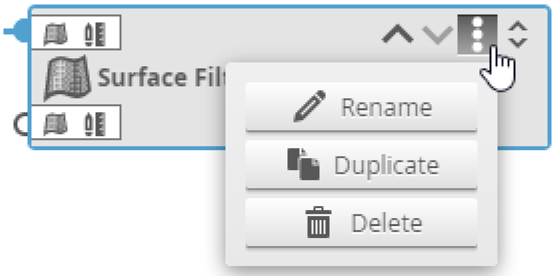
Deleting a Tool

In the Tools Diagram panel, you delete a tool using the Action menu of an individual tool.

To delete a tool in the Tools Diagram panel

1. Click the Action menu icon.

   A context menu appears.

2. In the context menu, choose **Delete**.

   The tool is removed from the Tools Diagram panel.

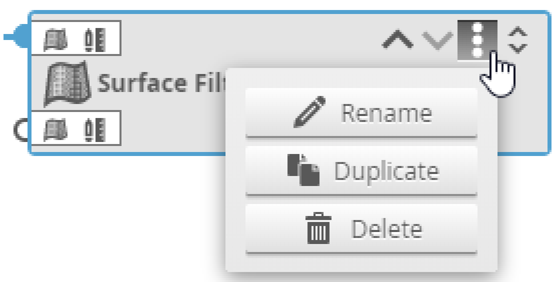
Renaming a Tool

In the Tools Diagram panel, you rename a tool using the Action menu of an indivual tool.

To rename a tool in the Tools Diagram panel

1. Click the Action menu icon.

   A context menu appears.

2. In the context menu, choose **Rename**.
3. In the tool name field, rename to the tool.

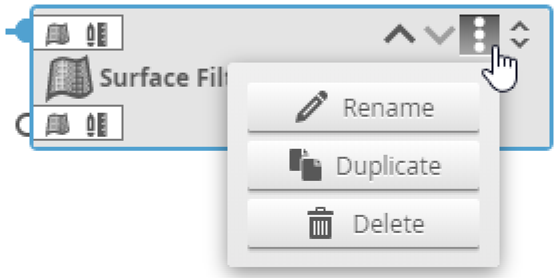
4. Press Enter on the keyboard or click the check icon (see above).

Duplicating a Tool

In the Tools Diagram panel, you duplicate a tool using the Action menu of an individual tool.

To duplicate a tool in the Tools Diagram panel

1. Click the Action menu icon.
2. A context menu appears.

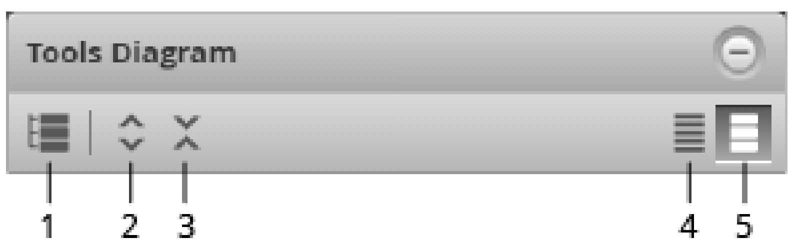


3. In the context menu, choose **Duplicate**.

   A copy of the tool appears below the tool you copied, with "Copy" appended to its name.

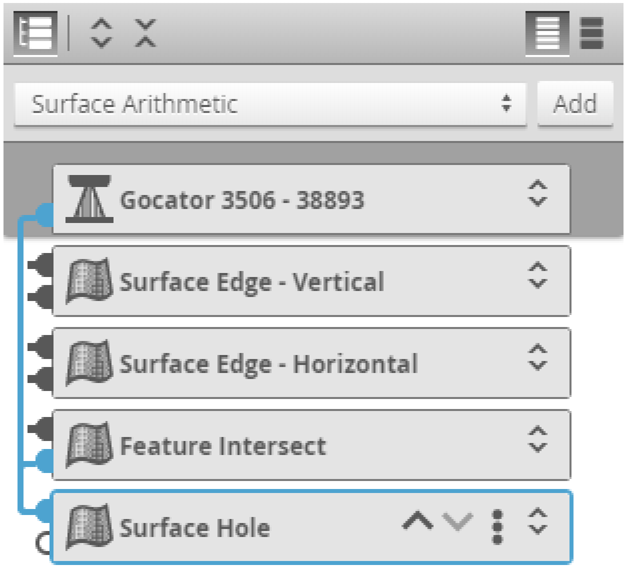
Displaying and Ordering Tools

The buttons at the top of the Tools Diagram panel let you control how the panel displays sensors, tools, and the data flow (tool chain). Buttons at the top of individual tools let you organize the tools in the list, as well as name, duplicate, and delete them.

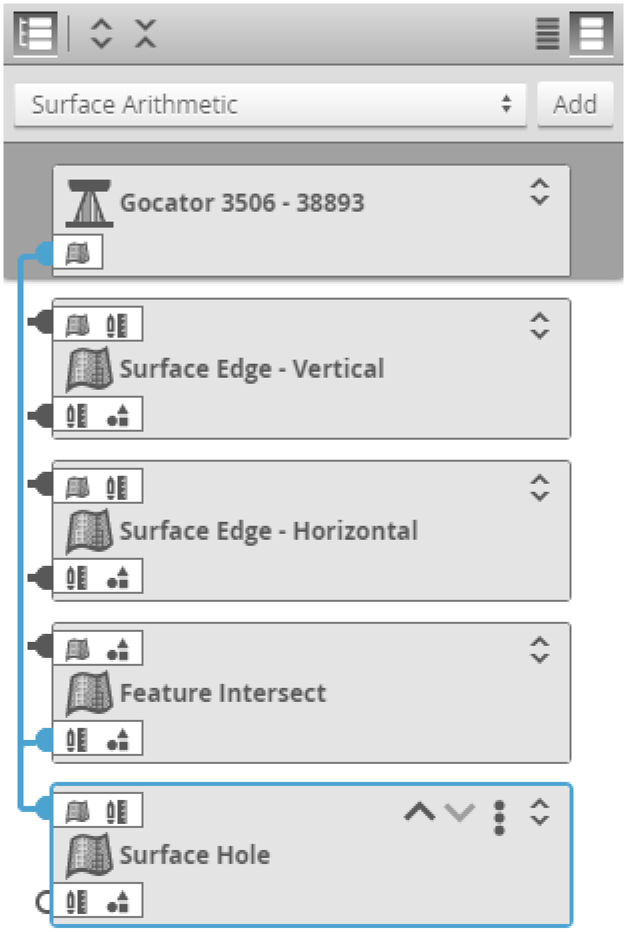


The following describes the toolbar's functions:

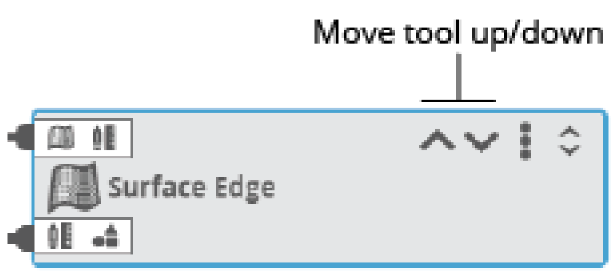
1. **Show/Hide Connections**: Toggles displaying lines showing the data flow related to the selected item (the sensor or a tool). The connection lines let you see at a glance how the tools are chained together. You can highlight subsections of connections to better understand the data flow. For more information see *Understanding the Data Flow in Tool Chains* on page 186. For more information on connecting and disconnecting tools, see *Connecting Tools* on page 191 and *Disconnecting Tools* on page 197.
2. **Open All**: Expands all the sensor and tools in the Tools Diagram panel, displaying a list of available inputs and enabled outputs for each one.
3. **Close All**: Collapses all items in the Tools Diagram panel.
4. **Compact View**: Hides the list of small input and output icons that indicate the types of the inputs and outputs the sensor or a tool has.


Surface Arithmetic
Add
Gocator 3506 - 38893
Surface Edge - Vertical
Surface Edge - Horizontal
Feature Intersect
Surface Hole

5. **Standard View**: Shows small icons that indicate the types of the inputs and outputs the sensor or a tool has. The icons are only shown on collapsed sensors or tools. For a list of inputs and outputs, see *Data Types* below.

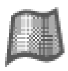
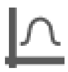
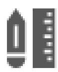
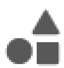


Use the up / down buttons next to the Action menu on individual tools to move the tool up or down in the panel. Note that the order of tools in the Tools Diagram panel does not affect the data flow. However, you can order tools to make the data flow clearer.



Data Types

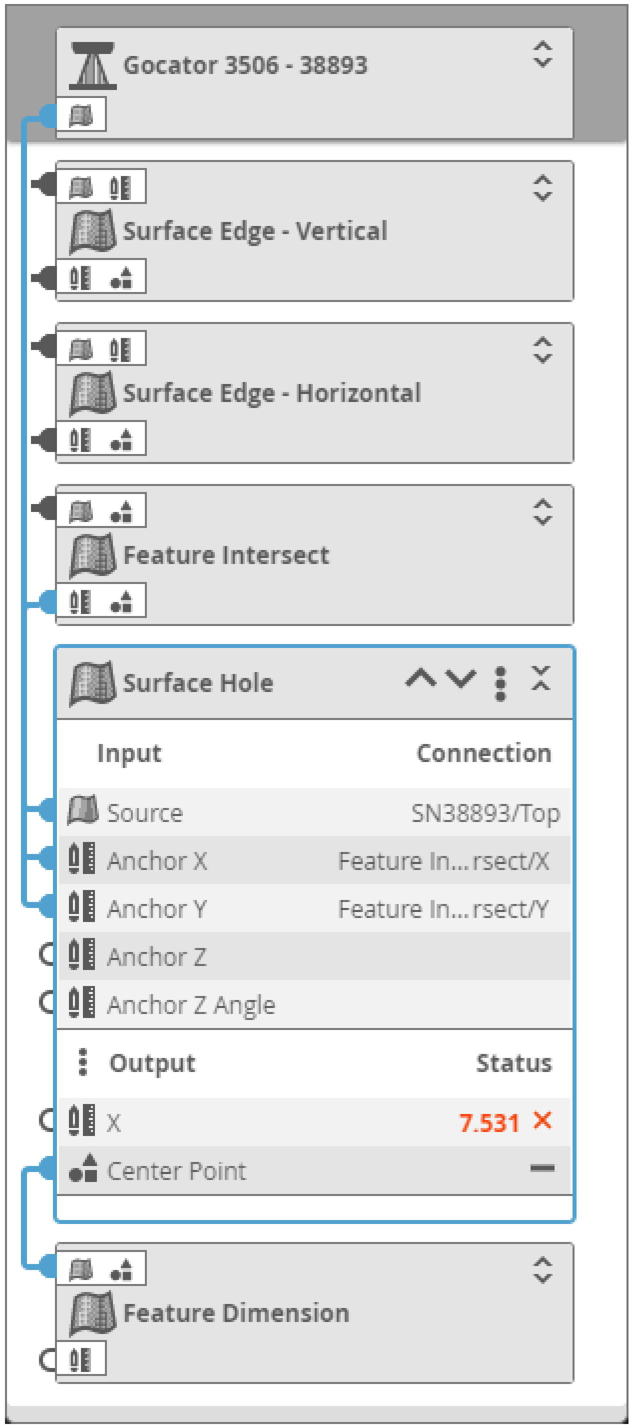
Gocator represents data types in the Tools Diagram panel by an icon. Larger icons indicate the type of a tool (for example, a Profile tool vs. a Surface tool). Smaller icons are used to indicate the types of a tool's inputs and outputs when the Tools Diagram panel is set to Standard view (the small icons are hidden in Compact view); for more information on views, see *Displaying and Ordering Tools* on page 183.

Icon	Description
	Surface data.
	Profile data.
	Range data.
	Measurement.
	Geometric feature.
	Tool data output.

Understanding the Data Flow in Tool Chains

The rectangular elements displayed in the Tools Diagram panel represent a sensor at the top (dark grey area) and any tools you have added below that. Sensors display output connection nodes, whereas tools display both input and output connection nodes.

The appearance of nodes changes depending on whether they are connected and whether they are selected. Connections that are used are filled. Connections that are not used are empty. When a sensor or tool is expanded, you can see which specific inputs or outputs are used and part of the tool chain. For example, in the expanded Surface Circular Edge tool below, we can see that the first three inputs (Source and two anchors, receiving their input from the sensor at the top and from Feature Intersect, respectively) and the Center output are involved in the chain of sensor and tools.

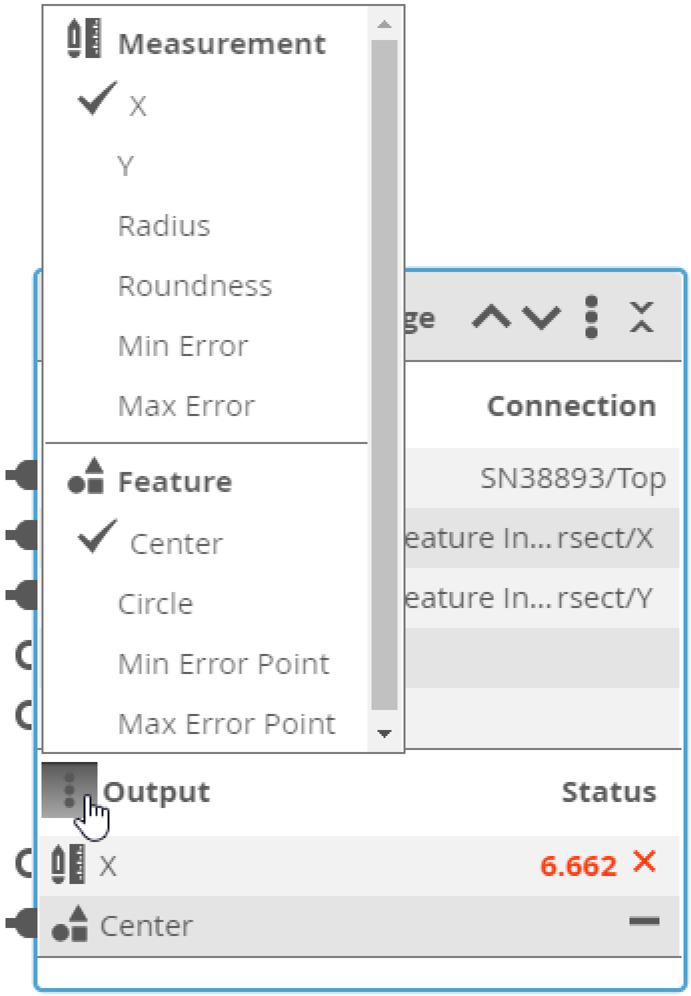


When a tool is collapsed, however, you only know that at least one input or output is used (or none at all). For example, looking at the collapsed Feature Dimension tool at the bottom, we know that at least one input (the connection node at the top) is used, and that none of the tool's outputs are used. Also, we know that inputs and outputs of the three collapsed tools at the top are used, but not exactly which ones.

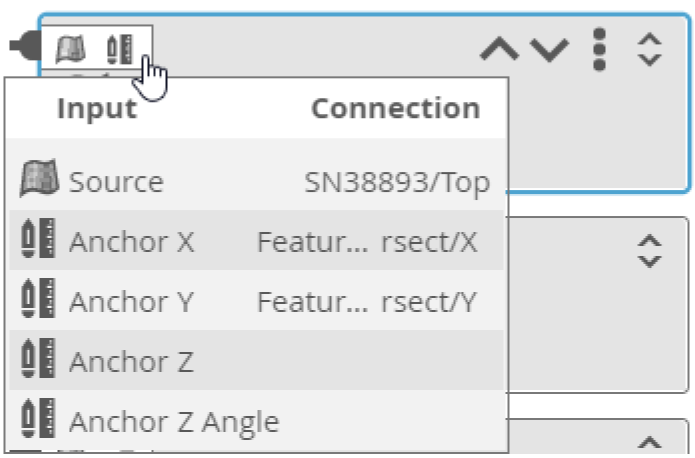
In both cases (collapsed or note), the data flow of the selected item is indicated by dark blue connection lines. For more information, see below.

By default, sensors and tools are collapsed, but you can expand them individually by clicking the expand / collapse button at the top right of a tool to display the complete list of available inputs and outputs. Note that for an output to be listed in the Outputs section, it must be enabled in the tool's configuration: in the tool's Output list, only enabled outputs are listed.

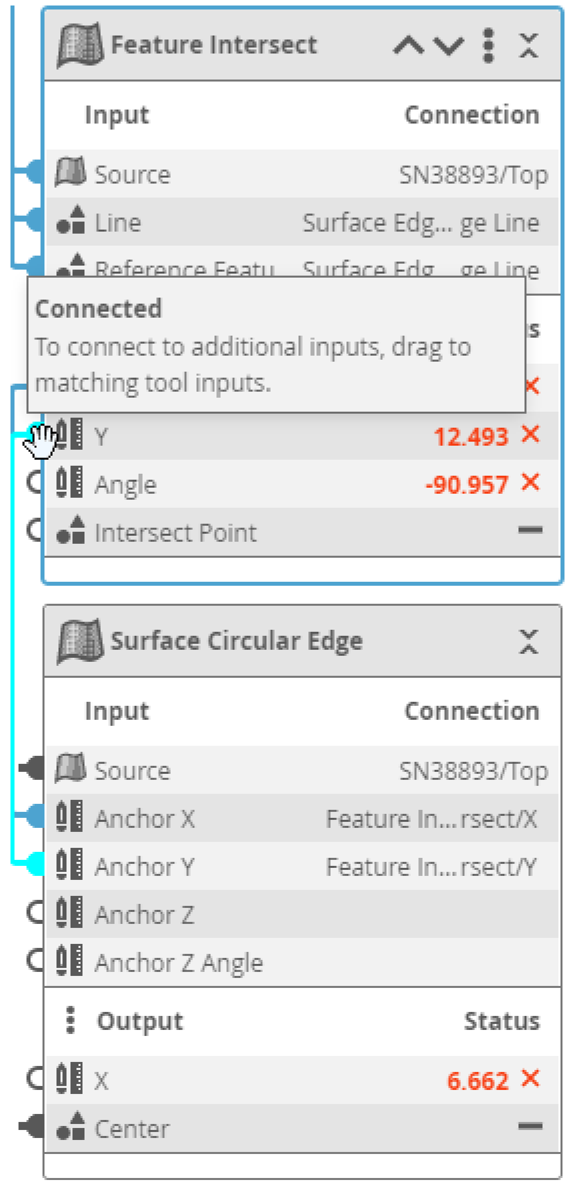
To see a complete list of a tool's outputs (as opposed to only the enabled ones), at the top of the tool's Output section, click the Output menu button (⋮). A pop-up list of all available outputs displays, indicating the enabled outputs with a checkmark.



When a tool is collapsed, you can "peek" the available inputs or the enabled outputs by clicking one of the horizontal lists of small icons (Standard view only).

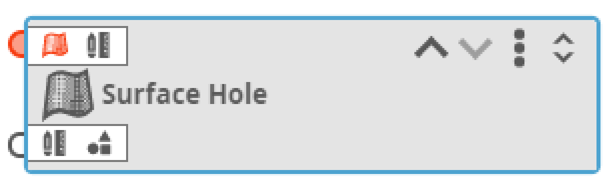


If you hover the mouse pointer over a blue connected node, a part of the blue connection line is highlighted to indicate what it is connected to. In the image below, you can see that by hovering over an output (the Y measurement of the Feature Intersect tool at the top) is used as an input (the Y anchor) of the Surface Circular Edge at the bottom.

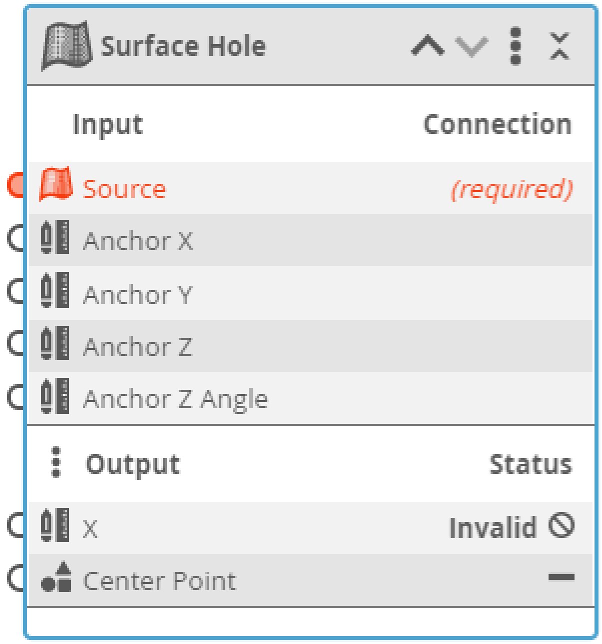


Script tools take no input in the Tools Diagram panel, as all outputs are available to these tools via their script functions.

If you remove a tool whose output is used by another tool as input, that input is displayed in red in the Tools Diagram panel to show that you must reconnect them.



Collapsed tool with a missing input



Expanded tool with a missing input

For information on connecting outputs to inputs, see See *Working with the Tools Diagram* on page 177.

Connecting Tools

The Tools Diagram panel lets you quickly connect tools using drag-and-drop operations.

	Displaying the connections (using the Display Connections button at the top of the panel) while connecting tools may be helpful.

In the following, we connect a geometric feature output from one tool to the input of another tool. However, the same procedure applies when connecting other kinds of outputs to inputs, such as connecting a measurement from one tool to one of the anchors available in another tool, or when connecting Surface output (such as the output from the Surface Filter tool) to the Source input of another tool (which is initially set to the direct output of a sensor).

To connect a tool's output to another tool's input:

1. Make sure you have added at least two tools and that you have configured the tools higher in the tool chain.

   The output you want to connect must be enabled in the first tool.

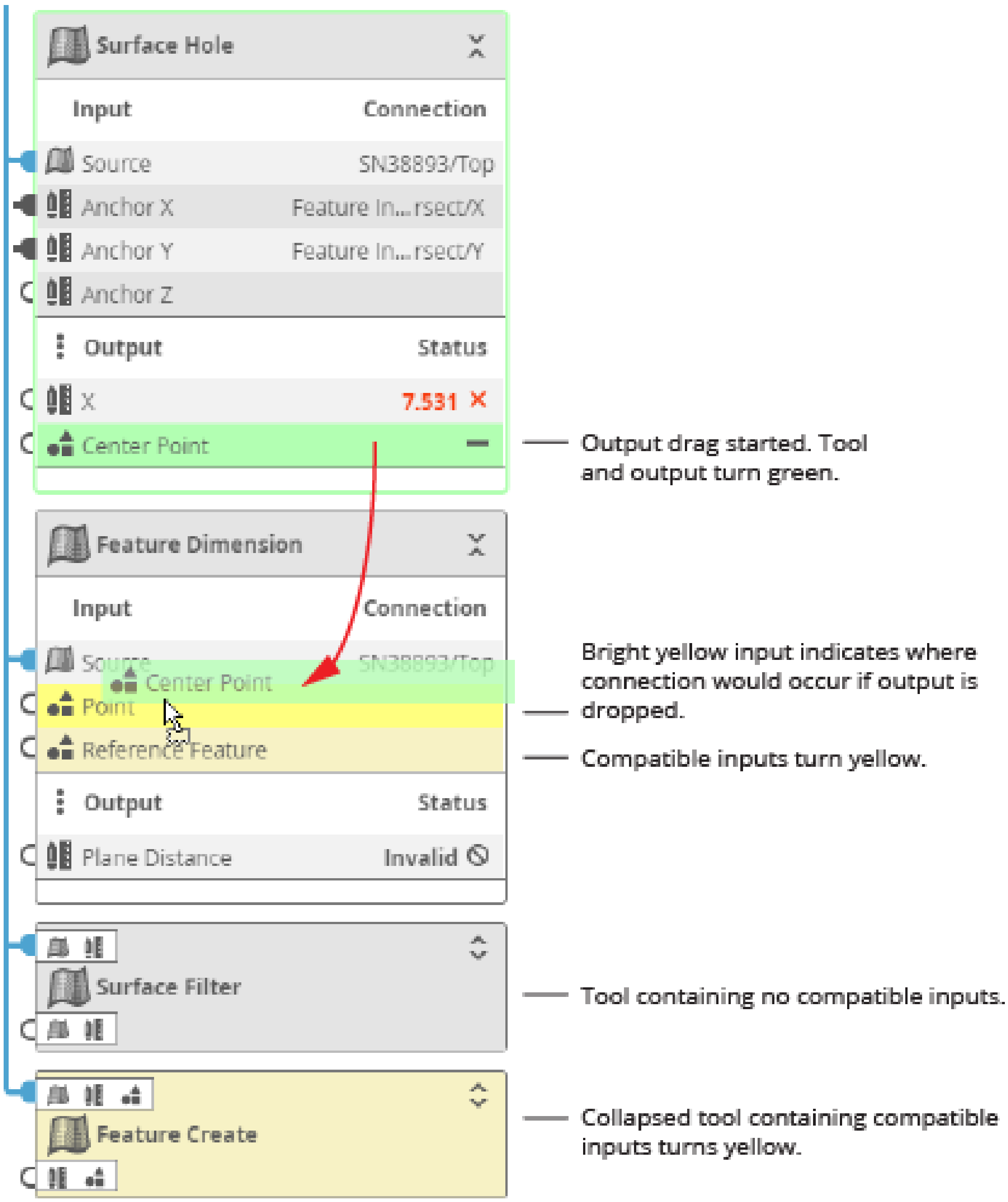
   For information on adding tools, see See *Working with the Tools Diagram* on page 177.

2. Locate the tool whose output you want to use (the "source" tool).

3. Do one of the following:

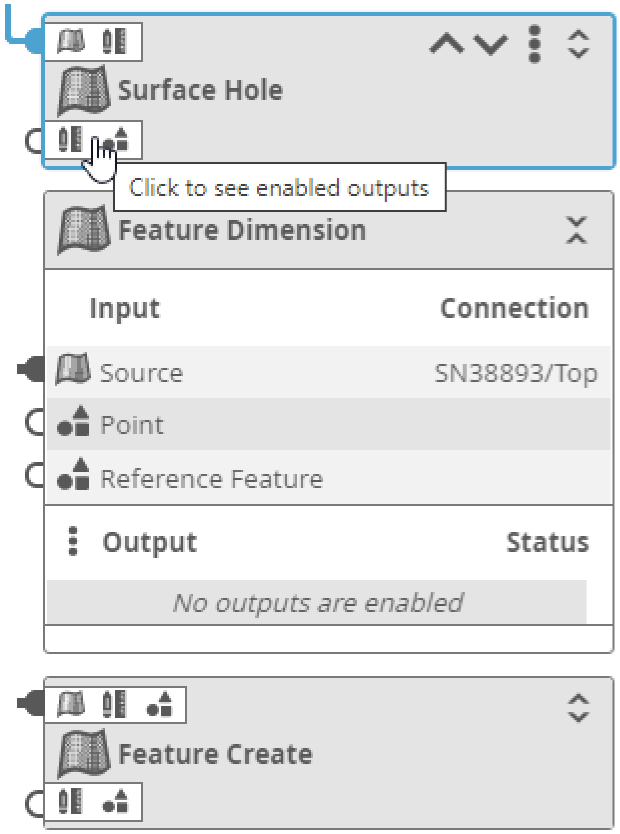
   With an expanded tool

   a. Click and hold the output you want to connect to the other tool's input and drag it to the input.

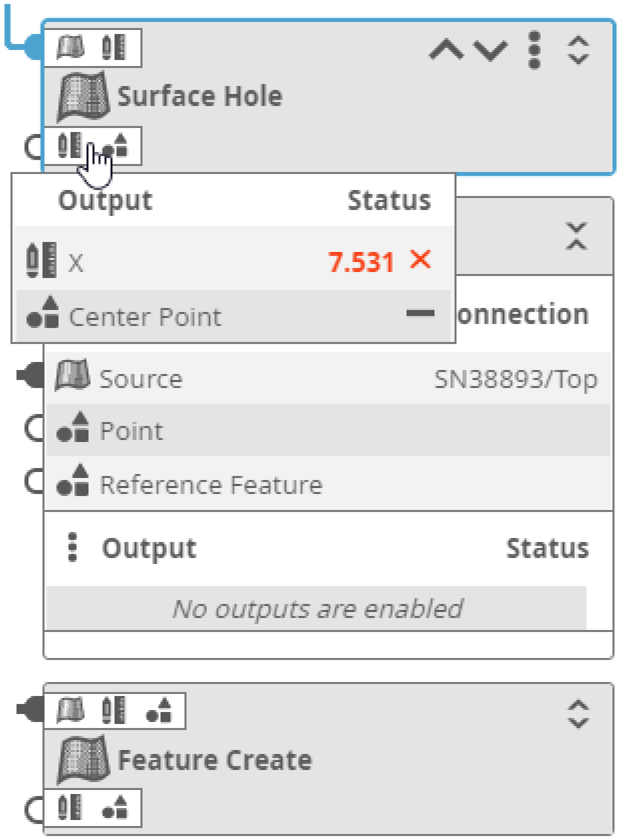


With a collapsed tool

a. Click the small output types at the bottom of the tool to expand the list of


Surface Hole
Click to see enabled outputs
Feature Dimension
Input
Connection
Source
SN38893/Top
Point
Reference Feature
Output
Status
No outputs are enabled
Feature Create

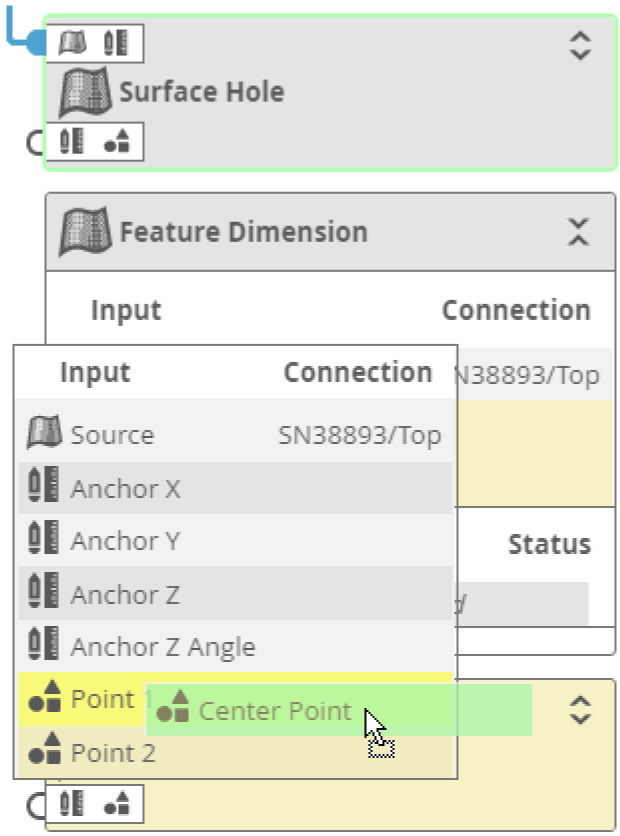
A list of enabled outputs is displayed in a pop-up list.



b. In the pop-up list, click and hold the output you want to connect to the other tool's input and drag it to the input.

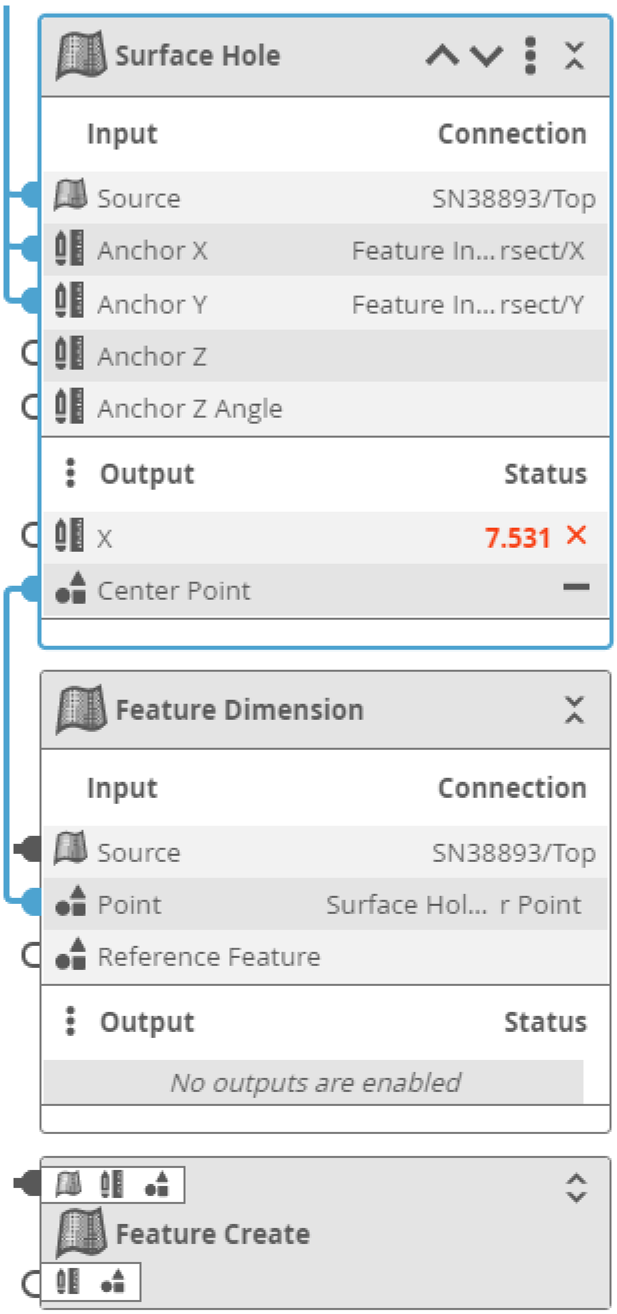
The source tool's border and the dragged output turn green. Compatible inputs turn yellow. The input to which the output will be linked if you drop it is highlighted in bright yellow; in the image above, this is the Point input.

Collapsed tools containing compatible inputs also turn yellow. If you move an output over a collapsed tool, a popup showing the tool's available inputs is displayed.

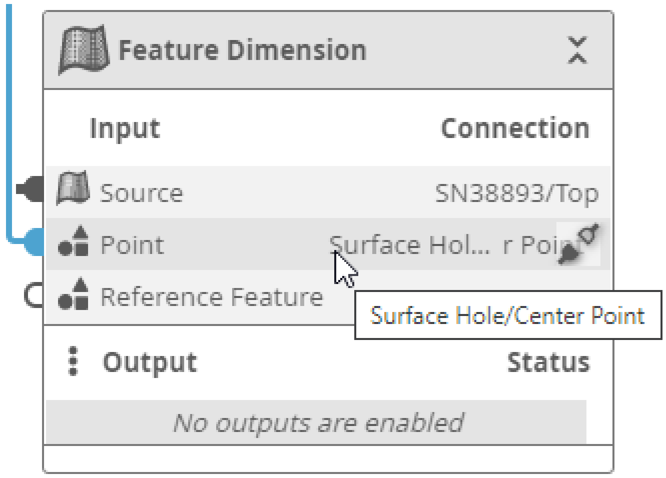


4. Drop the output on the desired input.

   A new connection appears between the first tool's output and the second tool's input (below, between the Surface Hole tool's Center Point output and the Point input in the Feature Dimension tool).



You can see the full name of an input or an output in a tooltip if you hover the mouse pointer over it.



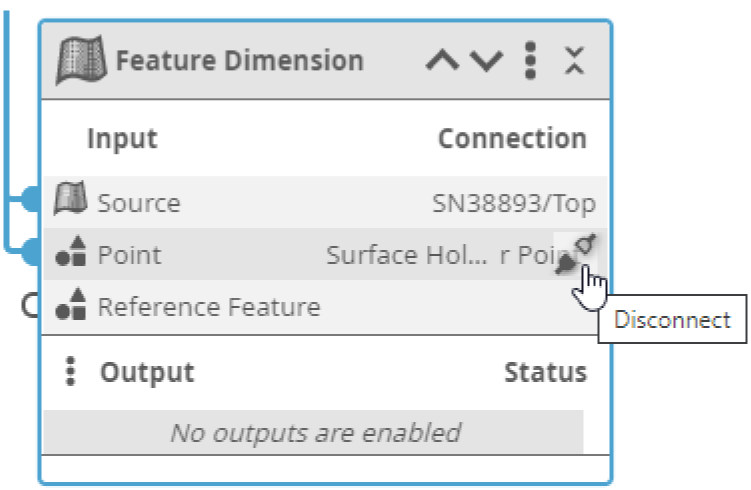
Disconnecting Tools

You can quickly disconnect an input in the Tools Diagram panel, but only if the tool containing the input is expanded.

To disconnect an input in a tool:

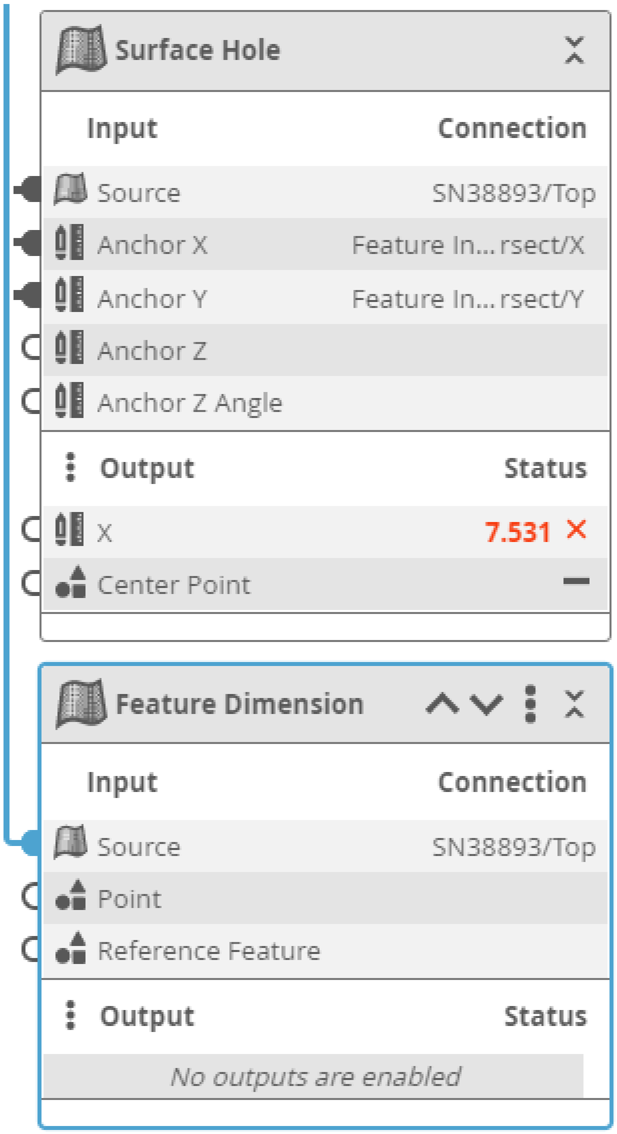
1. If the tool isn't expanded, click the Expand button at the top of the tool.

2. In the expanded tool, move the mouse pointer over the input you want to disconnect and move it to the right until the pointer is over the Disconnect icon.



3. Click the Disconnect icon.

   The input is disconnected from the other tool's output. (Below, the connection between Center Point and Point is remoeved.)

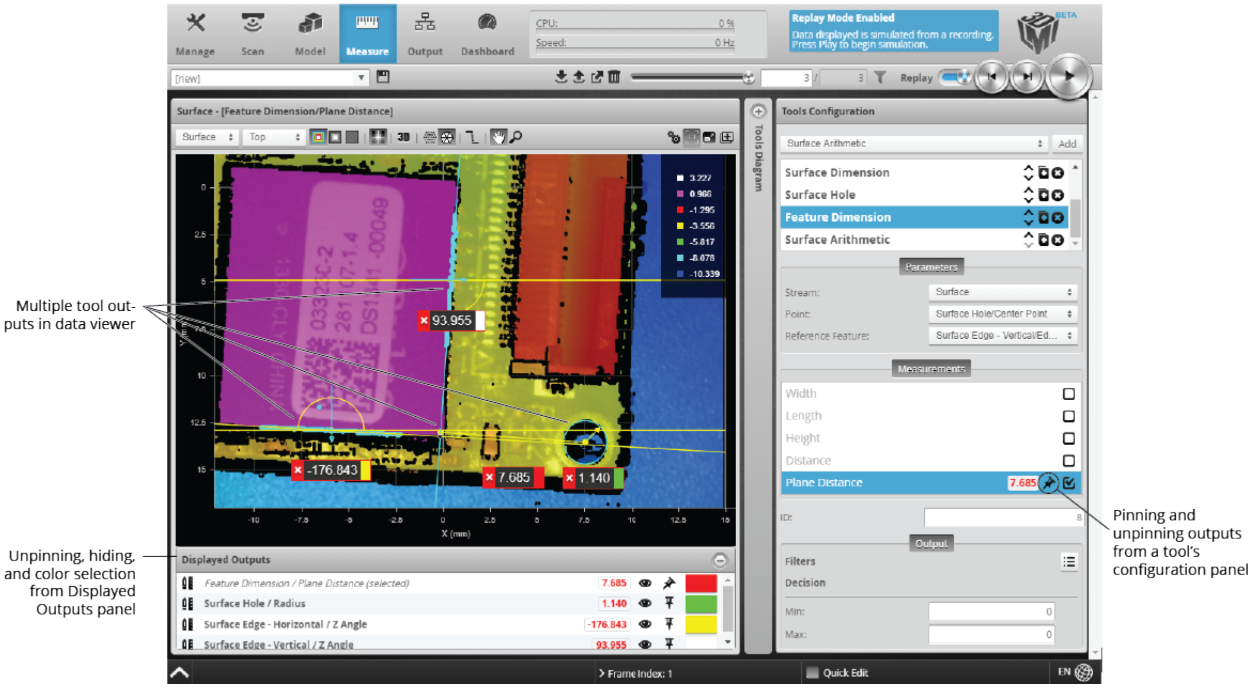


Pinning Measurements and Features

You can “pin” one or more tool outputs (measurements and geometric features) to a data viewer. When these outputs are pinned, they remain visible in a data viewer at all times, even when you click on a different tool, measurement, or feature in one of the lists the web interface displays. When no tool outputs are pinned, only the currently selected tool output is displayed in the data viewer. Pin information is stored in job files, so particular monitoring or configuration setups are automatically retrieved when you load a job containing pinned outputs.

Pinning outputs is useful if you want to monitor multiple, independent measurements while the Gocator is running in production. Pinning is also useful when setting up tools: you can change the parameters of a tool (such as a filter) earlier in a tool chain and immediately see the impact that modification has on another tool later in the chain. This minimizes toggling and clicking between tools and measurements. Pins are automatically stored as measurements in job files.

In the following image, a Feature Dimension Plane Distance measurement (measuring the distance between the corner of a CPU and a mounting hole) is currently selected. Three other measurement (Surface Edge Z Angle measurements on two sides of the CPU and a Surface Hole Radius measurement to the lower right) are pinned.



Data viewer showing the currently selected measurement and three pinned measurements.

You pin and unpin tool outputs from a tool's configuration panel (in the list to the right of the data viewer). You can also pin and unpin outputs on the Dashboard page (the procedure is very similar); however, pinned outputs in the Dashboard are not independent from those in the main data viewer. You *can* pin outputs independently when you have multiple data viewer windows open (for more information, see *Using Multiple Data Viewer Windows* on page 147).

You can unpin and hide outputs in the **Displayed Outputs** panel below the data viewer, and pin the currently selected output. You can also choose the color of the measurement value. The currently selected but unpinned output is indicated by "(selected)" in the panel's list, meaning it is automatically but temporarily added: it will be removed from the panel's list when you switch to another output.

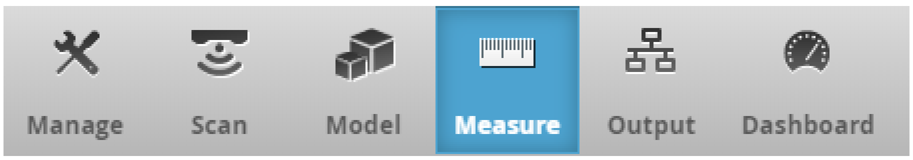
Tools (distinct from their outputs) with definable regions of interest can also appear in the list: this lets you temporarily hide the regions to reduce the visual elements in the data viewer. For example, in the following, the region definable in the Surface Hole tool is hidden, independently of the Surface Hole X *measurement*:

The naming convention for outputs in the **Displayed Outputs** panel is as follows:

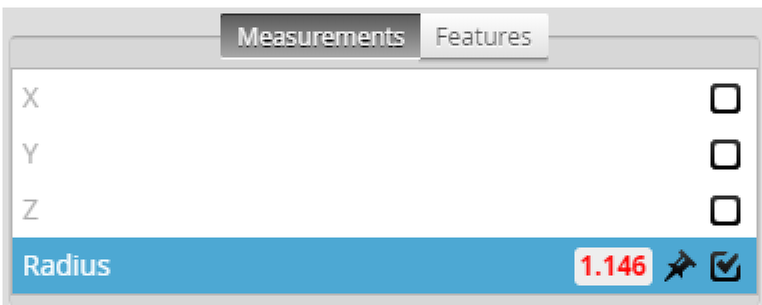
```
Tool_icon Tool_name / Measurement_name
```

To pin or unpin a tool output from a tool's configuration panel:

1. Go to the **Measure** page.



2. In a previously added and configured tool, go to the **Measurements** or **Features** tab.



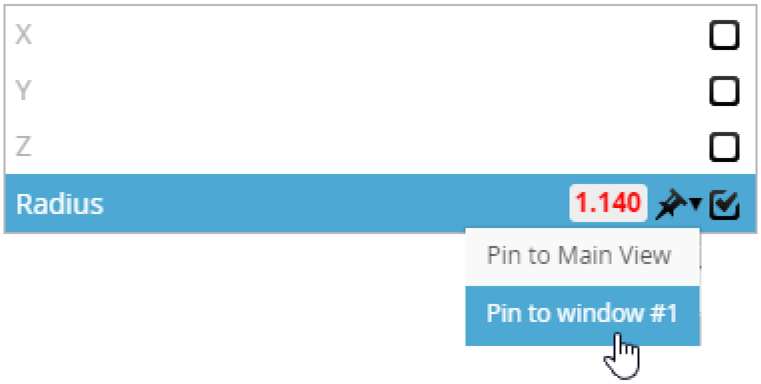
3. In the tab, locate the output you want to pin or unpin and do one of the following:

   Pin an output:

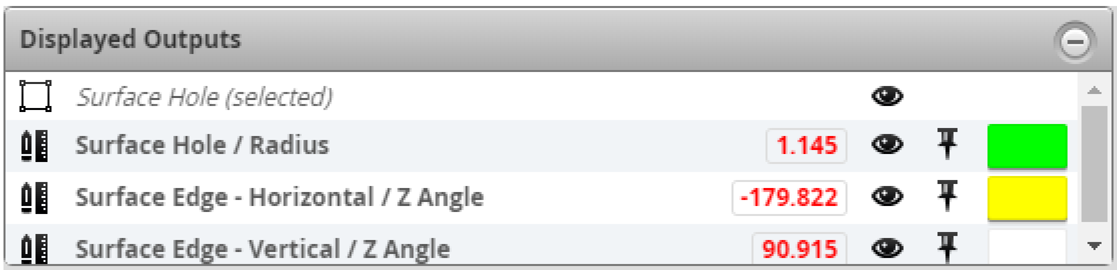
   a. If only the Main View data viewer is open, click the pin icon next to the output you want to pin.

   b. If you have opened additional data viewer windows, click the pin icon and choose the view to pin the output to from the drop-down.



The output is added to the list in the **Displayed Outputs** panel in the data viewer you chose and is pinned in that data viewer.

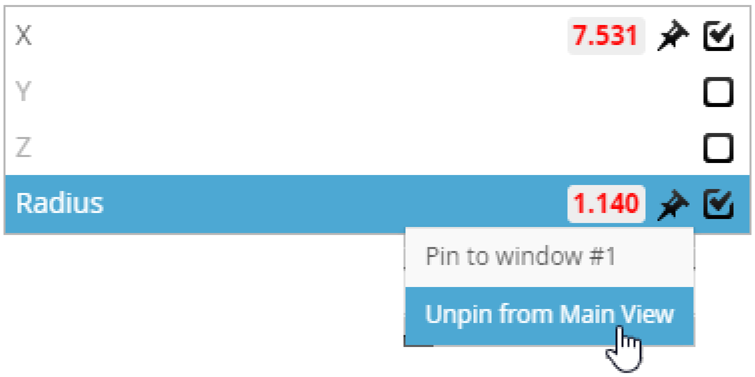


For more information on using multiple data viewer windows, see *Using Multiple Data Viewer Windows* on page 147.)

Unpin an output:

a. If only the Main View data viewer is open, click the pin icon next to the output you want to unpin.

If you have opened other data viewer windows, you choose which one from which to unpin the output. (For more information on using data viewer windows, see *Using Multiple Data Viewer Windows* on page 147.)

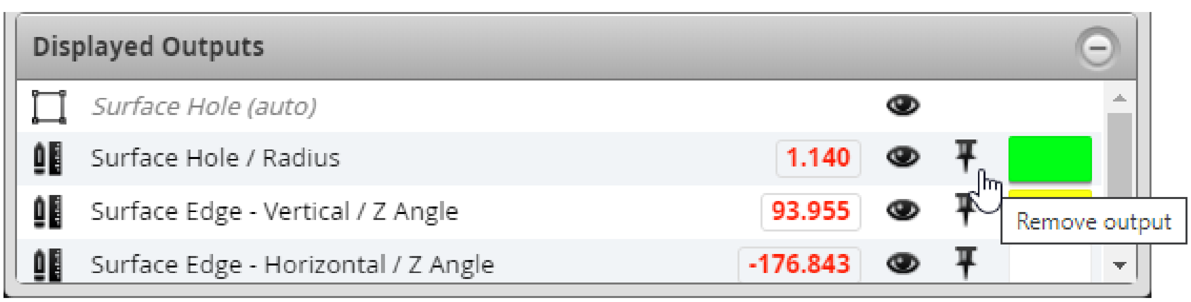


The output is removed from the **Displayed Outputs** panel and is no longer displayed in the data viewer, unless it is currently selected in a tool's list of outputs.

In the **Displayed Outputs** panel below a data viewer, you can also manage the pinned outputs of that data viewer, unpinning and hiding outputs, and choosing a measurement value's color.

To unpin an output in the Displayed Outputs panel:

- In the Displayed Outputs panel, click the pin next to the output you want to remove.



The output is removed from the list in the panel, and is no longer displayed in the data viewer, unless it is currently selected in a tool's configuration.

You can temporarily hide an output in a data viewer to make it easier to work with the data viewer. The state of outputs (shown vs. hidden) is *not* stored in the job file.

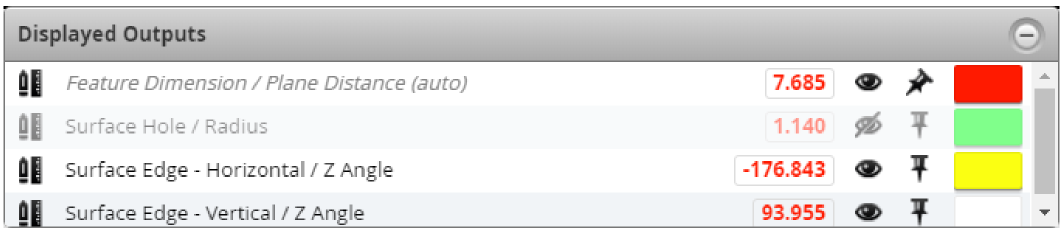
To hide or show an output in the Displayed Outputs panel:

1. In the Displayed Outputs panel, do one of the following:

Hide an output:

a. Click the eye icon ( ) of the output you want to hide

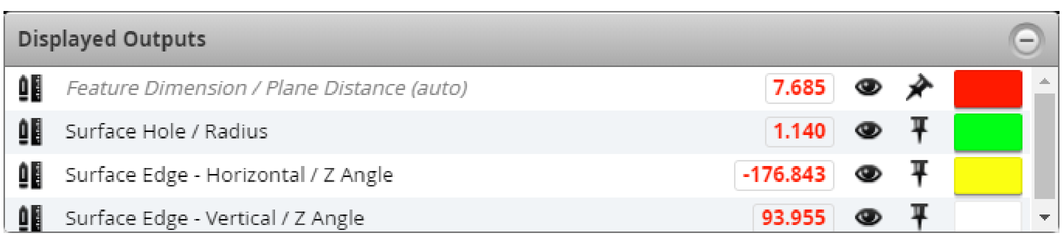
The output in the panel is greyed out and it is no longer displayed in the data viewer. The output is still pinned to the data viewer.



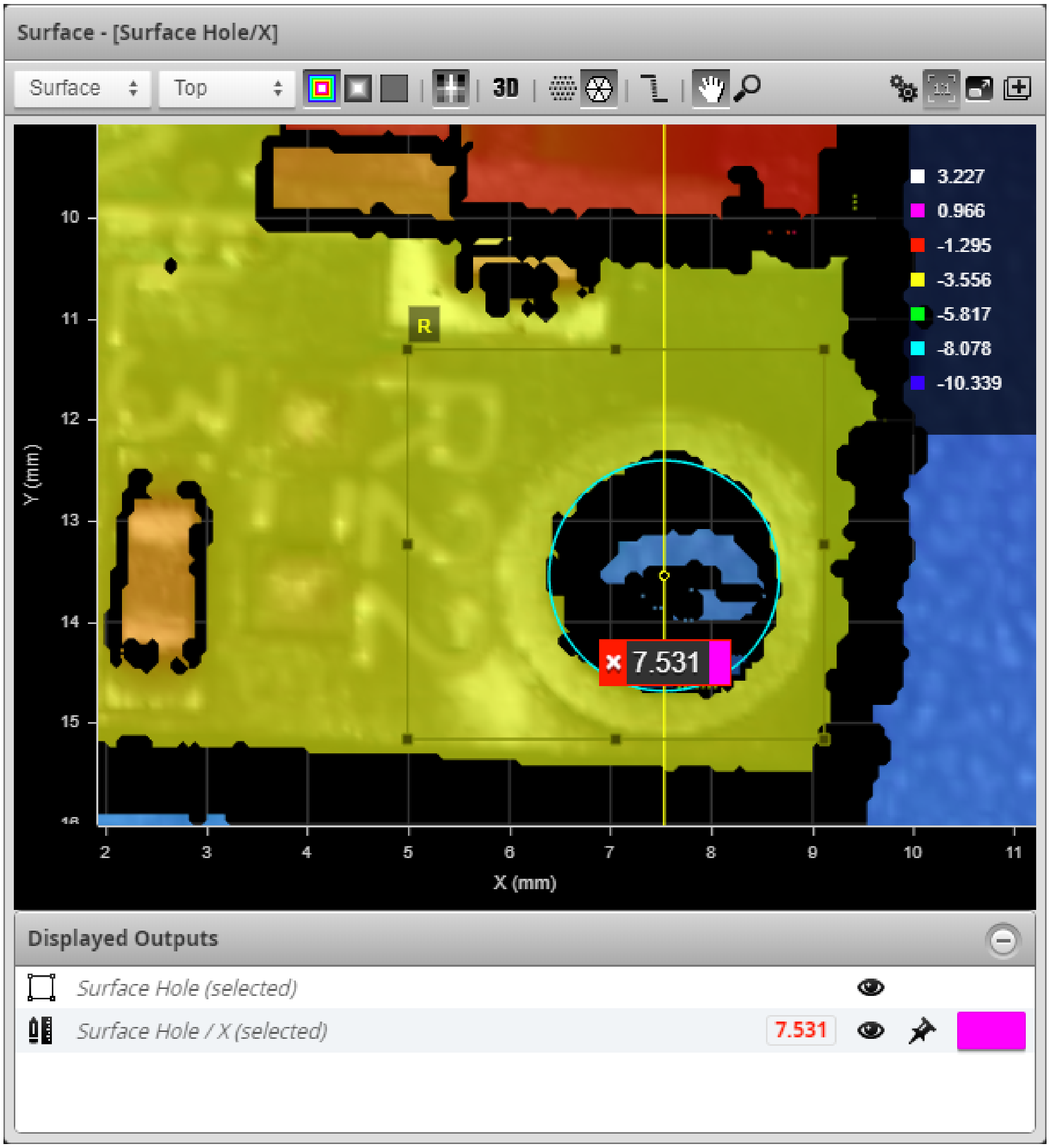
Show a hidden output:

a. Click the barred eye icon ( ) of the output you want to hide.

The output returns to the visible state.



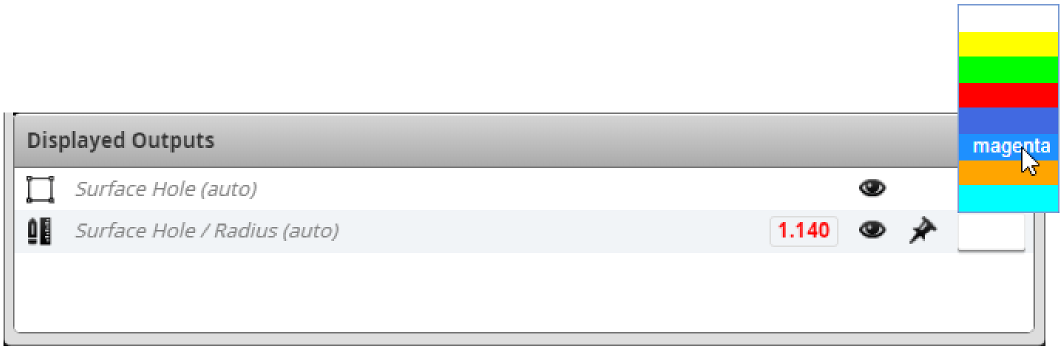
You can choose the color of the right vertical part of a measurement value that’s displayed in a data viewer. In the following image, the color associated with the Surface Hole X measurement value has been set to magenta:



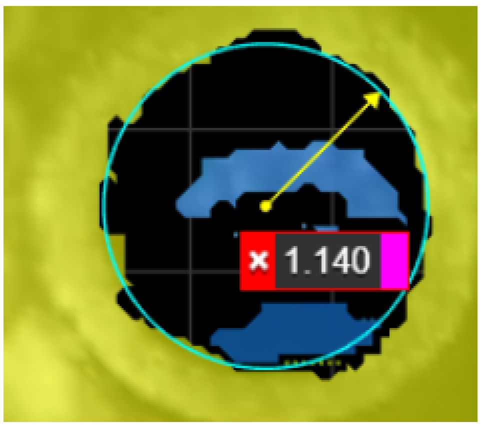
To change a measurement value's associated color:

1. In the Displayed Outputs panel, click one of the rectangles of color.

2. In the color picker, choose a color.



The color associated with a measurement value is changed.



Profile Measurement

This section describes the profile measurement tools available in Gocator sensors.

When Gocator is in Surface mode and you have defined a section, a **Stream** option displays in Profile tools. Choosing a section in the **Stream** option lets you apply profile measurements to the section.

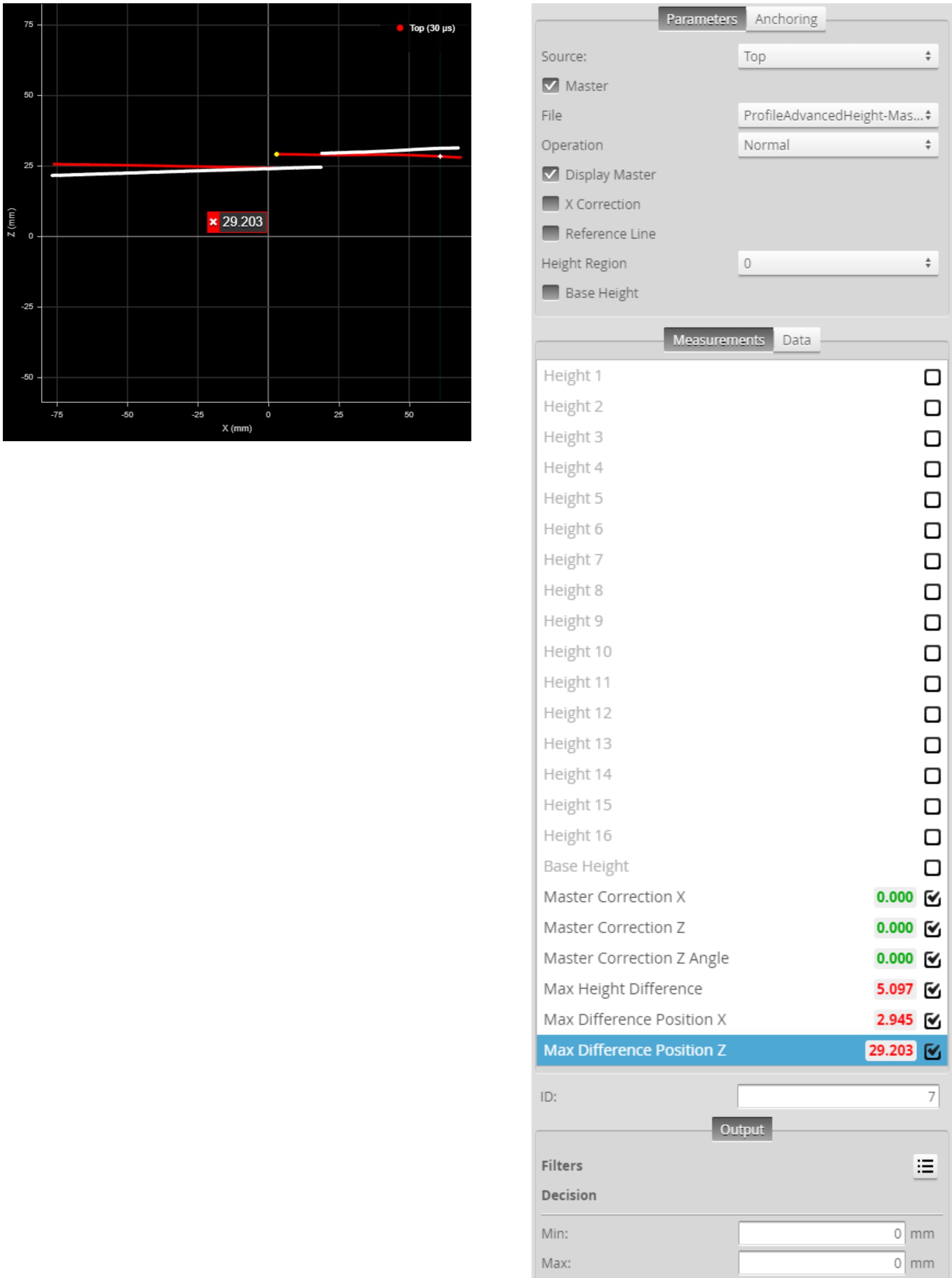
Profile measurement tools can be used on sections. For more information on sections, see *Sections* on page 140.

Advanced Height

The Advanced Height tool provides highly accurate and repeatable master (template) comparison and step height measurements (up to 16 in a tool instance).

🗋	All instances of the Advanced Height tool share the same template file set in **File**. For this reason, you must be careful when editing or removing template files shared by another instance of the tool.

Height measurements can be made relative to a reference line. Reference line sets the measurement direction (perpendicular to the reference line). A separate base line can also be set so that height measurements are between the base line and a profile feature, rather than the reference line (which in this case is used for angle correction).



Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Data, and Settings

Measurements

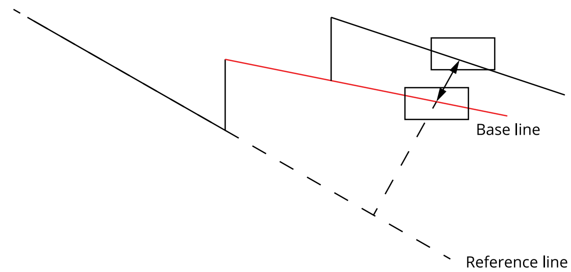
Measurement
Height {n} The height measured in height region {n}. Height is measured perpendicular Will be Invalid if the appropriate number of height regions has not been set in **Height Region**.
Master Correction X **Master Correction Z** **Master Correction Z Angle** The amount of correction applied to the profile with respect to the master.
Max Height Difference The maximum height difference.
Max Difference Position X **Max Difference Position Z** The X and Z positions of the maximum height difference.

Data

Type	Description
Difference Profile	A profile representing the difference between the master and the current frame's profile, available for use as input in the **Stream** drop-down in other tools.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Master	Toggles a set of settings related to master comparison. For more information, see *Master Comparison* on the next page.
Reference Line	Toggles a set of settings related to the reference line. For more information, see *Reference Line* on page 210.
Height Region	Sets the number of height region measurements the tools returns. For each height region, the tool displays an **Edit Height Region** checkbox that you use to edit the height region's location and size. The tool also displays a **Height{n} Feature** drop-down that lets you select the type of feature for that height region.

Parameter	Description
Base Height	Use base height to "set" the Z axis: when enabled height values are offset from the base. This is useful if you need to measure between two features, rather than between a feature and the reference line.  When enabled, the tool displays settings related to the base height: size and position of the base height's region (**Base Height** section) and the base height's feature.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Master Comparison

When you check the **Master** option, the tool displays several additional settings and disables measurement anchoring from other tools.

Master Parameters

Parameter	Description
File	The file containing the master (template) profile, created by choosing **Save** from the **Operation** drop-down.
Operation	Contains operations related to the master file. One of the following: • **Normal**: Selected by the tool after you perform another file operation. • **Create**: Saves the *current profile* as the master. • **Delete**: Deletes the master file selected in **File**.
Display Master	Overlays the master profile, in white, on the current profile.
X Correction	Enables settings related to X correction (left or right movement) of the profile compared to the master profile. For more information, see *X Correction* on the next page.

X Correction

When you check the **Master** option and enable **X Correction**, the tool displays several additional settings.

X Correction Parameters

Parameter	Description
Edit Edge Region	Enables an edge region section letting you configure the region. You can also edit this region in the data viewer.
Edge Direction	Determines the direction of the edge. One of the following: **Falling** or **Rising**.
Count Direction	Indicates how edges are counted. One of the following: **Left to Right** or **Right to Left**.
Edge Index	Indicates which edge the tool uses.

Reference Line

When you check the **Master** option and enable **Reference Line**, the tool displays several additional settings. The reference line is used to set the measurement direction (perpendicular to the reference line).

Reference Line Parameters

Parameter	Description
Line Region	The number of line regions the tool uses.
Edit Line Region	Enables settings that let you edit the size and position of the line's region.
Fitting Method	Indicates the fitting method the tool uses. One of the following: **Simple** or **Robust**.

Anchoring

Anchoring

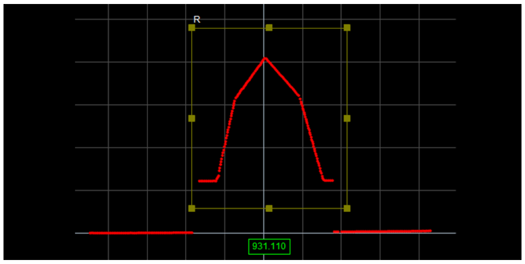
Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

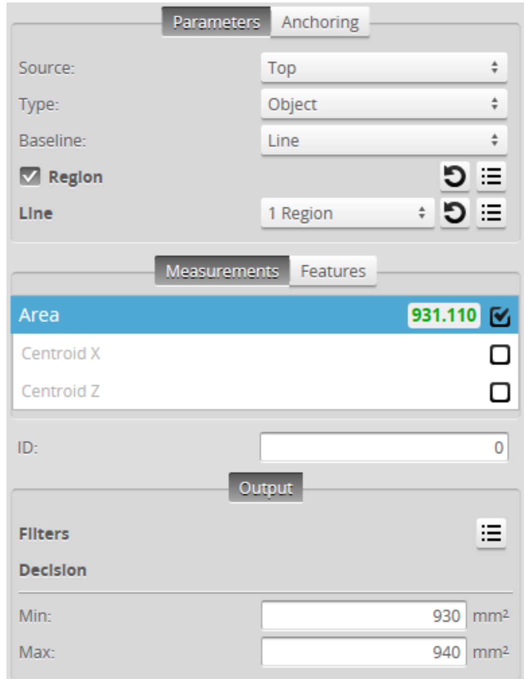
> A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

> For more information on anchoring, see *Measurement Anchoring* on page 169.

Area

The Area tool determines the cross-sectional area within a region.



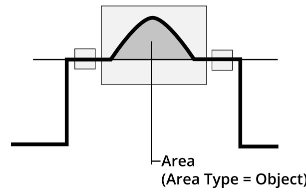
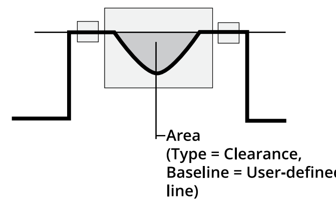
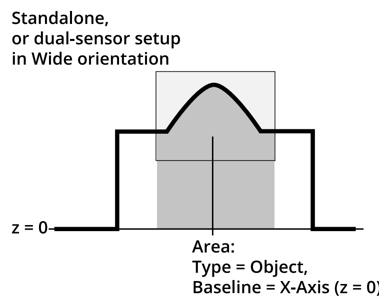
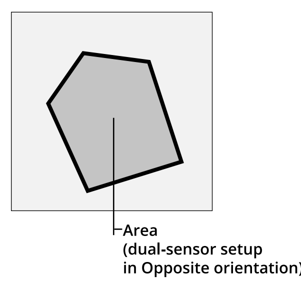
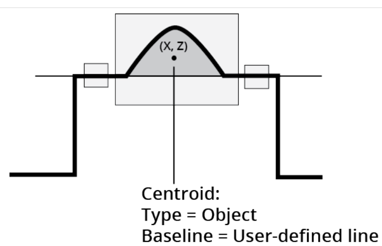


Areas are positive in regions where the profile is above the X axis. In contrast, areas are negative in regions where the profile is below the X axis.

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
Area Measures the cross-sectional area within a region that is above or below a fitted baseline.	   
Centroid X Determines the X position of the centroid of the area.	
Centroid Z Determines the Z position of the centroid of the area.	

Features

Type	Description
Center Point	The center point of the area.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
Type	**Object** area type is for convex shapes above the baseline. Regions below the baseline are ignored. **Clearance** area type is for concave shapes below the baseline. Regions above the baseline are ignored.
Baseline	Baseline is the fit line that represents the line above which (Object clearance type) or below which (Clearance area type) the cross-sectional area is measured. When this parameter is set to **Line**, you must define a line in the Line parameter. See *Fit Lines* on page 166 for more information on fit lines. When this parameter is set to **X-Axis**, the baseline is set to z = 0.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Line	When **Baseline** (see above) is set to **Line**, set this to one of the following: **1 Region** or **2 Regions**: Lets you set one or two regions whose data the tool will use to fit a line. **All Data**: The tool uses all of the data in the active area. For more information on regions, see *Regions* on page 152). For more information on fit lines, see *Fit Lines* on page 166.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

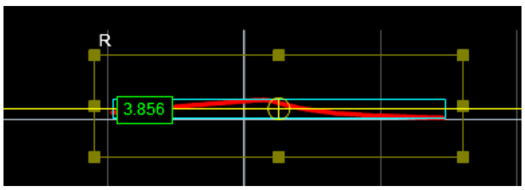
For more information on anchoring, see *Measurement Anchoring* on page 169.

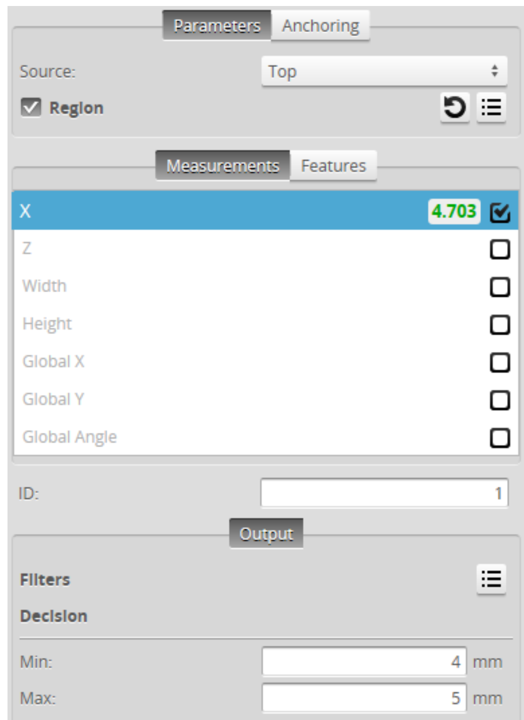
Bounding Box

The Bounding Box tool provides measurements related to the smallest box that contains the profile (for example, X position, Z position, width, etc.).

The bounding box provides the absolute position from which the Position centroids tools are referenced.

> When you use measurement tools on parts or sections, the coordinates returned are relative to the part or section. You can use the values returned by the Bounding Box tool's "Global" (see below) measurements as an offset in a Gocator script to convert the positional (X, Y, or Z) measurements of other measurement tools to sensor or system coordinates (depending on whether the sensor is aligned). For more information on Gocator scripts, see *Scripts* on page 505.



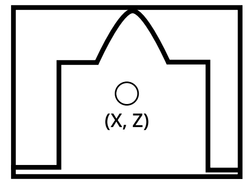
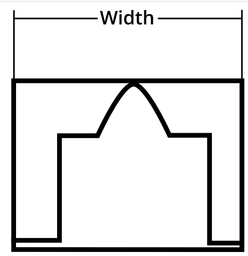
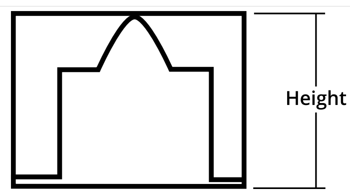


Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Determines the X position of the center of the bounding box that contains the profile. The value returned is relative to the profile.	
Z Determines the Z position of the center of the bounding box that contains the profile. The value returned is relative to the profile.	
Width Determines the width of the bounding box that contains the profile. The width reports the dimension of the box in the direction of the minor axis.	
Height Determines the height (thickness) of the bounding box that contains the profile.	
Global X* Determines the X position of the center of the bounding box that contains the profile relative to the surface from which the profile is extracted.	
Global Y* Determines the Y position of the center of the bounding box that contains the profile relative to the surface from which the profile is extracted.	
Global Angle* Determines the angle around Z of the section used to create the profile, relative to the surface from which it is extracted, where a line parallel to the X axis is 0 degrees. Angles of sections pointing to the bottom of the data viewer are positive. Angles of sections pointing to the top of the data viewer are negative.	

*The Global X, Global Y, and Global Angle measurements are intended to be used with profiles extracted from a surface using a section.

When used with profiles not generated from a section, the Global X measurement returns the same value as the X measurement, and the Global Y and Global Angle measurements return 0.000.

Features

Type	Description
Center Point	The center point of the bounding box.
Corner Point	The lower left corner of the bounding box.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

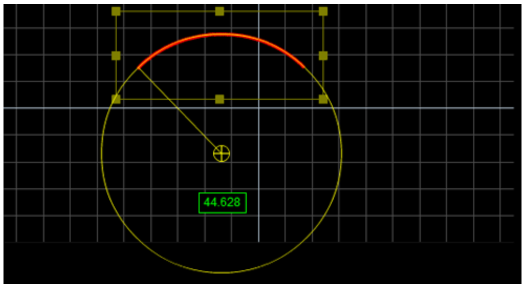
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

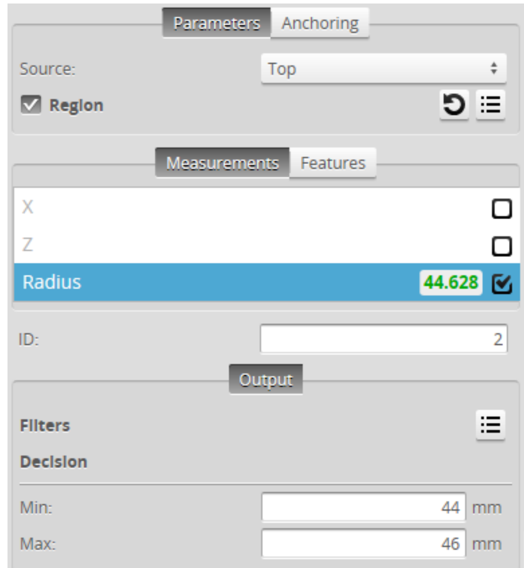
For more information on anchoring, see *Measurement Anchoring* on page 169.

Circle

The Circle tool provides measurements that find the best-fitted circle to a profile and measure various characteristics of the circle.

> The tool may be unable to fit a circle to the profile when attempting the fit on a small number of relatively collinear data points.

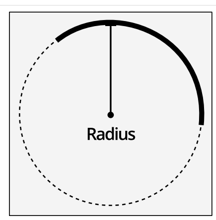
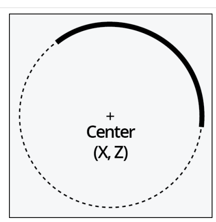




For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
Radius Measures the radius of the circle.	
X Finds the circle center position in the X axis.	
Z Finds the circle center position in the Z axis.	

Measurement	Illustration
Standard Deviation Returns the standard deviation of the data points with respect to the fitted circle.	
Min Error **Max Error** The minimum and maximum error among the data points with respect to the fitted circle.	
Min Error X **Min Error Z** The X and Z position of the minimum error.	
Max Error X **Max Error Z** The X and Z position of the maximum error.	

Features

Type	Description
Center Point	The center point of the fitted circle.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

	A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

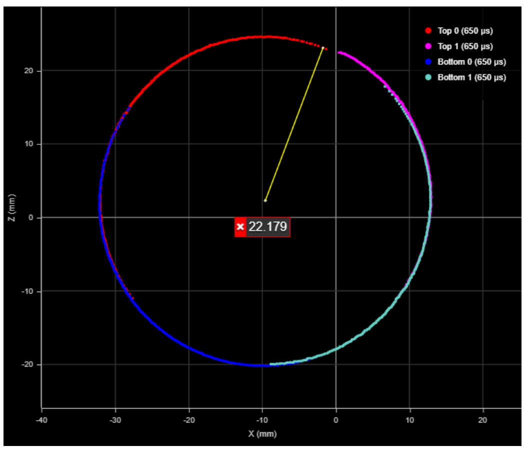
	For more information on anchoring, see *Measurement Anchoring* on page 169.

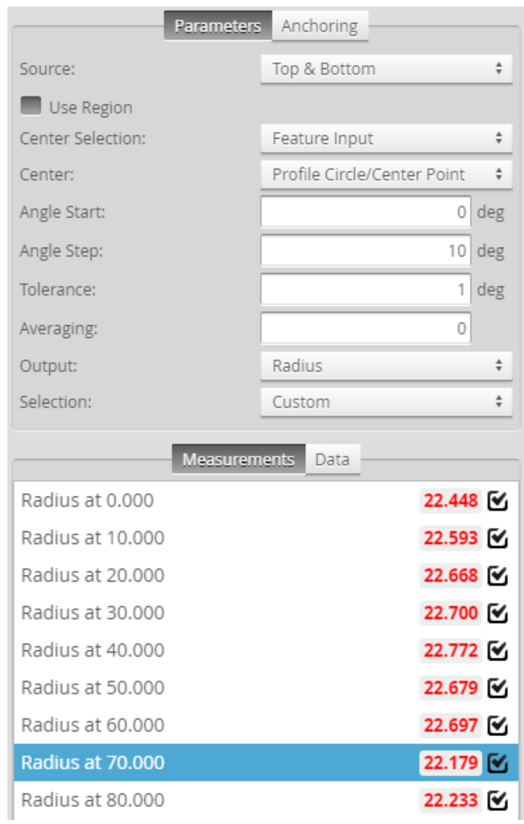
Circle Radii

	This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Profile Circle Radii tool lets you measure radii and diameters at specified angle steps, given a specified center point. The tool draws rays from the center point and returns radii or diameter measurements for each ray.

For example, in the following scan of an exhaust pipe by a four-sensor system, the tool is showing a radius measurement at 70 degrees that indicates a dent in the pipe. The tool also provides settings to compensate for missing data and for rough surfaces or noise.

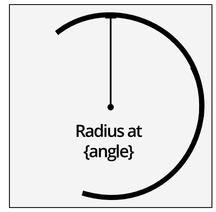
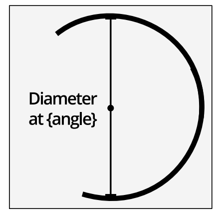




For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

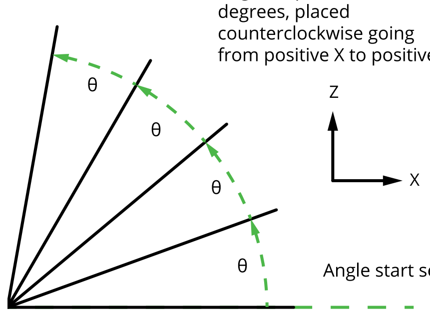
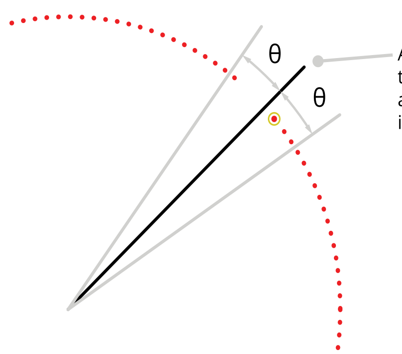
Measurement	Illustration
Radius at {angle} Returns the radius at {angle}.	
Diameter at {angle} Returns the diameter at {angle}.	

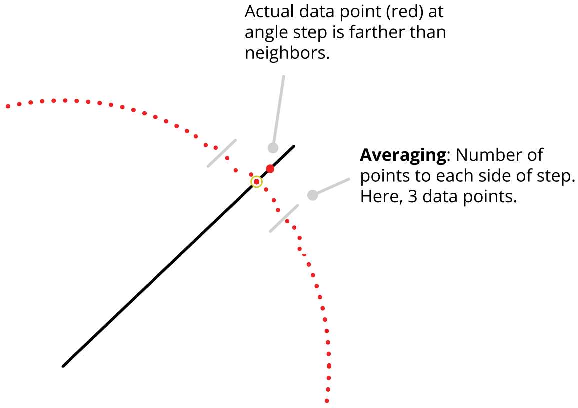
Data

Type	Description
Points	An array of the points at the end of the rays.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Use Region	Indicates whether the tool uses a user-defined region. If this option is not checked, the tool uses data from the entire active area.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Center Selection	The source for the point geometric feature the tool uses as a center point. One of the following: **Bounding Box** – Uses the center of the bounding box that encloses the scan data selected in **Source**. If **Use Region** is enabled, the tool places a bounding box only around the data in the region. If **Use Region** is disabled, the tool places a bounding box around all scan data; this will include any outliers in the bounding box, which could produce an undesired center point. **Feature Input** – A point geometric feature provided by another tool, such as the center point from a Circle tool.
Center	The point geometric feature coming from another tool that the Circle Radii tool uses as the center point from which rays are drawn to search for data points. The parameter is only available when **Center Selection** is set to **Feature Input**.

Parameter	Description
Angle Start Angle Step	**Angle Start**: The angle at which ray steps start. **Angle Step**: The angle step in degrees. The following shows how these settings work together:  The tool searches for a data point at each angle step and returns the radius from the center point or the diameter.
Tolerance	If no data point is found at the angle step, the tool searches within the specified number of degrees to each side of the step to find a data point. Useful to compensate for gaps in the data.  The graphic above shows how the tool searches to each side of the angle step until it finds a data point (circled and in yellow).

Parameter	Description
Averaging	The number of data points to each side of the point the tool uses to average. Use this to compensate for noise or rough surfaces.  The graphic above shows how the tool averages the data point at the angle step with the number of data points specified in **Averaging** to each side of the angle step, replacing the original data point with the average (circled and in yellow).
Output	Selects whether to output radius, diameter, or both at each step.
Selection	Lets you quickly enable or disable all measurements.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

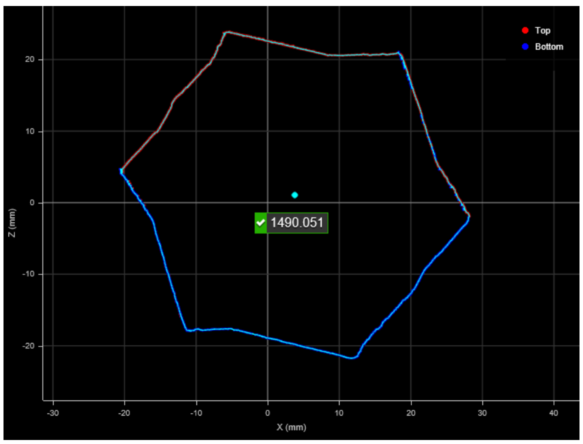
For more information on anchoring, see *Measurement Anchoring* on page 169.

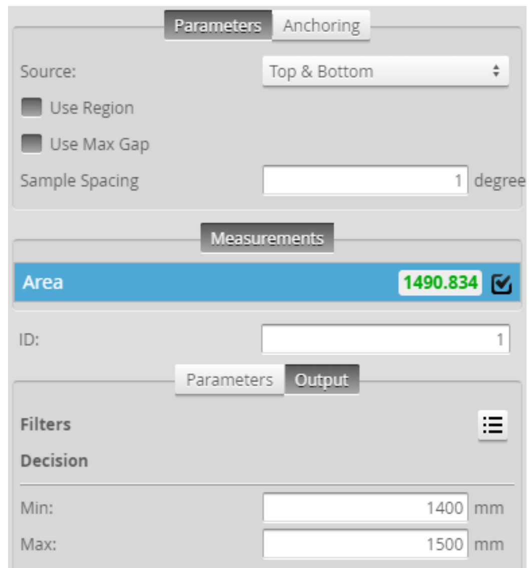
Closed Area

The Closed Area tool determines the cross-sectional area within a region using point cloud data from a dual- or multi-sensor system.

The tool is intended for use with roughly circular shaped profiles, or profiles that do not contain excessive concavity. The tool renders a polygon corresponding to the profile in the data viewer. Use this polygon to decide whether the tool can correctly calculate an acceptable representation of the profile. Minor gaps in the profile are permitted; the size of these gaps is configurable.

When the tool is used in conjunction with a script tool, you can calculate the volume of a target; for more information on the Script tool, see *Script* on page 275.

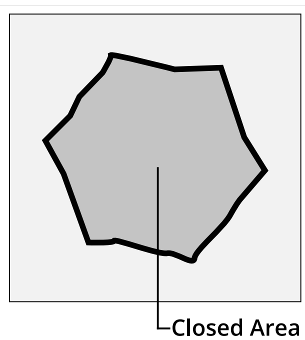




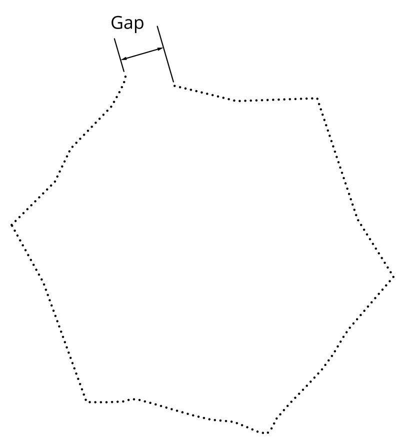
For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

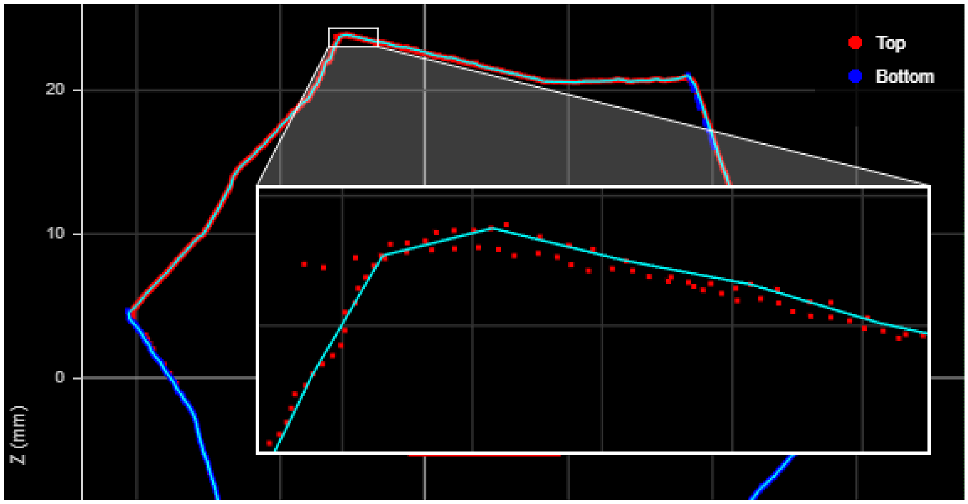
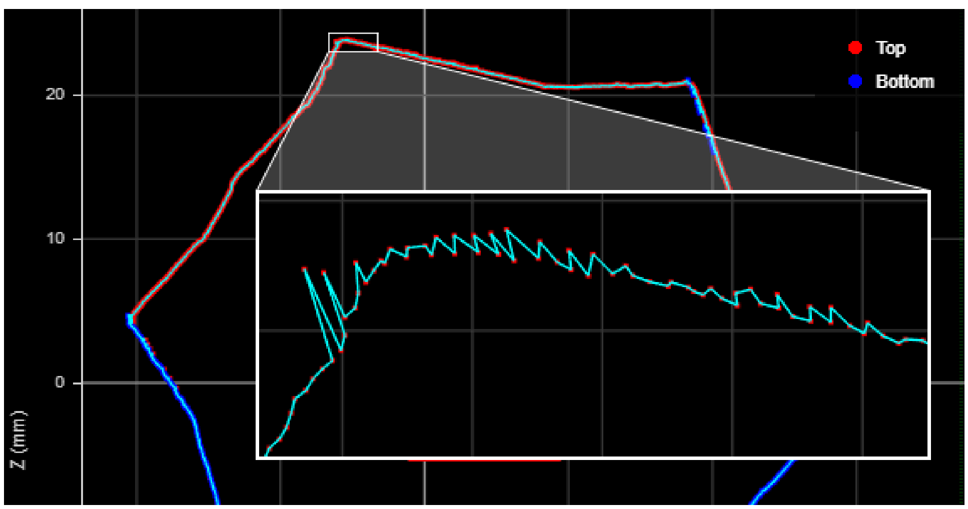
Measurements and Settings

Measurements

Measurement	Illustration
Closed Area Measures the cross-sectional area within a region using data from a dual- or multi-sensor system.	

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151. For this tool, you should set this parameter to **Top and Bottom**.
Use Region	Indicates whether the tool uses a user-defined region. If this option is not checked, the tool uses data from the entire active area.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Center Selection	The origin of the rays used to create the polygon (which in turn is used to calculate the area). One of the following: **Bounding Box** (default) Sets the center to the center of a bounding box that contains the tool data or the data in the region. **Feature Input** Lets you set the center to a point geometric feature output from another tool. When you choose this option, a **Center** dropdown lets you choose the center point. For more information on geometric features, see *Geometric Features* on page 164.
Use Max Gap	Indicates whether the tool uses the **Max Gap** setting (see below).
Max Gap	The maximum gap allowed between any two profile points on the contour of the target, in millimeters. In the following illustration of a profile, if the gap were greater than the value set in **Max Gap**, the tool would return an invalid value. 

Parameter	Description
Sample Spacing	The angle interval around the center of the profile the tool uses to calculate area. Enabling this setting and setting a value can increase the tool's performance. In the following image, the spacing is set to 1 degree. The polygon calculated from the profile points, which is then used to calculate the area, is simplified, increasing performance but reducing accuracy.  In the following image, **Sampling Spacing** is set to 0. Accuracy is increased, but performance is reduced.  If you set the value to 0, the tool uses the smallest angle permitted internally by the sampling engine.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

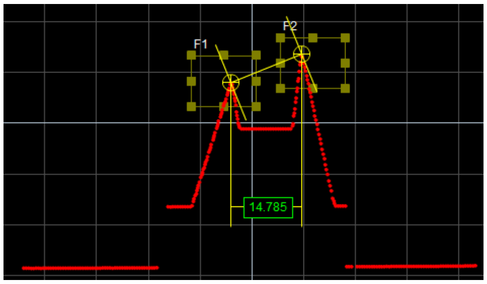
Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

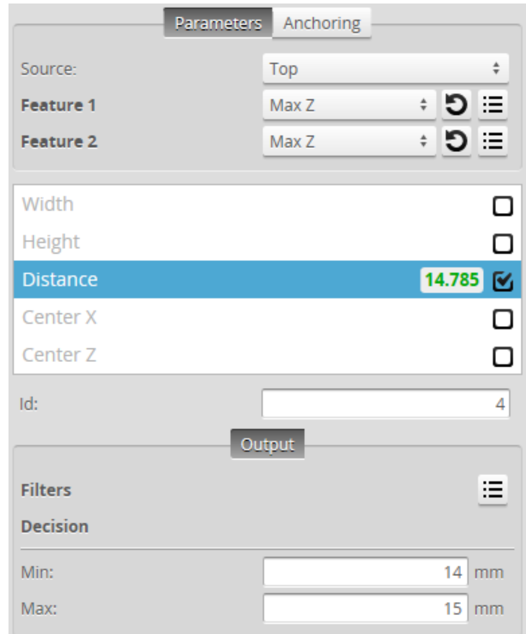
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Dimension

The Dimension tool provides Width, Height, Distance, Center X, and Center Z measurements.

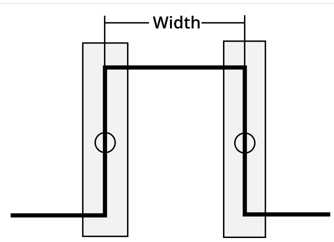
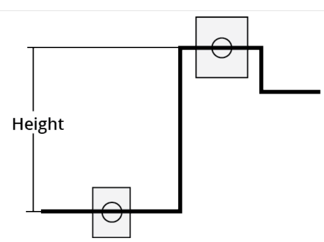


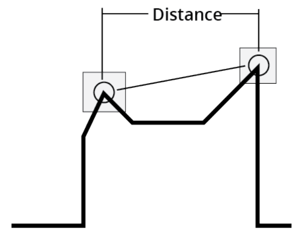
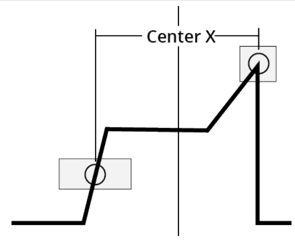
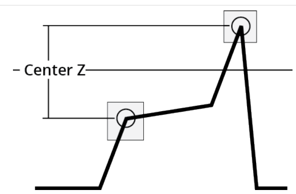


For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements and Settings

Measurements

Measurement	Illustration
Width Determines the difference along the X axis between two feature points. The difference can be calculated as an absolute or signed result. The difference is calculated by: *Width = Feature 2$_{X\ position}$ − Feature 1$_{X\ position}$*	
Height Determines the difference along the Z axis between two feature points. The difference can be expressed as an absolute or signed result. The difference is calculated by: *Height = Feature 2$_{Z\ position}$ − Feature 1$_{Z\ position}$*	

Measurement	Illustration
Distance Determines the direct, Euclidean distance between two feature points.	
Center X Finds the average location of two features and measures the X axis position of the average location	
Center Z Finds the average location of two features and measures the Z axis position of the average location.	

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.

Parameter	Description
Feature 1 Feature 2	The **Feature 1** and **Feature 2** settings represent the two features the tool uses to perform measurements. For each, one of the following: • Max Z • Min Z • Max X • Min X • Corner • Average • Rising Edge • Falling Edge • Any Edge • Top Corner • Bottom Corner • Left Corner • Right Corner • Median To set the region of a feature, adjust it graphically in the data viewer, or expand the feature using the expand button (☰) and enter the values in the fields. For more information on regions, see *Regions* on page 152.
Absolute *(Width and Height measurements only)*	Determines if the result will be expressed as an absolute or a signed value.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

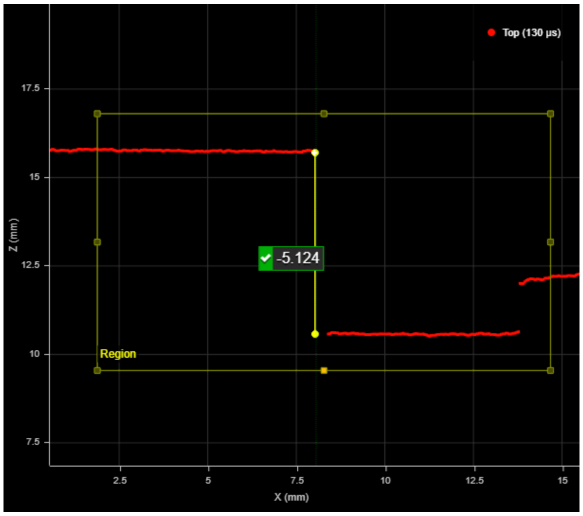
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

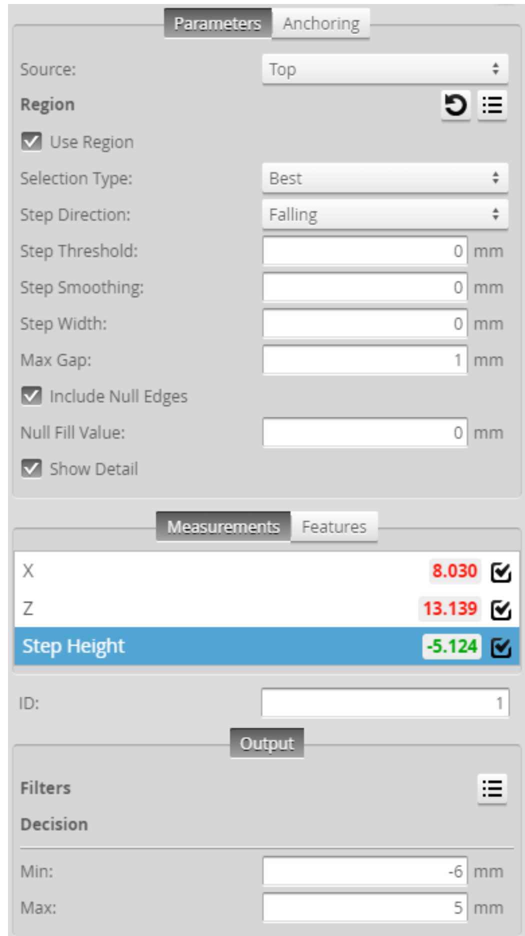
For more information on anchoring, see *Measurement Anchoring* on page 169.

Edge

The Profile Edge tool finds an edge on a profile, searching from left to right. The tool's settings help fit the edge point when multiple potential edges are in the region of interest. After the tool locates an edge, the position (X and Z) of the center of the step and the step height can be returned as measurements.

The tool can also generate a point geometric feature corresponding to the center of the step that Feature tools can take as input for measurement. For more information on Feature tools, see *Feature Measurement* on page 484.





For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement
X **Z** These measurements return the X and Z position of the edge point, respectively. The edge point is located half-way between the upper and lower data points of the step.
Step Height Returns the height of the step on the profile.

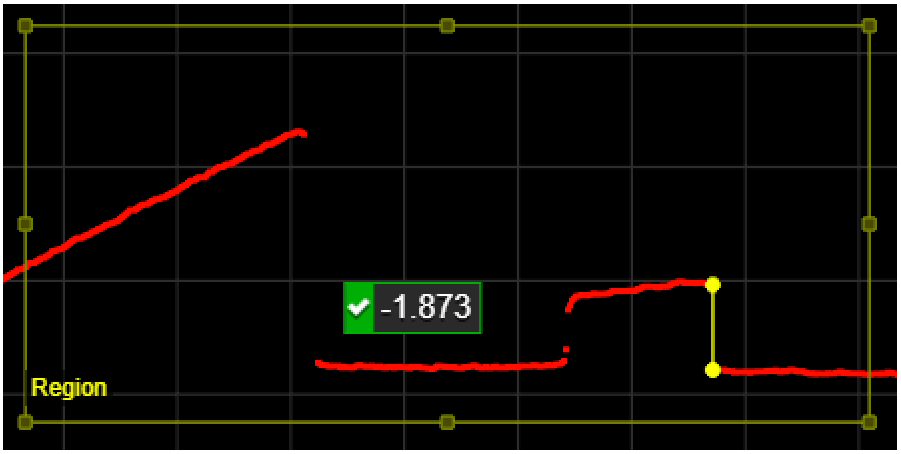
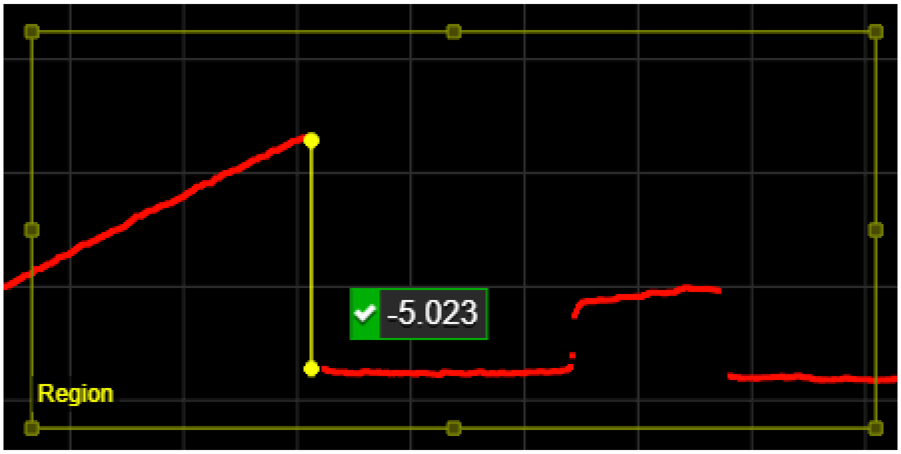
Features

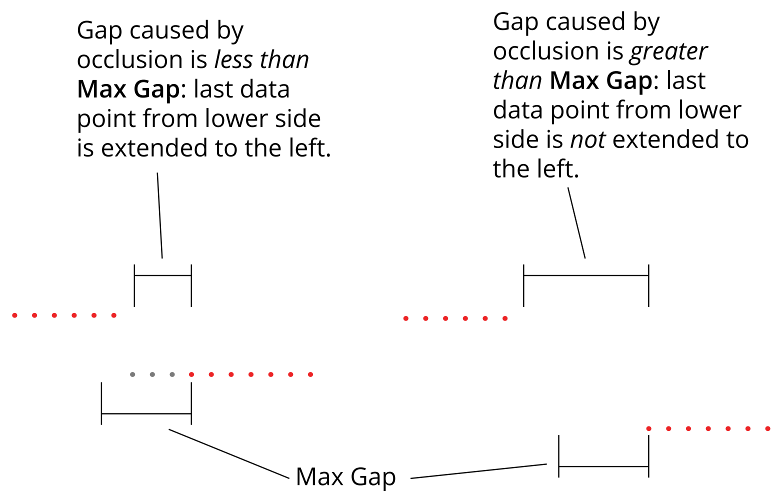
Type	Description
Edge Center Point	The edge point.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Selection Type	Determines which step the tool uses when there are multiple steps in the profile. An edge point is placed the chosen step. Steps must satisfy the tool's **Step Threshold** and **Step Direction** settings. **Best**: Selects the greatest step on the profile. **First**: Selects the first step on the profile. **Last**: Selects the last step on the profile.

Parameter	Description
Step Threshold	The minimum step accepted as an edge candidate. Steps on the profile are treated as absolute values when compared to this setting. In the following profile, with **Step Threshold** set to 1.7 (and **Selection Type** set to Last), the tool accepts the step to the right, with a step of -1.873 mm, because it is above the step threshold.  In the following, when **Step Threshold** is increased to 1.9, the tool excludes the falling step to the right, because it is no longer above the step threshold, and instead uses the step to the left. 
Step Direction	Determines whether the expected step rises or falls, moving left to right, along the profile. Either **Rising**, **Falling**, or **Rising or Falling**.
Step Smoothing	The size of the (moving) window along the profile used to calculate an average for each data point on the profile. The setting is useful for averaging out noise. If **Step Smoothing** is set to 0, no averaging is performed.

Parameter	Description
Step Width	The distance, along a path profile, separating the points the tool uses to find steps on a profile. In the following, a step width of 5.5 mm causes the tool to consider profile points that distance apart as steps. Consequently, the curved portion of the profile is not used to measure the step.  The setting is useful when you must detect a slope as an edge, rather than a sharply defined edge: setting **Step Width** to a value greater than the width of the edge ensures that the tool measures the height difference between the flat regions on either side of the edge. As a result, the height of the step is accurately measured, and the edge is correctly located. Setting **Step Width** wider than necessary can reduce the precision of edge location.
Max Gap	Fills in regions of missing data caused by an occlusion near the desired edge. Use this setting when continuity on the target is expected. When **Max Gap** is set to a non-zero value, the tool holds and extends the last data point on the low side next to an edge across a gap of null points, up to the distance specified in **Max Gap**.  The tool uses data points "filled in" by **Max Gap** before data points filled in by **Null Fill Value** (see below).

Parameter	Description
Include Null Edges	Indicates whether null points (points where no height value is available, due to dropouts or regions outside of the measurement range) are filled with the value in **Null Fill Value** as a general “background level.” To find an edges next null points, you must use either this option and an appropriate value in **Null Fill Value** or **Max Gap**. Otherwise, only edges within areas of contiguous data will be detected.
Null Fill Value	The height value (in mm) used to replace null points when **Include Null Edges** is enabled. If both **Null Fill Value** and **Max Gap** fill in null points at the same position, the tool uses the value extended by **Max Gap**, regardless of the value of **Null Fill Value**.
Show Detail	When disabled, reduces what is indicated in the data viewer.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

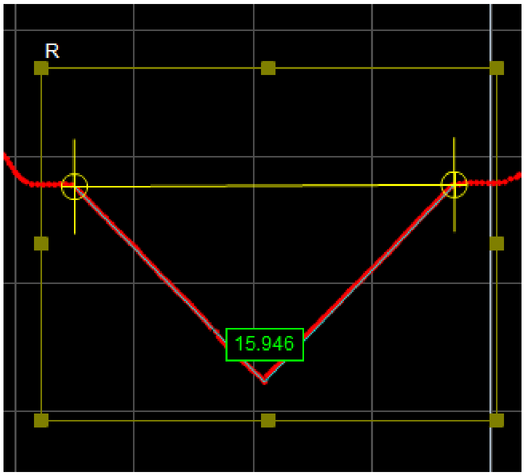
Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

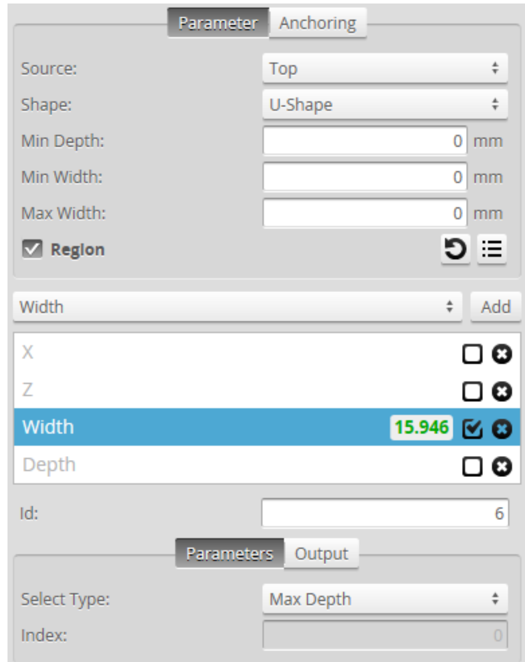
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Groove

The Groove tool provides measurements of V-shape, U-shape, or open-shape grooves.





The Groove tool uses a complex feature-locating algorithm to find a groove and then return measurements. See "Groove Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel.

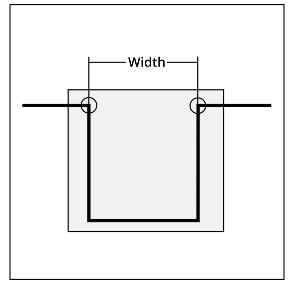
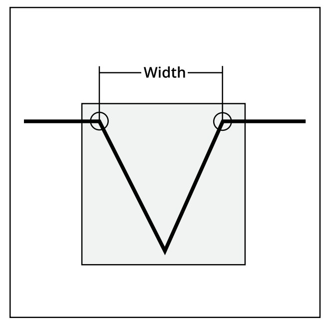
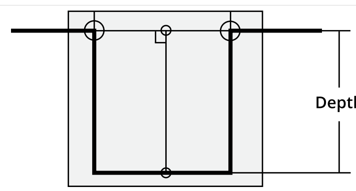
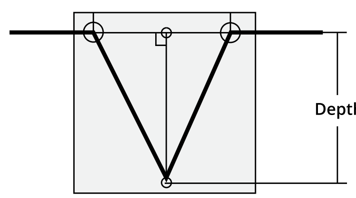
The Groove tool lets you add multiple measurements of the same type to receive measurements and set decisions for multiple grooves. Multiple measurements are added by using the drop-down above the list of measurements and clicking on the **Add** button.

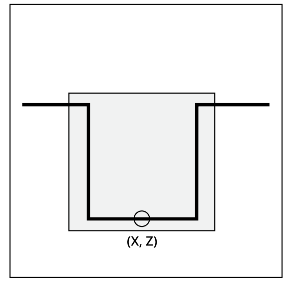
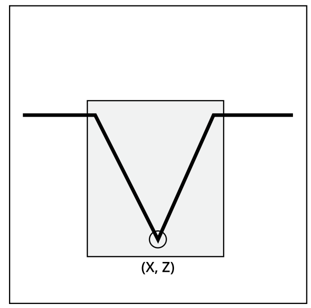
For example, if a target has three grooves, by adding two measurements, choosing **Index From The Left** in the **Select Type** setting of those measurements, and providing values of 0 and 2 in the **Index** setting of the measurements, respectively, the Groove tool will return measurements and decisions for the first and third grooves.

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

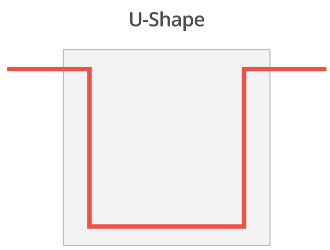
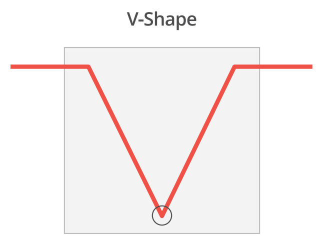
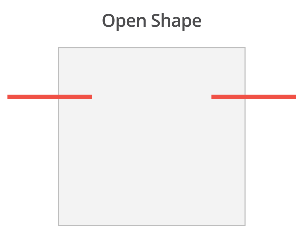
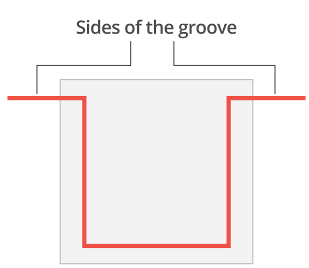
Measurements

Measurement	Illustration
Width Measures the width of a groove.	 
Depth Measures the depth of a groove as the maximum perpendicular distance from a line connecting the edge points of the groove.	 

Measurement	Illustration
X Measures the X position of the bottom of a groove.	
Z Measures the Z position of the bottom of a groove.	

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.

Parameter	Description
Shape	Shape of the groove 
Min Depth	Minimum depth for a groove to be considered valid.
Min Width	Minimum width for a groove to be considered valid. The width is the distance between the groove corners.
Max Width	Maximum width of a groove to be considered valid. If set to 0, the maximum is set to the width of the measurement area.
Region	The measurement region defines the region in which to search for the groove. For a stable measurement, the measurement region should be large enough to cover some data on the left and right sides of the groove.  For more information on regions, see *Regions* on page 152.

Parameter	Description
Location *(Groove X and Groove Z measurements only)*	Specifies the location type to return Bottom - Groove bottom. For a U-shape and open-shape groove, the X position is at the centroid of the groove. For a V-shape groove, the X position is at the intersection of lines fitted to the left and right sides of the groove. See algorithm section below for more details. Left - Groove's left corner. Right - Groove's right corner.
Select Type	Specifies how a groove is selected when there are multiple grooves within the measurement area. Maximum Depth - Groove with maximum depth. Index from The Left - 0-based groove index, counting from left to right Index from the Right - 0-based groove index, counting from right to left.
Index	0-based groove index.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

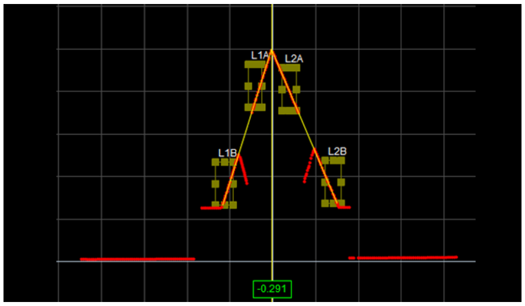
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

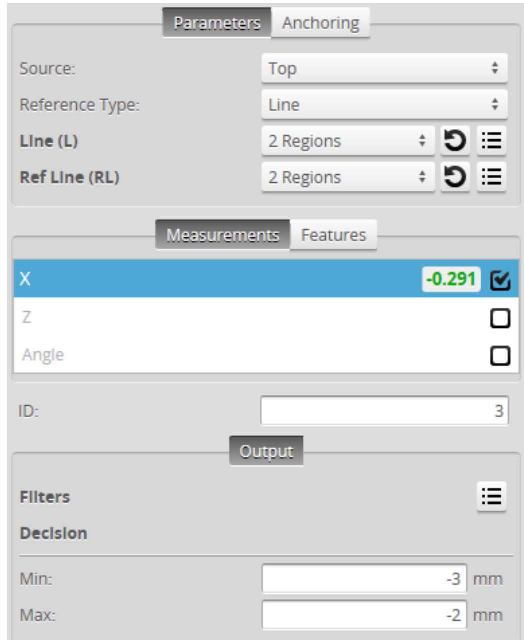
For more information on anchoring, see *Measurement Anchoring* on page 169.

Intersect

The Intersect tool determines intersect points and angles.

The Intersect tool's measurements require two fit lines, one of which is a reference line set to the X axis (z = 0), the Z axis (x = 0), or a user-defined line.

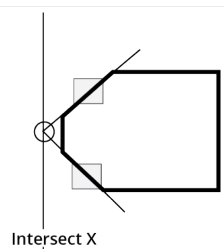


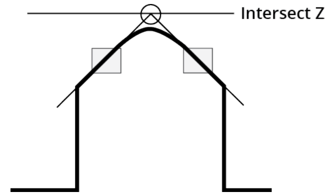
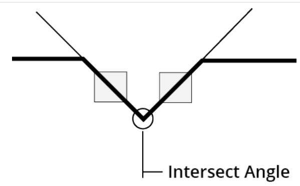


For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Finds the intersection between two fitted lines and measures the X axis position of the intersection point.	

Measurement	Illustration
Z Finds the intersection between two fitted lines and measures the Z axis position of the intersection point.	
Angle Finds the angle subtended by two fitted lines.	

Features

Type	Description
Intersect Point	The point of intersection.
Line	The intersect line.
Base Line	The base line.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
Reference Type	Determines the type of the reference line. **X-Axis**: The reference line is set to the X axis. **Z-Axis**: The reference line is set to the Z axis **Line**: The reference line is defined manually using the **Ref Line** parameter. One or two regions can be used to define the line.
Line	You can use one or two fit areas for the fit line. To set the region (or regions) of the fit line, adjust it graphically in the data viewer, or expand the feature using the expand button (☰) and enter the values in the fields. For more information on regions, see *Regions* on page 152. For more information on fit lines, see *Fit Lines* on page 166.

Parameter	Description
Ref Line	Used to define the reference line when **Line** is selected in the **Reference Type** parameter. To set the region (or regions) of the reference line, adjust it graphically in the data viewer, or expand the feature using the expand button (☰) and enter the values in the fields. For more information on regions, see *Regions* on page 152. For more information on fit lines, see *Fit Lines* on page 166.
Angle Range *(Angle measurement only)*	Determines the angle range. The options are: **-90 – 90** **0 – 180**
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

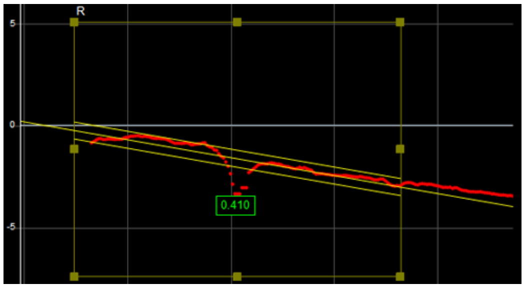
Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

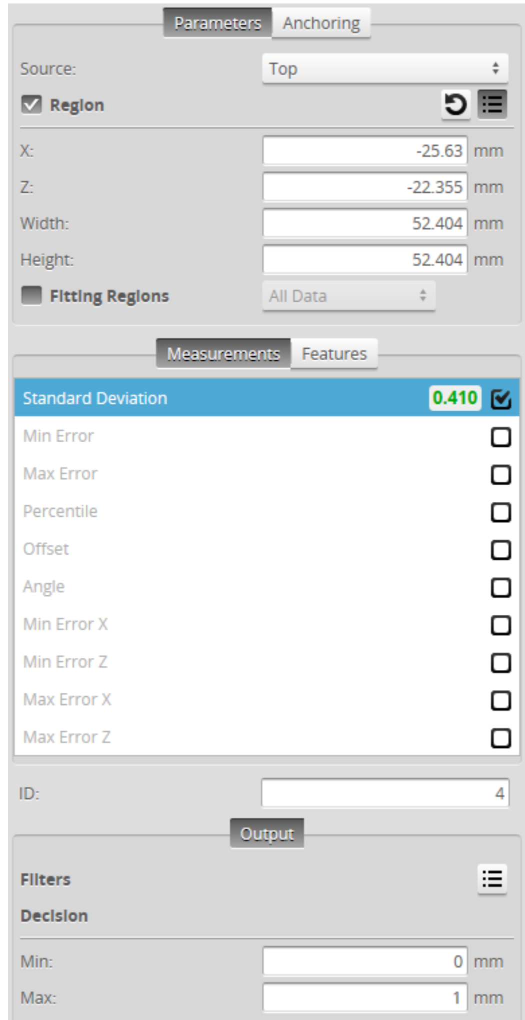
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Line

The Line tool fits a line to the profile and measures the deviations from the best-fitted line. The sensor compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 166.

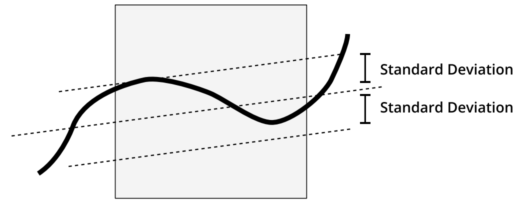
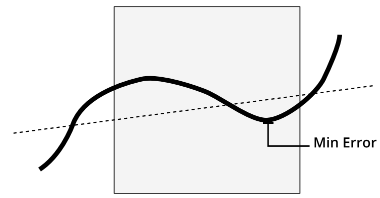
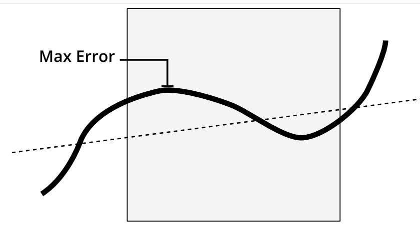
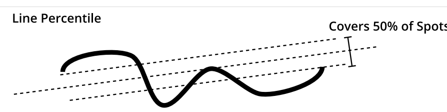
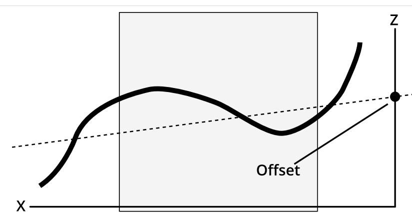
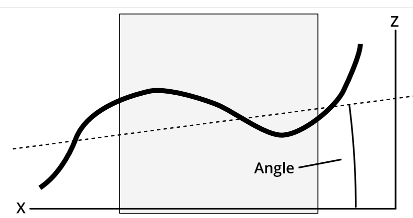
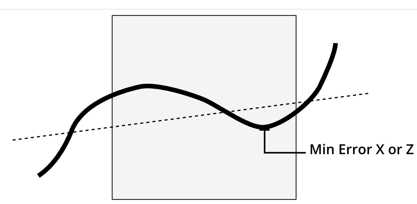


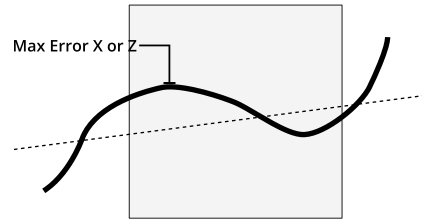


For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
Standard Deviation Finds the best-fitted line and measures the standard deviation of the data points from the line.	
Min Error Finds the best-fitted line and measures the minimum error from the line (the maximum distance below the line).	
Max Error Finds the best-fitted line and measures the maximum error from the line (the maximum distance above the line).	
Percentile Finds the best-fitted line and measures the range (in Z) that covers a percentage of points around the line.	
Offset Finds the best-fitted line and returns the intersection point between that line and the Z axis.	
Angle Finds the best-fitted line and returns the angle relative to the X axis.	
Min Error X **Min Error Z** Finds the best-fitted line and returns the X or Z position of the minimum error from the line (the maximum distance below the line).	

Measurement	Illustration
Max Error X **Max Error Z** Finds the best-fitted line and returns the X or Z position of the maximum error from the line (the maximum distance above the line).	

Features

Type	Description
Line	The fitted line.
Error Min Point	The point of minimum error.
Error Max Point	The point of maximum error.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.

Parameter	Description
Fitting Regions	Determines which data Gocator uses to *fit the line* over the profile. When **Fitting Regions** is enabled, Gocator uses the data indicated by one of the following options: • **All Data**: All of the data in the profile is used to fit the line. • **1 Region**: Data from a fitting region you define in the data viewer is used to fit the line. • **2 Regions**: Data from two fitting regions you define is used to fit the line. When **Fitting Regions** is disabled, to fit the line, Gocator uses the measurement region if **Region** is enabled, or the entire profile if **Region** is disabled. When **Fitting Regions** is enabled and **1 Region** or **2 Regions** is selected, you can set the region (or regions) graphically in the data viewer, or you can expand the feature using the expand button () and enter the values in the fields. For more information on regions, see *Regions* on page 152.
Percent *(Percentile measurement only)*	The specified percentage of points around the best-fitted line.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

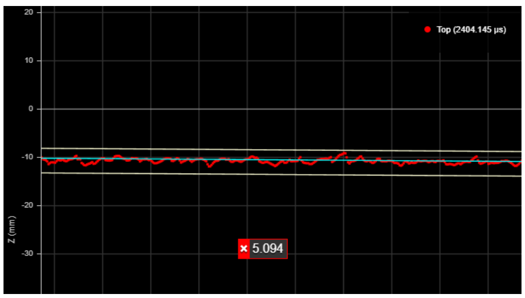
Line Advanced

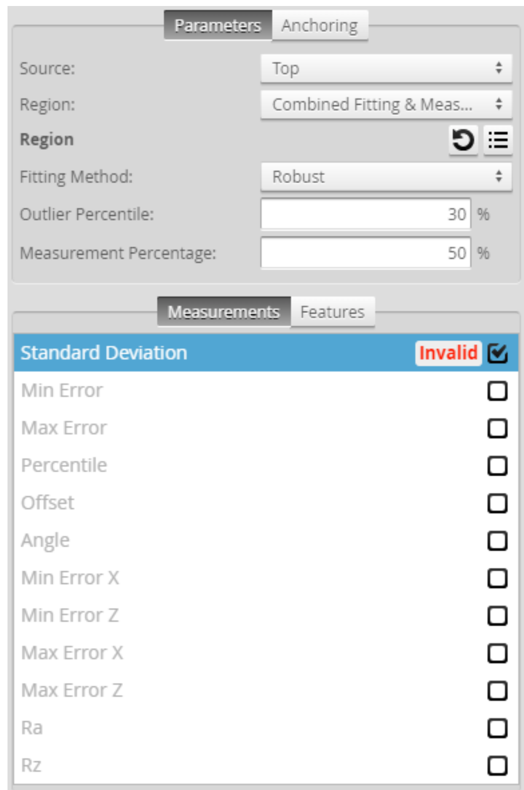
	This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

Like the Profile Line tool, the Profile Line Advanced tool fits a line to a profile and measures the deviations from the best-fitted line. Additionally, this version of the tool provides two “roughness parameter” measurements: Ra and Rz. Note that the region-related parameters have been reorganized to make the tool easier to use. The sensor compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 166.

	If you do not need the roughness parameters, LMI currently recommends using the Profile Line tool (see *Line* on page 245).

	Set **Fitting Method** to **Simple** to cause the tool to behave like the older Profile Line tool.

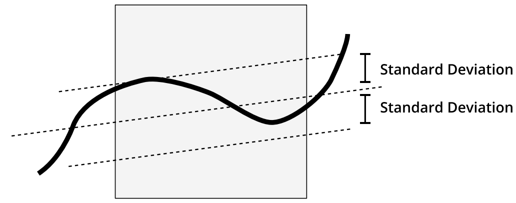
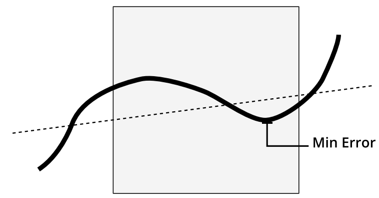
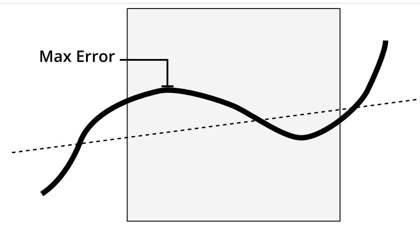
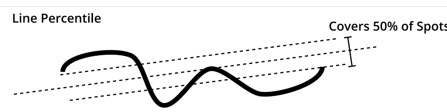
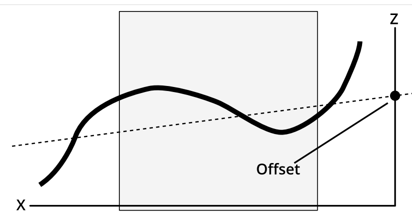
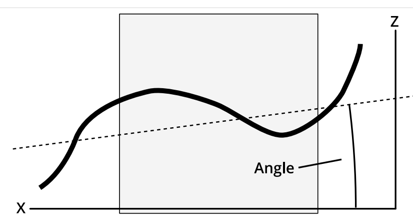
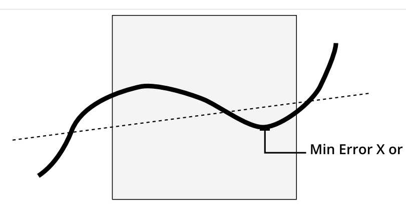


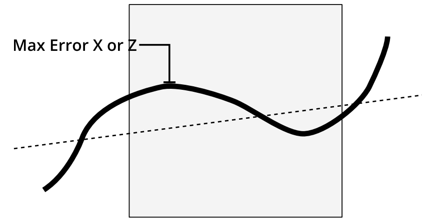
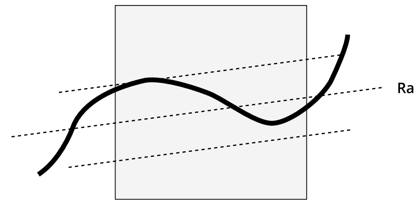
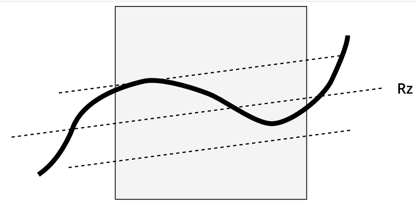


For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
Standard Deviation Finds the best-fitted line and measures the standard deviation of the data points from the line.	
Min Error Finds the best-fitted line and measures the minimum error from the line (the maximum distance below the line).	
Max Error Finds the best-fitted line and measures the maximum error from the line (the maximum distance above the line).	
Percentile Finds the best-fitted line and measures the range (in Z) that covers a percentage of points around the line.	
Offset Finds the best-fitted line and returns the intersection point between that line and the Z axis.	
Angle Finds the best-fitted line and returns the angle relative to the X axis.	
Min Error X **Min Error Z** Finds the best-fitted line and returns the X or Z position of the minimum error from the line (the maximum distance below the line).	

Measurement	Illustration
Max Error X **Max Error Z** Finds the best-fitted line and returns the X or Z position of the maximum error from the line (the maximum distance above the line).	
Ra Returns the roughness average of the profile data.	
Rz Returns the average maximum height of the profile data.	

Features

Type	Description
Line	The fitted line.
Error Min Point	The point of minimum error.
Error Max Point	The point of maximum error.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Region</td><td>

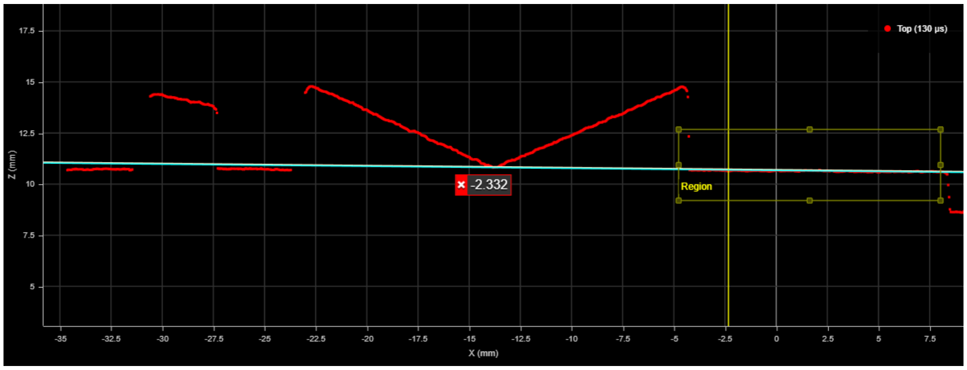
Whether the fitting and measurement regions are combined or separate (or not used). One of the following:

None

The tool uses the entire profile to fit the line and perform measurements.

Combined Fitting & Measurement

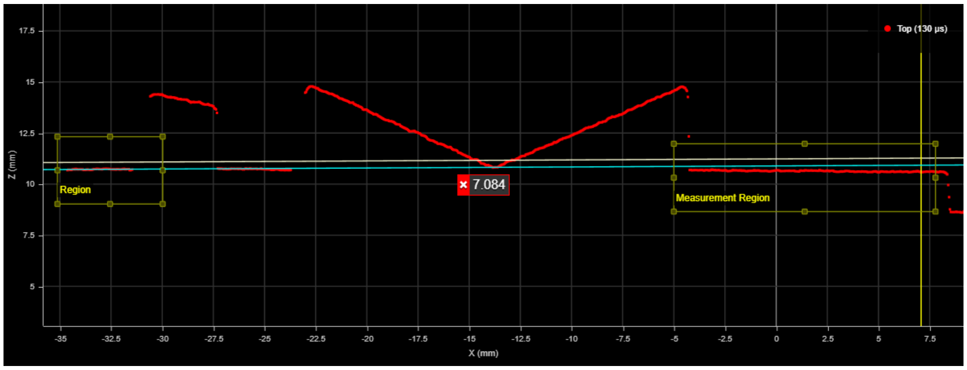
The tool uses a single, user-defined region to fit the line and in which it performs measurements. In the following image,



Separate Fitting & Measurement

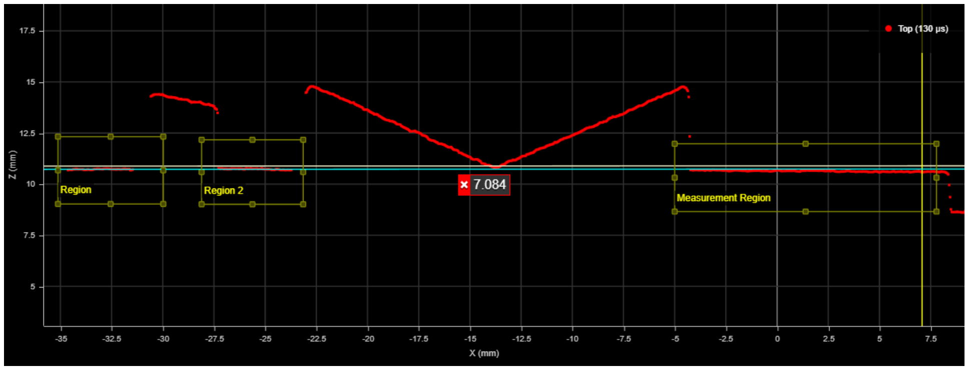
The tool uses one or two regions to fit the line, and a single, separate region in which it performs measurements.

In the following image, the tools uses a single region to the left to fit the line, and performs measurements in the measurement region to the right:



In the following image, the uses two regions to the left to fit the line, and performs measurements in the measurement region to the right:

</td></tr>
</table>

Parameter	Description
	
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
Region Region 2 Measurement Region (for region definition)	These settings contain parameters to define the position and size of the fitting and measurement regions.
Fitting Method	Determines how the tool fits the line to the data. One of the following: **Simple** Uses a less accurate but faster line-fitting method. Use this setting to cause the tool to behave like Profile Line. **Robust** An iterative line-fitting method that removes points and attempts to fit a line until only one-third of the original profile data points is left. More accurate but takes longer.
Outlier Percentile	Indicates the number of outlier points to be removed overall during line fitting. Adjust this value based on how much noise is present in the profile. Only displayed when **Method** is set to **Robust**.
Measurement Percentage *(Percentile measurement only)*	The specified percentage of points around the best-fitted line that the Percentile measurement uses.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

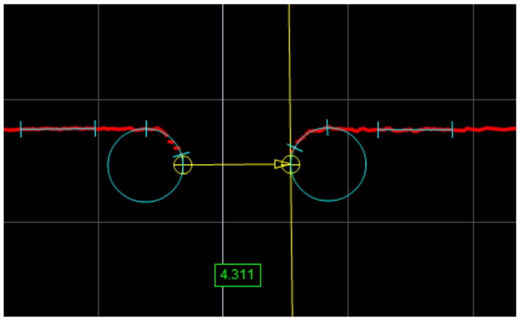
Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

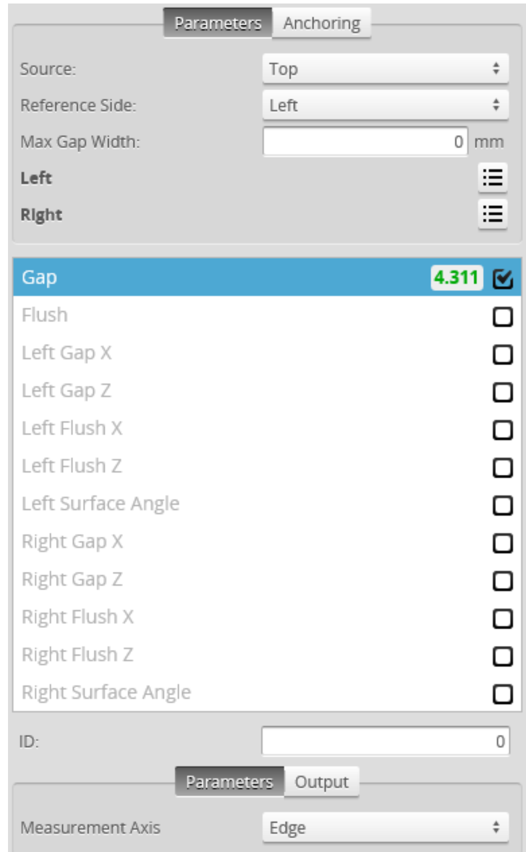
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Panel

The Panel tool provides Gap and Flush measurements.



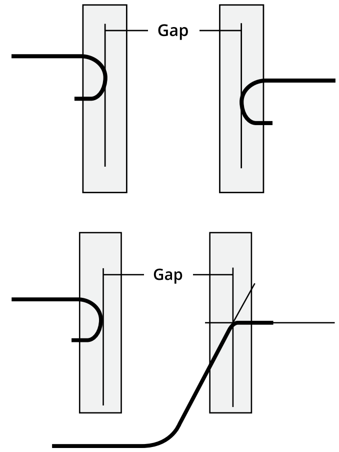
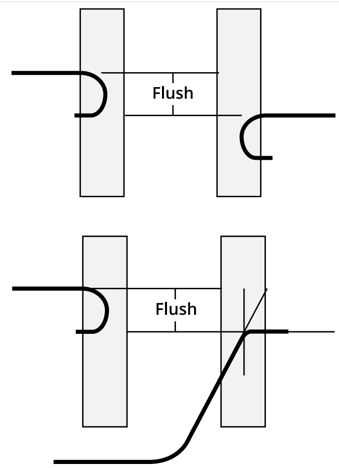


The Panel tool uses a complex feature-locating algorithm to find the gap or calculate flushness and return measurements. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel. See "Gap and Flush Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm.

	You must make sure that there are enough data points to define the edge in the profile, by properly settng up exposure, etc. If not, the algorithm will not function.

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements

Measurement	Illustration
Gap Measures the distance between two surfaces. The surface edges can be curved or sharp.	
Flush Measures the flushness between two surfaces. The surface edges can be curved or sharp.	
Left Gap X Returns the X position of the edge feature on the left side used to measure the gap.	
Left Gap Z Returns the Z position of the edge feature on the left side used to measure the gap.	
Left Flush X Returns the X position of the feature on the left side used to measure flushness.	
Left Flush Z Returns the Z position of the feature on the left side used to measure flushness.	
Left Surface Angle The angle of the left side surface relative to the X axis.	

Measurement	Illustration
Right Gap X Returns the X position of the edge feature on the right side used to measure the gap.	
Right Gap Z Returns the Z position of the edge feature on the right side used to measure the gap.	
Right Flush X Returns the X position of the feature on the right side used to measure flushness.	
Right Flush Z Returns the Z position of the feature on the right side used to measure flushness.	
Right Surface Angle The angle of the right side surface relative to the X axis.	

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
Reference SideDirection	Defines the side used to calculate the measurement axis (see below) rounded corner.
Max Gap Width	The maximum width of the gap. Allows the tool to filter gaps greater than the expected width. This can be used to single out the correct gap when there are multiple gaps in the field of view.
Measurement Axis *Gap measurement only*	Defines the direction that the gap is calculated, in relation to the reference side (see above). **Surface**: In the direction of the fitted surface line of the reference surface. **Edge**: In the direction perpendicular to the edge of the reference surface. **Distance**: The Cartesian distance between the two feature locations.
Absolute *Flush measurement only*	When enabled, returns an absolute value rather than a signed value.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Left/Right SideEdge Parameters

Parameter	Description
Max Void Width	The maximum allowed width of missing data caused by occlusion or data dropout.
Min Depth	Defines the minimum depth before an opening could be considered to have a potential edge. The depth is the perpendicular distance from the fitted surface line.
Surface Width	The width of the surface area in which data is used to form the fitted surface line. This value should be as large as the surface allows.
Surface Offset	The distance between the edge region and the surface region. Setting a small value allows the edge within a tighter region to be detected. However, the measurement repeatability could be affected if the data from the edge are considered as part of the surface region (or vice versa). A rule of thumb is to set **Surface Offset** equal to **Nominal Radius**.
Nominal Radius	The radius of the curve edge that the tool uses to locate the edge region.
Edge Angle	A point on the best fit circle to be used to calculate the feature point. The selected point is on the circumference at the specified angle from the start of the edge region. The angle is measured from the axis perpendicular to the fitted surface line.
Edge Type	Defines the type of feature point to use for the edge (Corner or Tangent). A tangent edge point is the point selected based on the defined Edge Angle. A corner edge point is the intersect point between the fitted surface line and a edge line formed by interpolating the points at and after the tangent within the edge region.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

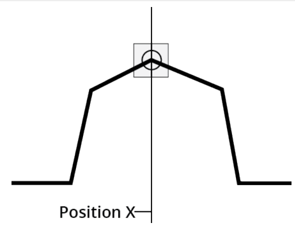
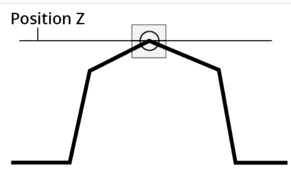
Position

The Position tool finds the X or Z axis position of a feature point. The feature type must be specified and is one of the following: Max Z, Min Z, Max X, Min X, Corner, Average (the mean X and Z of the data points), Rising Edge, Falling Edge, Any Edge, Top Corner, Bottom Corner, Left Corner, Right Corner, or Median (median X and Z of the data points).

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Finds the position of a feature on the X axis.	
Z Finds the position of a feature on the Z axis.	

Features

Type	Description
Point	The returned position.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.

Parameter	Description
Feature	The feature the tool uses for its measurements. One of the following: • Max Z • Min Z • Max X • Min X • Corner • Average • Rising Edge • Falling Edge • Any Edge • Top Corner • Bottom Corner • Left Corner • Right Corner • Median To set the region of a feature, adjust it graphically in the data viewer, or expand the feature using the expand button (☰) and enter the values in the fields. For more information on regions, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

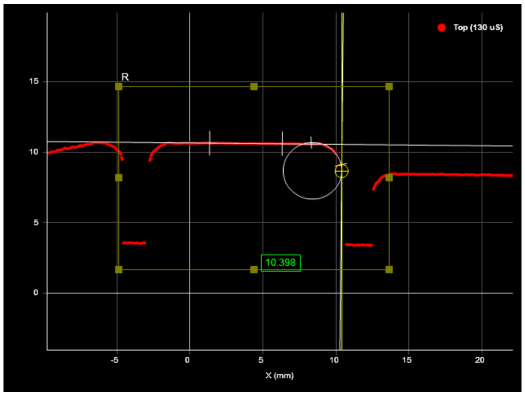
Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

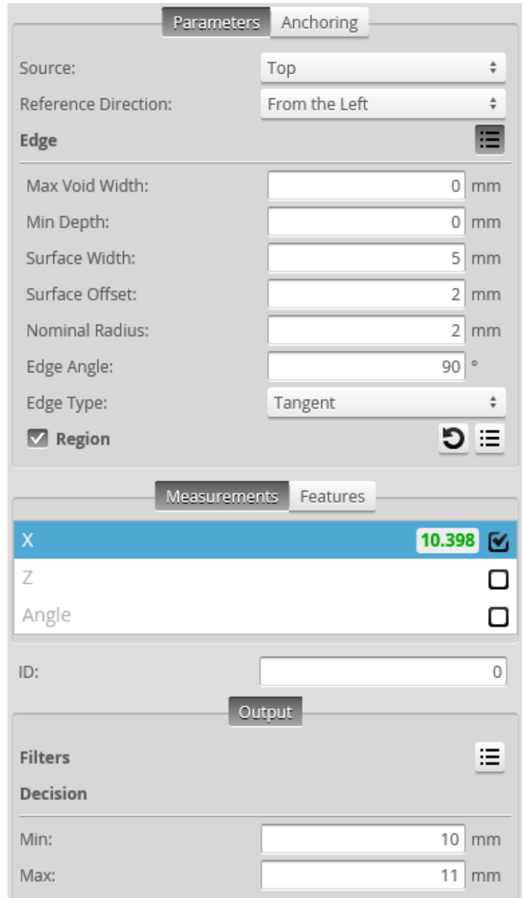
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Round Corner

The Round Corner tool measures corners with a radius, returning the position of the edge of the corner and the angle of adjacent surface with respect to the X axis.



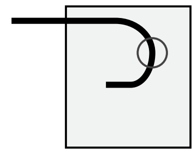
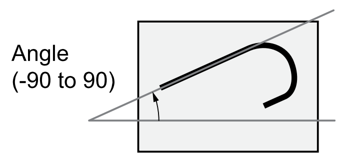
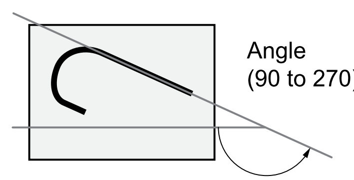


For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

The Round Corner tool uses a complex feature-locating algorithm to find the edge and return measurements. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel. See "Gap and Flush Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm.

> You must make sure that there are enough data points to define the edge (proper exposure, etc.). If not, the algorithm will not function.

Measurements

Measurement	Illustration
X Measures the X position of the location where the tangent touches the edge, or intersect of the tangent and the line fitted to the surface used by the measurement (see **Reference Side**, below).	
Z Measures the Z position of the location where the tangent touches the edge, or intersect of the tangent and the line fitted to the surface used by the measurement (see **Reference Side**, below).	
Angle Measures the angle of the line fitted to the surface next to the corner (see **Reference Side**, below), with respect to the x-axis. Left edge angles are from -90 to 90. Right edge angles are from 90 to 270.	 

Features

Type	Description
Edge Point	The position of the edge.
Radius Center Point	The center of the radius.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
Reference Direction	Defines the side used to calculate the rounded corner.
Max Gap Width	The maximum width of the gap. Allows the tool to filter gaps greater than the expected width. This can be used to single out the correct gap when there are multiple gaps in the field of view.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Edge Parameters

Parameter	Description
Max Void Width	The maximum allowed width of missing data caused by occlusion or data dropout.
Min Depth	Defines the minimum depth before an opening could be considered to have a potential edge. The depth is the perpendicular distance from the fitted surface line.
Surface Width	The width of the surface area in which data is used to form the fitted surface line. This value should be as large as the surface allows.
Surface Offset	The distance between the edge region and the surface region. Setting a small value allows the edge within a tighter region to be detected. However, the measurement repeatability could be affected if the data from the edge are considered as part of the surface region (or vice versa). A rule of thumb is to set **Surface Offset** equal to **Nominal Radius**.
Nominal Radius	The radius of the curve edge that the tool uses to locate the edge region.
Edge Angle	A point on the best fit circle to be used to calculate the feature point. The selected point is on the circumference at the specified angle from the start of the edge region. The angle is measured from the axis perpendicular to the fitted surface line.
Edge Type	Defines the type of feature point to use for the edge (Corner or Tangent). A tangent edge point is the point selected based on the defined Edge Angle. A corner edge point is the intersect point between the fitted surface line and a edge line formed by interpolating the points at and after the tangent within the edge region.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.

Anchoring

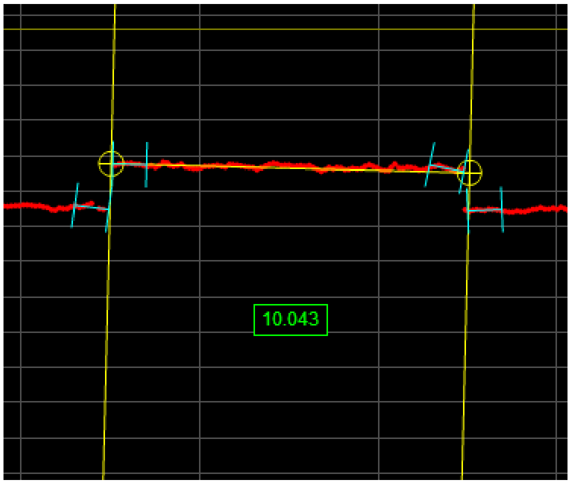
Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

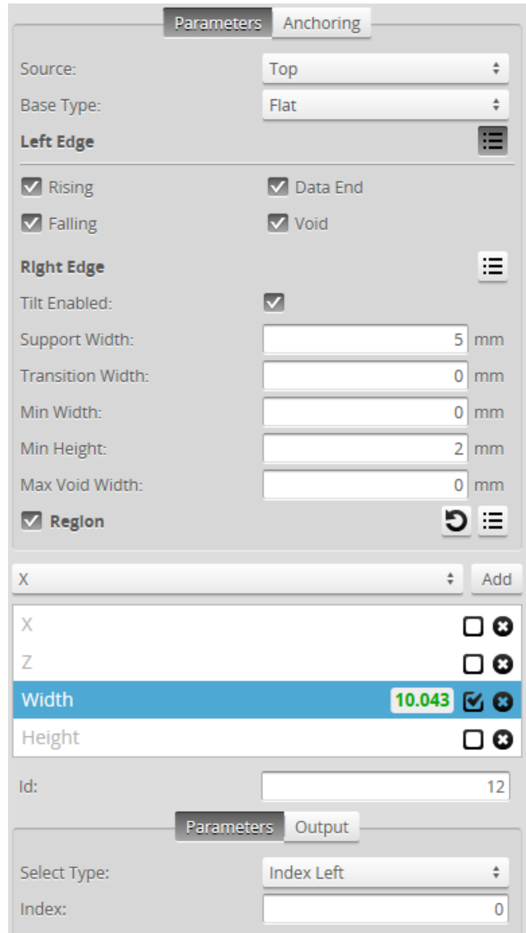
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Strip

The Strip tool measures the width of a strip.





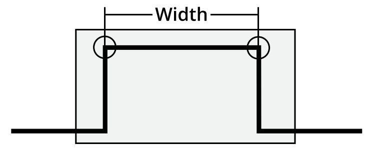
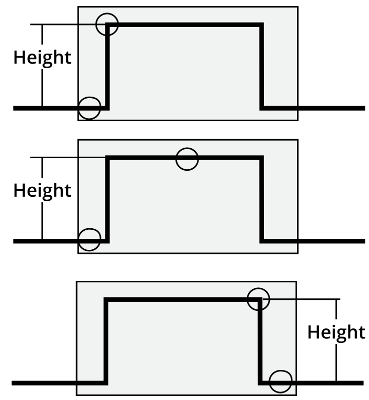
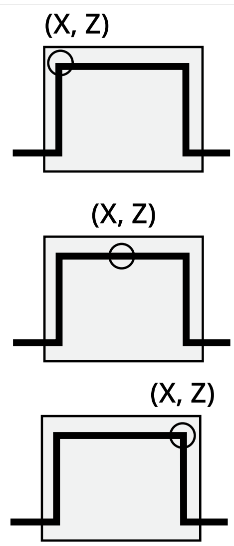
The Strip tool uses a complex feature-locating algorithm to find a strip and then return measurements. See "Strip Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel.

The Strip tool lets you add multiple measurements of the same type to receive measurements and set decisions for multiple strips. Multiple measurements are added by using the drop-down above the list of measurements and clicking on the **Add** button.

For example, if a target has three strips, by adding two measurements, choosing **Index From The Left** in the **Select Type** setting, and providing values of 1 and 3 in the **Index** of field of the measurements, respectively, the Strip tool will return measurements and decisions for the first and third strip.

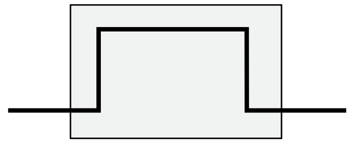
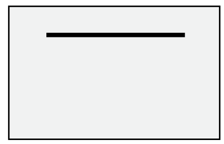
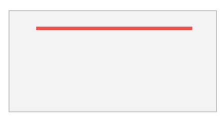
For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

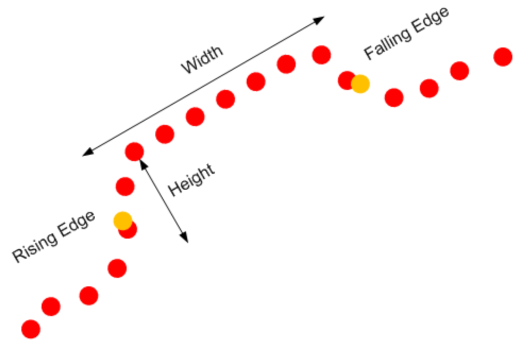
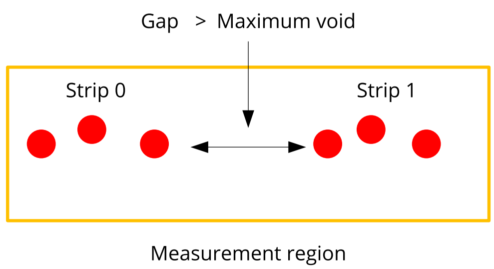
Measurements

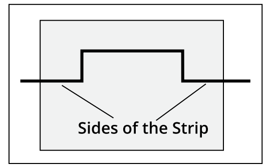
Measurement	Illustration
Width Measures the width of a strip.	
Height Measures the height of a strip.	
X Measures the X position of a strip.	
Z Measures the Z position of a strip.	

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.

Parameter	Description
Base Type	Affects detection of rising and falling edges. Base Type = Flat  Base Type = None  Base Type = Flat  Base Type = None  When **Base Type** is set to **Flat**, both strip (raised area) and base support regions are needed. When set to **None**, only a point that deviates from a smooth strip support region is needed to find a rising or falling edge.
Left Edge Right Edge	Specifies the features that will be considered as the strip's left and right edges. You can select more than one condition. **Rising** - Rising edge detected based on the strip edge parameters. **Falling** - Falling edge detected based on the strip edge parameters. **Data end** - First valid profile data point in the measurement region. **Void** - Gap in the data that is larger than the maximum void threshold. Gaps connected to the measurement region's boundary are not considered as a void. See "Strip Start and Terminate Conditions" in the *Gocator Measurement Tool Technical Manual* for the definitions of these conditions.

Parameter	Description
Tilt Enabled	Enables/disables tilt correction. The strip may be tilted with respect to the sensor's coordinate X axis. This can be caused by conveyor vibration. If the Tilt option is enabled, the tool will report the width and height measurements following the tilt angle of the strip. 
Support Width	Specifies the width of the region around the edges from which the data is used to calculate the step change. See "Strip Step Edge Definitions" in the *Gocator Measurement Tool Technical Manual* on how this parameter is used by different base types.
Transition Width	Specifies the nominal width needed to make the transition from the base to the strip. See "Strip Step Edge Definitions" in the *Gocator Measurement Tool Technical Manual* on how this parameter is used by different base types.
Min Width	Specifies the minimum width for a strip to be considered valid.
Min Height	Specifies the minimum deviation from the strip base. See "Strip Step Edge Definitions" in the *Gocator Measurement Tool Technical Manual* on how this parameter is used for different base types.
Max Void Width	The maximum width of missing data allowed for the data to be considered as part of a strip when **Void** is selected in the **Left** or **Right** parameter. This value must be smaller than the edge **Support Width**.  When occlusion and exposure causes data drops, users should use the gap filling function to fill the gaps. See *Gap Filling* on page 109 for information.

Parameter	Description
Region	The measurement region defines the region in which to search for the strip. If possible, the region should be made large enough to cover the base on the left and right sides of the strip.  For more information, see *Regions* on page 152.
Location *(Strip Height, Strip X, and Strip Z measurements only)*	Specifies the strip position from which the measurements are performed. **Left** - Left edge of the strip. **Right** - Right edge of the strip. **Center** - Center of the strip.
Select Type	Specifies how a strip is selected when there are multiple strips within the measurement area. **Best** - The widest strip. **Index Left** - 0-based strip index, counting from left to right. **Index Right** - 0-based strip index, counting from right to left.
Index	0-based strip index.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

> A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

> For more information on anchoring, see *Measurement Anchoring* on page 169.

Template Matching

> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

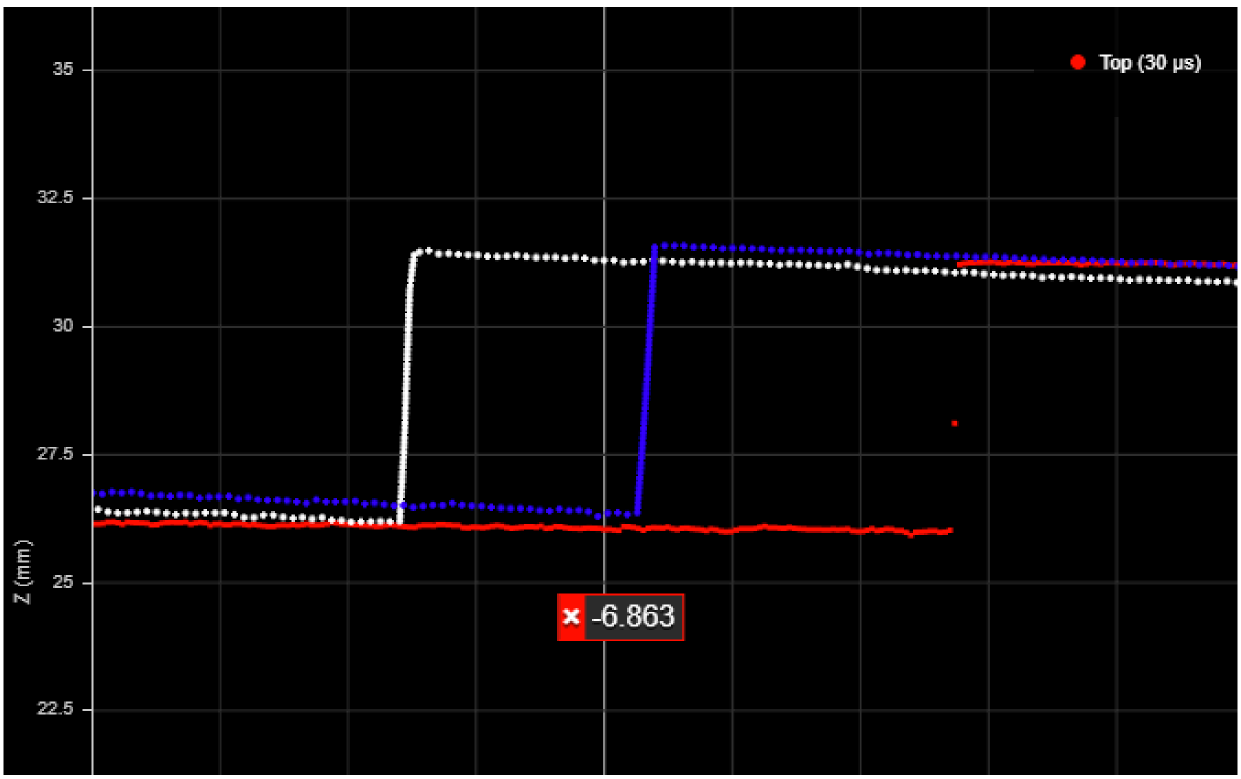
The Profile Template Matching tool lets you align a profile to a "master" template profile you create in the tool (a "golden template"), compensating for movement of the target from frame to frame. As a result, you can perform measurements over on a "stabilized" profile.

The tool returns measurements that represent differences between the profile and the master, letting you perform simple defect detection and location from within the tool.

The tool also outputs an aligned profile that other Profile measurement tools can use as input (via their **Stream** parameter). Finally, the tool produces a "difference" profile on which you can similarly perform measurements.

The sensor compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 166.

In the data viewer, the profiles are rendered using different colors:



The master profile is rendered in white. The aligned profile is rendered in blue. The current profile is rendered in red.

Note that in the image above, the tool is performing only a rough alignment to ensure that the different profiles are clearly visible. Typically, the blue aligned profile will be on top of the white master profile.

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Note that if no profile alignment is performed (both **Coarse Align** and **Fine Align** are disabled), for example, if the targets are sufficiently fixed from profile to profile, the following measurements return 0.000:

- Transform X
- Transform Z
- Transform Y Angle

Master Compare must be enabled for the following measurements; otherwise, they return Invalid values:

- Max Height Difference
- Max Difference Position X
- Max Difference Position Z
- Standard Deviation
- Difference Average
- Difference Sum
- Variance
- Matching Score

Also, for these "master compare" measurements, if the profile has been aligned to the master (either **Coarse Align** or **Fine Align** is enabled), the measurement compares the *aligned* profile and the master. If the profile has not been aligned (both alignment parameters are disabled), the measurement compares the *original* (unaligned) profile and the master.

Measurements

Measurement
Transform X **Transform Z** The distance the profile has shifted on the X and Z axis after alignment to the master, respectively.
Transform Y Angle The rotation of the profile around the Y axis after alignment.
Max Height Difference The maximum height difference between the profile and the master.
Max Difference Position X **Max Difference Position Z** The X and Z positions of the maximum height difference between the profile and the master.
Standard Deviation The standard deviation between the profile and the master.

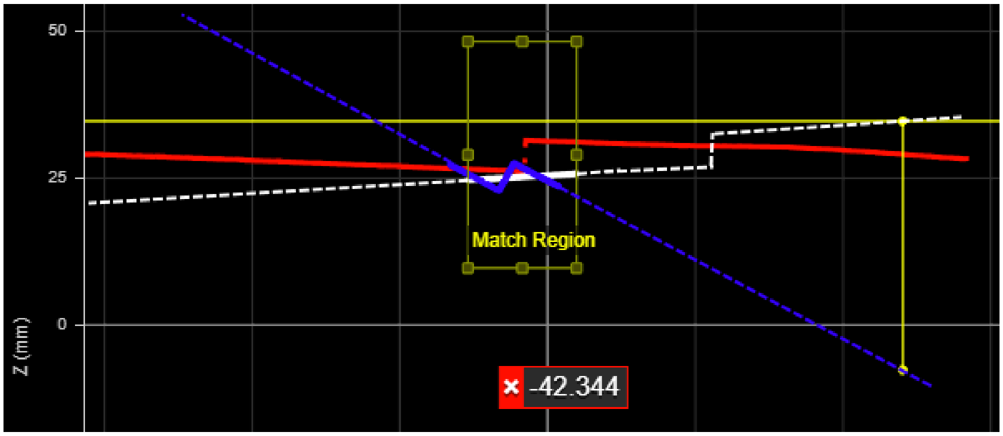
Measurement
Difference Average The average difference on the Z axis between the profile and the master.
Difference Sum The sum of the differences on the Z axis between the profile and the master.
Variance Returns the variance of a difference profile calculated by subtracting the current profile from the master.
Matching Score Returns a value between 0 and 1 that is the is the percentile of standard deviation of a difference profile (calculated by subtracting the current profile from the master) from the tolerance.

Data

Type	Description
Aligned Profile	The profile aligned to the master.
Difference Profile	A profile representing the differences between the profile and the master. Z values in the difference profile above 0 represent data points higher in the profile than in the master. Z values in the difference profile below 0 represent data points lower in the profile than in the master. Z values in the difference profile at 0 represent data points that are the same in the profile and the master.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
File	A list of templates available to the tool. The template containing the profile the tool uses as a master profile for alignment and comparisons. Use the **Operation** parameter to add and remove templates to this list.
Operation	Provides operations related to profile template files (masters). One of the following: **Save** – Saves the current profile to a template file in the local file system and adds it to the list in **File**. Multiple templates can be available. Files are persistent. **Delete** – Deletes the template file selected in **File**. (This parameter switches to “Normal” after the tool performs one of the file operations.)

Parameter	Description
Use Region	Indicates whether the tool uses a user-defined region to perform *matching*. (The tool uses *only* the data profile and master data in this region to perform matching.) If this option is not checked, the tool performs matching using data from the entire active area.
Match Region	Size and position of the region in which the matching (alignment) is performed.. Master comparison measurements however are applied to the entire profile (current profile and master). For example, in the following image, the tool limits matching to the data in the match region. But the measurement (Max Height Difference in this case) is calculated on the data outside the region.  (The dashed lines are added to illustrate the hidden aligned profile and master.)
Coarse Align	When enabled, shows the **X Shift Window** parameter. Use this setting by itself if you expect targets will only move along the X and Z axes (that is, you don't expect rotation). Otherwise, when combined with **Fine Align**, it provides a good initial start position for fine alignment.
X Shift Window	The maximum distance on the X axis the tool can move the current profile in order to align it. Should be set to the maximum amount the part is expected to shift left or right. (Enabled using the **Coarse Align** parameter.)
Fine Align	When enabled, lets you set the **Max Iteration** and **Match Window** parameters for fine alignment. This alignment method is more accurate than coarse alignment but takes more time to run.
Max Iteration	The maximum number of iterations the tool uses to perform fine alignment of the profile to the master.
Match Window	The region in which points are evaluated for a match. It there's a larger difference between the current profile and the master than the match window size, it would ignore the point.
Master Compare	Causes the tool to compare the current profile to the master profile and return results in some of the tool's measurements. (See list above.) When disabled, the measurements that compare the profile to the master return invalid values.
Difference Profile Median Size	Defines the size of the window the tool uses to smooth out noise in the Difference Profile data output.
Tolerance	The difference tolerance for the master comparison.

Parameter	Description
Display Master	Displays the Master template (white profile).
Display Aligned Profile	Displays the aligned (blue profile).
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

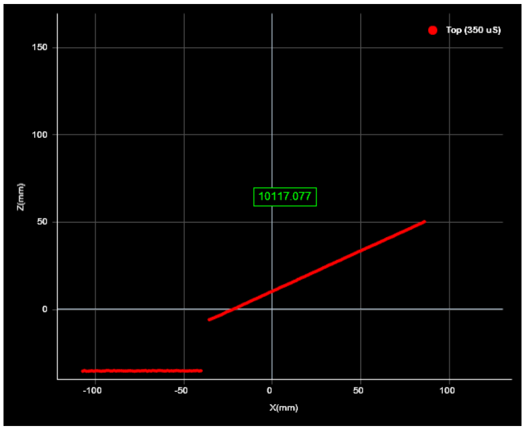
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

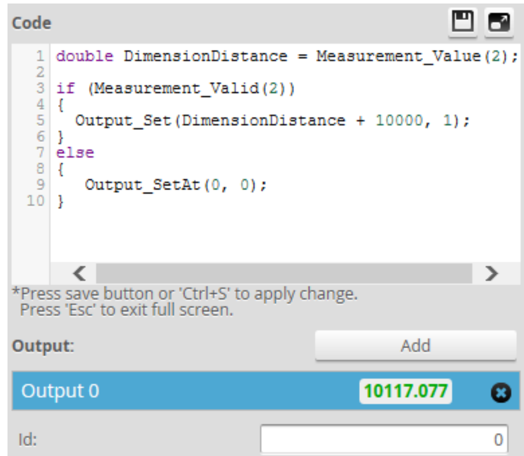
For more information on anchoring, see *Measurement Anchoring* on page 169.

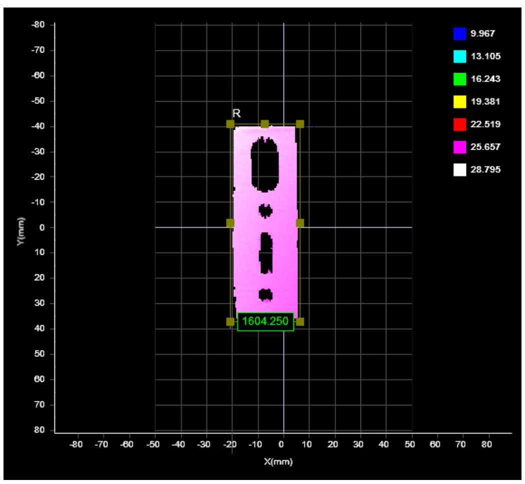
Script

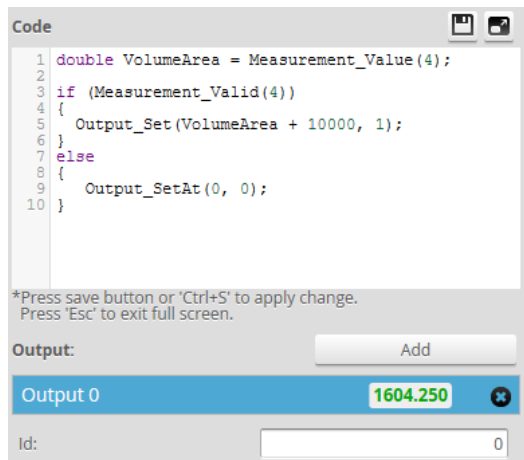
A Script measurement can be used to program a custom measurement using a simplified C-based syntax. A script measurement can produce multiple measurement values and decisions for the output.

For more information on script tool syntax, see *Scripts* on page 505.









To create or edit a Script measurement:

1. Add a new Script tool or select an existing Script measurement.

2. Edit the script code.

3. Add script outputs using the **Add** button.
   For each script output that is added, an index will be added to the **Output** drop-down and a unique ID will be generated.
   To remove a script output, click on the button next to it.

4. Click the **Save** button to save the script code.
   If there is a mistake in the script syntax, the result will be shown as a "Invalid" with a red border in the data viewer when you run the sensor.
   Outputs from multiple measurement tools can be used as inputs to the script. A typical script would take results from other measurement tools using the value and decision function, and output the result using the output function. Stamp information, such as time and encoder stamps, are available in the script, whereas the actual profile3D point cloud data is not. (The script engine is not powerful enough to process the data itself.) Only one script can be created.

Surface Measurement

Surface measurement involves capturing 3D point cloud data, optionally identifying discrete objects, and measuring properties of the surface or the objects, such as the volume of the object or the height at a certain position of the object. All volumetric tools have the ability to operate either on the entire surface or the full object, or within a region of interest at a certain position in relation to the surface or an object.

Multiple measurements can be performed on the entire surface or each discrete object, limited only by the available CPU resources.

Isolating Parts from Surface Data

Gocator lets you isolate and then measure parts in two different ways: by configuring the **Part Detection** panel on the **Scan** page in the web interface (for more information, see *Part Detection* on page 111); and using one of two Surface measurement tools (for more information on these tools, see *Blob* on page 289 and *Segmentation* on page 426).

The following table lists several differences between the two methods. A key difference however is that part detection extracts scan data that is identified as a "part" and outputs it as a *separate frame*. This lets you use any measurement tool on parts individually. Note however that parts must be clearly separated and be relatively consistently spaced for the part detection algorithm to separate the parts. In general, if you can successfully isolate parts using part detection, use this method rather than the Surface tools.

With the two Surface measurement tools on the other hand, areas are not extracted as individual frames, and for this reason you can't easily apply measurement tools to the areas individually: given that damaged areas may appear anywhere in the source surface data, you can't know where to place the measurement tools. The individual parts are however available for consumption by an SDK application or a GDK tool. (For information on the SDK and GDK, see *Development Kits* on page 836.) The main advantage of these tools is that they can separate objects that are touching. Although you can't apply other measurement tools to the identified blobs, the tools do provide measurements such as length, width, and area, which lets you handle common pass/fail needs.

Main Differences Between Part Detection, Surface Blob, and Surface Segmentation

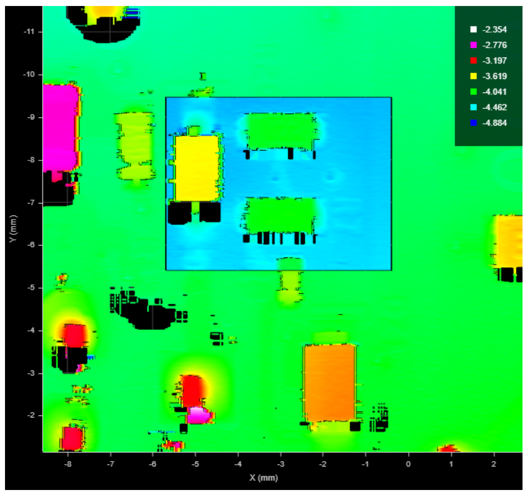
	Part Detection	**Surface Blob**	**Surface Segmentation**
Allows output of individual surfaces to separate frames	Yes	No	No
Allows separating touching objects	No	Yes - Limited Through Open filter, some connections between parts can be separated, but the control is more limited than with Surface Segmentation.	Yes
Supports background present	Yes Height threshold must be set above/below background	Yes Height threshold must be set above/below background	Yes Full support in firmware v6.0 and later
Supports background with	No	No	Yes

	Part Detection	Surface Blob	Surface Segmentation
significant tilt or intensity gradient	Fixed height threshold is used	Fixed height threshold is used	Adaptive threshold is used
Integrated Width/Length/Area measurements	N/A	Yes	Yes
Includes circularity and convexity filtering	No	Yes	No
Fast operation	Yes	Yes	No
Finds objects above or below background	Yes	Yes	Yes But requires careful region placement

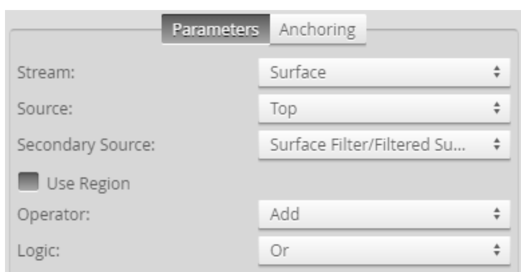
Arithmetic

🗋	This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Surface Arithmetic tool lets you perform various operations on a pair of surfaces. For example, you can use the tool to perform dynamic masking from frame to frame. The tool performs bitwise operations (AND or OR) on the corresponding data points in the source surfaces, and also combines height and intensity data with add, subtract, average, and mask operations.



2D View



Tool Setup

Settings

Parameters

Parameter	Description
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool. If you switch from one type of data to another (for example, from section profile data to surface data), currently set input features will become invalid, and you will need to choose features of the correct data type.

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151. Can only accept Surface scan data (that is, cannot accept data from other tools).
Secondary Source	The data output of another tool, for example, of a Surface Filter tool.
Use Region	Indicates whether the tool uses a user-defined region. If this option is not checked, the tool uses data from the entire active area.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Use Intensity	If enabled, the tool uses intensity data instead of heightmap data. Only available if **Acquire Intensity** is enabled on the Scan page during the scan; for more information, see *Scan Modes* on page 88.
Operator	One of the following: Add – Adds the height values of the corresponding data points in the two sources. Subtract – Subtracts the height values of the corresponding data points in the two sources. Average – Averages the height values of the corresponding data points in the two sources. Mask – Uses the secondary source as a mask.
Logic	Performs bitwise-operations on the source and secondary source surface data. One of the following: **And** or **Or**. When **Operator** is set to **Average**, this parameter is unavailable.

Data

Type	Description
Surface	The processed surface data.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

> A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

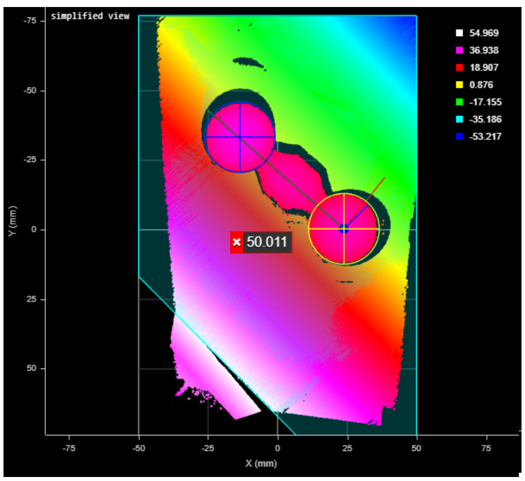
> For more information on anchoring, see *Measurement Anchoring* on page 169.

Ball Bar

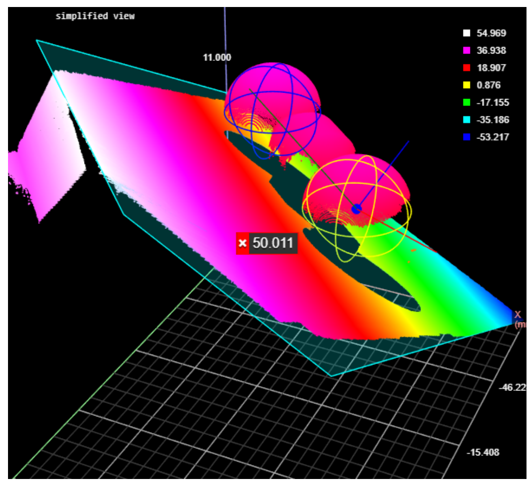
	This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Surface Ball Bar tool returns measurements useful for calibrating systems using a ball bar, particularly systems that include a robot.

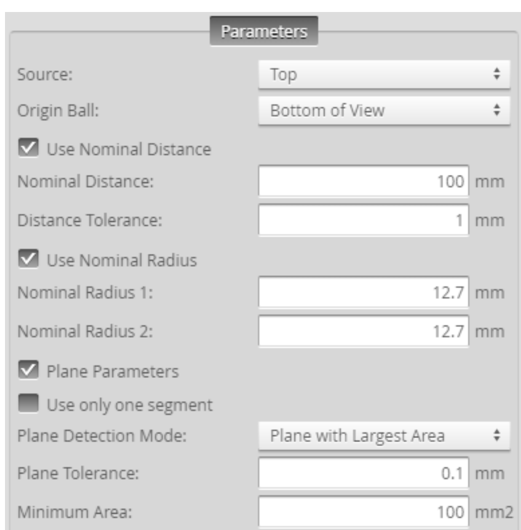
You can use this tool with the Gocator URCap plugin to quickly and easily perform hand-eye calibration between a G3 sensor and a Universal Robots robot. The Gocator URCap plugin lets you automatically add an instance of the Surface Ball Bar tool and enables the measurements and outputs the URCap needs, in the correct order. LMI recommends this over adding it manually via the Gocator web interface. For information on the plugin, see *Universal Robots Integration* on page 814. After adding it via the plugin, you only need to configure the parameters, described below. Note that you still have to configure other sensor settings, such as exposure.



2D View



3D View



Measurement Panel

Measurements, Data, Features, and Settings

Measurements

Measurement
Distance 3D The direct distance between the centers of the spheres fitted to the balls.
Center X1 / Y1 / Z1 **Center X2 / Y2 / Z2** These measurements return the X, Y, and Z positions of the centers of the spheres fitted to the balls. Ball 1 (Center X1 / Y1 / Z1) is always used as the origin. (Corresponds to the values returned in Tx / Ty / Tz.)
Normal X / Y / Z These measurements return the X, Y, and Z components of the normal vector of the surface surrounding the calibration target.
Ix / Iy / Iz **Jx / Jy / Jz** **Kx / Ky / Kz** These measurements return the X, Y, and Z components of the I, J, and K unit vectors defining the coordinate system orientation.
Tx / Ty / Tz These measurements return the X, Y, and Z components of the translation vector defining the coordinate system origin location.
Processing Time The time the tool takes to run.

Data

Type	Description
Difference Surface	Used for diagnostics.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Origin Ball	Determines which ball is used as the origin. The **Bottom of View** option selects the ball at the bottom of the data viewer in the Gocator web interface.
Use Nominal Distance	When enabled, displays **Nominal Distance** and **Distance Tolerance** settings. Set these to reflect the distance between the balls of the ball bar (refer to the specifications of the ball bar) and the tolerance you need. This can be useful to ensure invalid results due to false or inaccurate detection are rejected.

Parameter	Description
Use Nominal Radius	When enabled, displays **Nominal Radius** settings. Set these to reflect the radius of the balls of the ball bar (refer to the specifications of the ball bar) and the tolerance you need. This can be useful to ensure invalid results due to false or inaccurate detection are rejected.
Plane Parameters	Enables advanced plane settings. For UR integration, you should leave the settings at their default. These parameters allow ensuring the plane detection is accurate and robust to variations.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Barcode

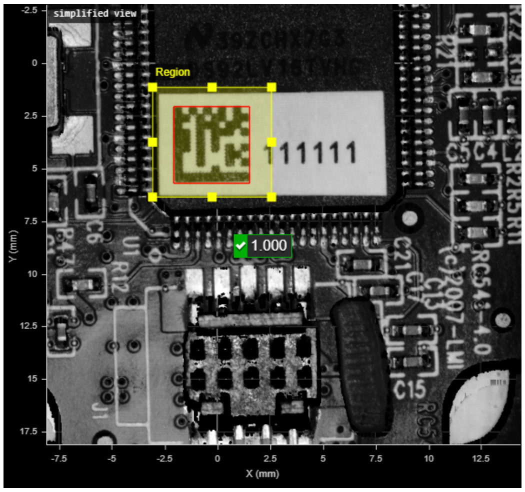
> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Barcode tool lets you decode data encoded in 1D (linear) and 2D barcodes from surface data (intensity data or heightmap data) without the need for 2D vision cameras or barcode readers. The tool also supports dot-peened types (Datamatrix and QR code). For a complete list of the types the tool supports, see "Type" in *Parameters* on page 286.

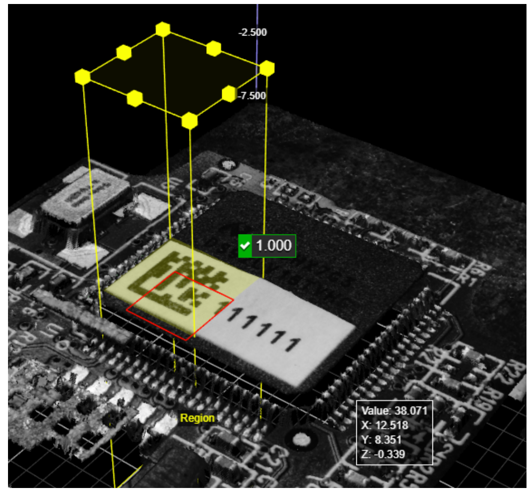
> When configuring the tool, make sure you switch the data viewer to the appropriate type of visualization for the barcode: for intensity-based barcodes (such as printed barcodes), switch the data viewer to intensity mode using the Intensity button ( ); for height-based barcodes (such as dot peen codes), switch the data viewer to heightmap mode using the Heightmap button ( ).

The tool returns whether it has found the barcode and whether it is valid, as well as the X, Y, and Z position of the barcode's lower left corner.

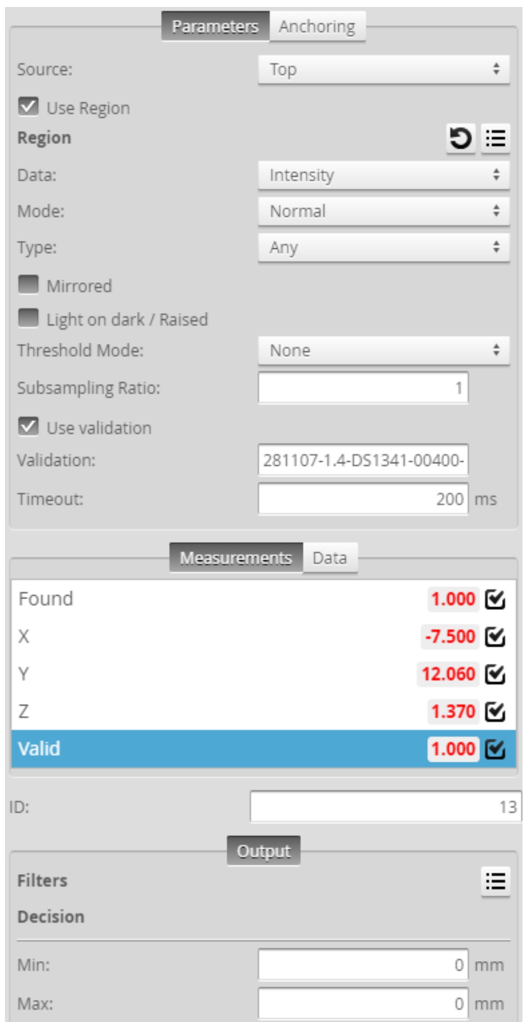
You can use the String Encoding tool to extract the string and pass it as output to a PLC; for more information, see *String Encoding* on page 442. The extracted string is also available via the SDK; for information on the SDK, see *GoSDK* on page 836 and the SDK reference documentation.



2D View



3D View



Measurement Panel

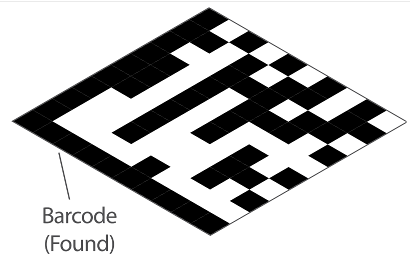
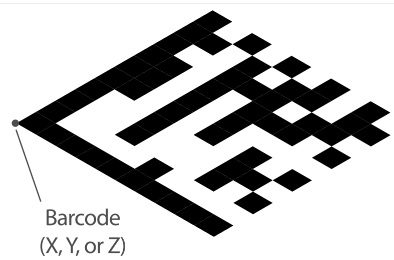
The decoded data is also displayed in the log; for more information on the log, see *Log* on page 73.

The tool provides two "learn" functions that can speed up the process of determining appropriate settings. (For more information, see **Mode** below.)

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
Found Returns 1.000 if the tool detects the configured barcode; otherwise, 0. Places a red rectangle around detected QR codes and Datamatrix codes.	
X **Y** **Z** These measurements return the X, Y, and Z position of the code, respectively.	
Valid	Determines whether the barcode is valid by comparing the string in the **Validation** parameter with the decoded string.

Data

Type	Description
Output String	Data output containing the decoded string.
Location Image	The image the tool uses to find the a dot-peen barcode. (When **Type** is set to a printed barcode, that is, a type other than a dot-peen code, this image is the same as the decode image.)
Decode Image	The image the tool uses as part of the dot peen decode algorithm. Use this to adjust the image (for example, using one of the filter tools) and to diagnose issues.

Type	Description
Dot peen decode Image	A binarized image the tool runs the dot peen decode algorithm on. The points of the code should appear clearly in the image to ensure proper decoding. Use this to adjust the image (for example, using one of the filter tools) and to diagnose issues.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Use Region	Indicates whether the tool uses a user-defined region. If this option is not checked, the tool uses data from the entire active area.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Data	The data the tool uses to detect a bar code. One of the following: • Intensity • Heightmap
Type	The type of barcode the tool expects. One of the following: • Any: Detects any type of barcode. • 1D Barcode (All): Detects any type of 1D (linear) barcode. • EAN-8 • EAN-13 • ISBN-10 • ISBN-13 • UPC-A • UPC-E • Code-39 • Code-128 • Interleave 2 of 5 • PDF417 • Data Matrix • Data Matrix dot peened • QR Code • QR Code dot peened
Mirrored	Reverses the scan. Use this if the scan is mirrored. Only useful with 2D barcodes.
Light on dark / Raised	If you are scanning *light-on-dark* barcodes or *raised* barcodes, enable this option.
Use Threshold	Enables the **Threshold** setting (see below).

Parameter	Description
Threshold Mode	Sets the threshold mode the tool uses. Any data points below the threshold are ignored and considered part of the "background"; data points not excluded are considered part of the barcode. Useful for cases where the surrounding surface is similar to the intensity or height of the barcode itself. One of the following: • None: No thresholding is performed. • Fixed: A global thresholding method. Set **Threshold** to a value between 0 and 255. When **Data** is set to Intensity, the value in **Threshold** is simply the intensity cut-off. When **Data** is set to Heightmap, the value is a percentile of the height values, converted to the 0-255 range. • Otsu: A global thresholding method. Illumination of the target should be relatively uniform and tilt should be removed (for example, using the Surface Transform tool; see *Transform* on page 465). • Adaptive: A local thresholding method that can help deal with local variation (intensity or height) in the target.
Threshold	The threshold of intensity or height values the tool uses to distinguish between the code and the surrounding surface. The parameter accepts a value between 0 and 255, whether **Data** is set to Heightmap or Intensity. This setting is only displayed when **Threshold Mode** is set to **Fixed**.
Subsampling ratio	Downsamples the image. Can make the tool run faster. (A value of at least 2 is usually necessary.)
Use validation	Enables validation of the decoded string, using the string in **Validation** for the comparison.
Validation	The case-sensitive string the tool compares to the decoded string. The parameter does not support wild cards or truncated values. If the comparison is valid, the Valid measurement returns 1.000.
Timeout	The maximum time the tool is allowed to take.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Blob

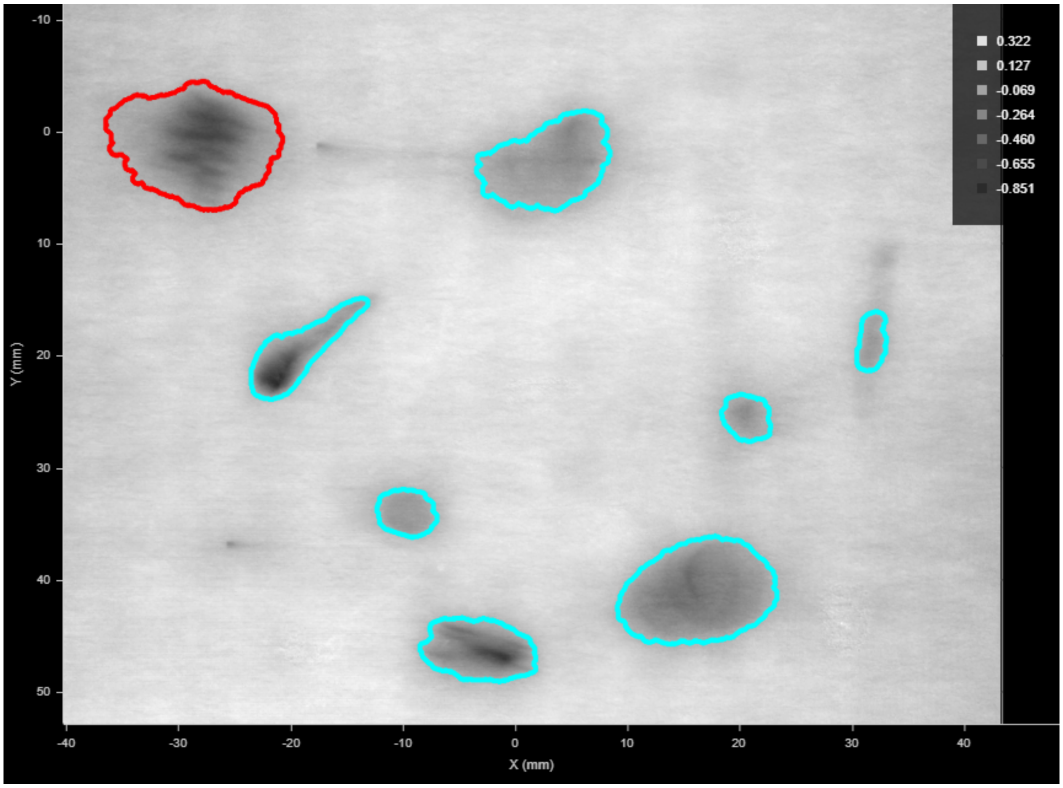
> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Surface Blob tool lets you detect surface defects, such as uneven or excess material, gouges, or blemishes, on a relatively uniform or flat background, in either 3D height map data or intensity data. It can also extract targets from the surface. The tool optionally lets you set its height threshold relative to a user-defined reference region. It also lets you use a reference plane to correct for an overall tilt of the target surface; this lets you detect low or shallow defects that would otherwise not be detectable due to a tilt.

> The Surface Blob tool provides functionality similar to the Surface Segmentation tool. For a comparison of these tools and the part detection capabilities you can configure on the **Scan** page, see *Isolating Parts from Surface Data* on page 277. For information on the Surface Segmentation tool, see *Segmentation* on page 426.

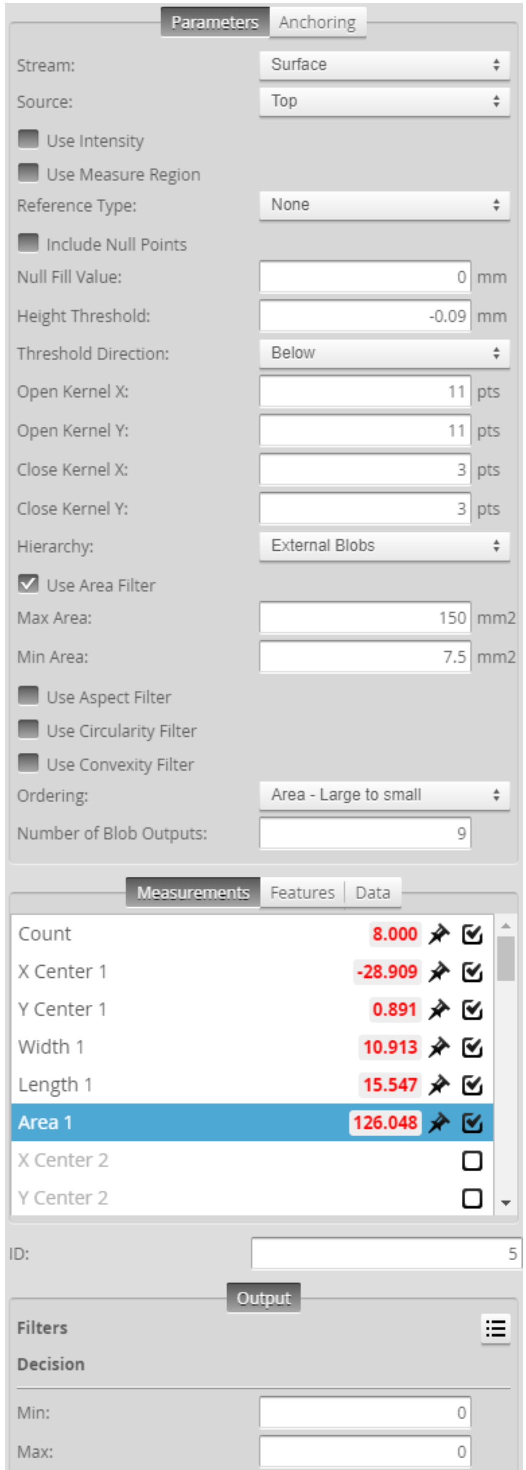
The tool first filters data based on a height or intensity threshold (above or below it), and then uses configurable morphological operations to better isolate parts. Finally, the tool uses various size- and shape-based filters that let you exclude or include the expected defects or the targets you need (potential blobs).

The tool lets you configure the maximum number of "blobs" to output, and returns the total blob count, and for each blob, the X and Y center, the width and length, and the area. The center point of each blob is available as a geometric feature. The blobs themselves are available in an array that can be accessed and processed by an SDK application or a GDK tool. For more information on the SDK, see *GoSDK* on page 836. For more information on the GDK, see *GDK* on page 847.



Several dents outlined on a surface. The currently selected blob is outlined in red. (Grayscale heightmap mode is used to better see the outlines.)

Note that knowing the rough size and shape of the kinds of detects you expect is important when you are configuring the open and close kernels and the tool's filters.



Tool configuration panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Data, and Settings

Measurements

Measurement
Count Returns the total number of blobs identified, based on the tool's parameters.
Area {n} The area of a blob. The area is calculated using the contour of the blob and resampling. For this reason, areas calculated using the Surface Volume tool will produce different measurements; for more information, see *Area* on page 481.
X Center {n} **Y Center {n}** The X and Y positions of the center of mass of a blob extracted from the surface. The **Number of Blob Outputs** setting determines the number of measurements listed in the **Measurements** tab.
Length {n} **Width {n}** The length and width of the rotated bounding box that encapsulates the blob extracted from the surface. These are always the major and minor axis of a blob, respectively. The **Number of Blob Outputs** setting determines the number of measurements listed in the **Measurements** tab.

Features

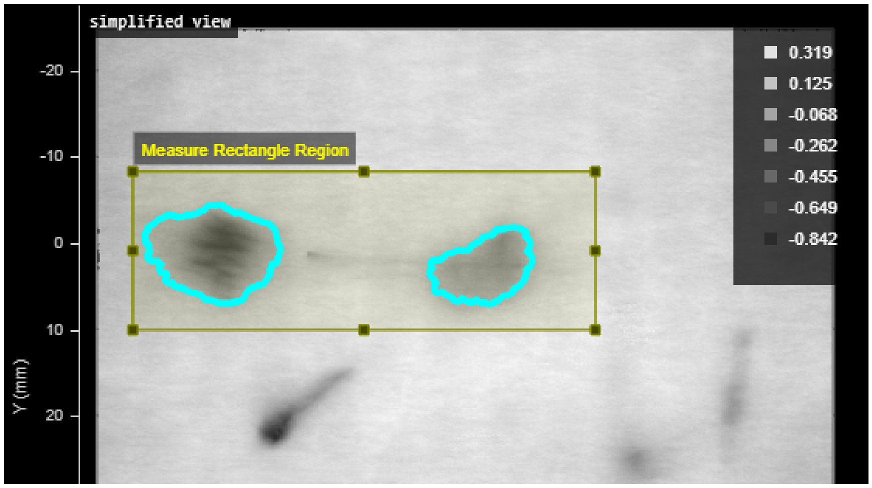
Type	Description
Center Point {n}	The point representing the center of a blob.

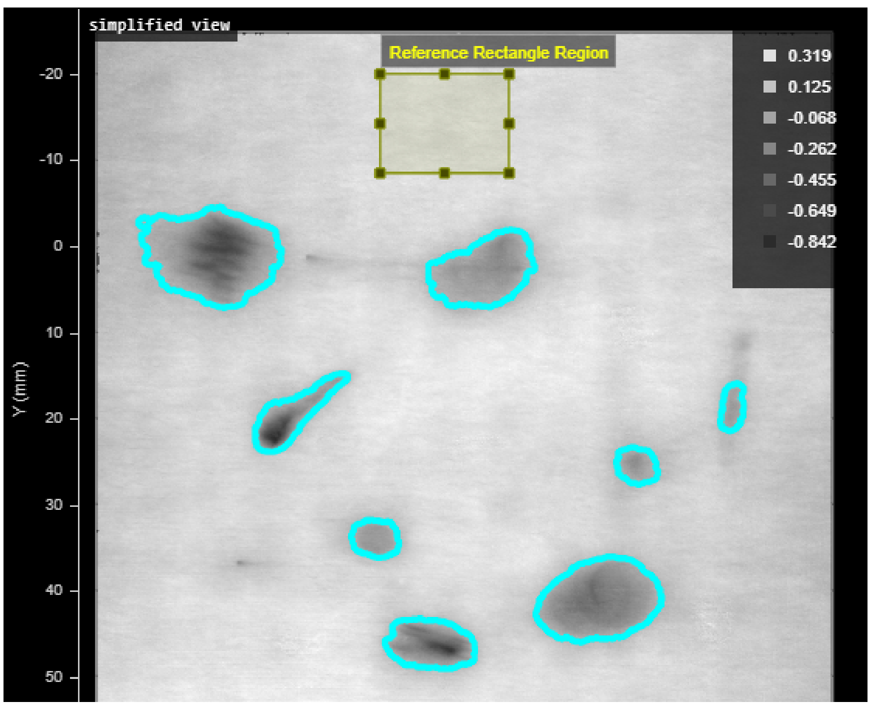
Data

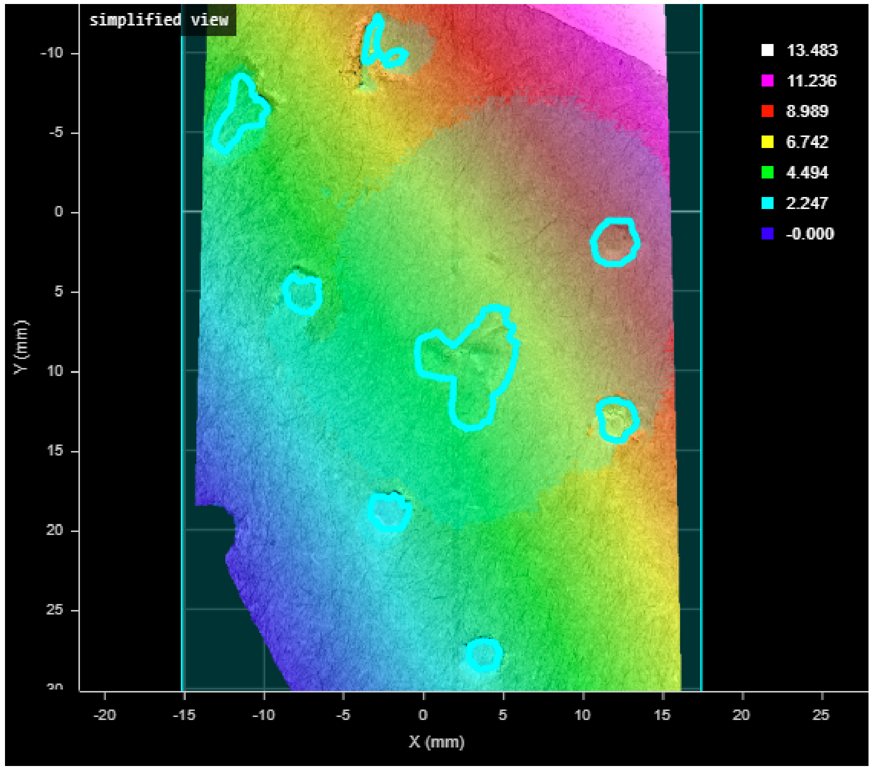
Type	Description
Blobs Array	An array containing the blobs. For an example of how to access this data from an SDK application or a GDK tool, see the appropriate sample in the SDK samples; for more information, see *Setup and Locations* on page 837.
Diagnostics Surface	Surface data you can use to evaluate the impact of the tool's parameters, before the tool's filters are applied, to properly separate the areas corresponding to the defects or targets you need to detect.
Surface {n}	Surface data corresponding to each blob.

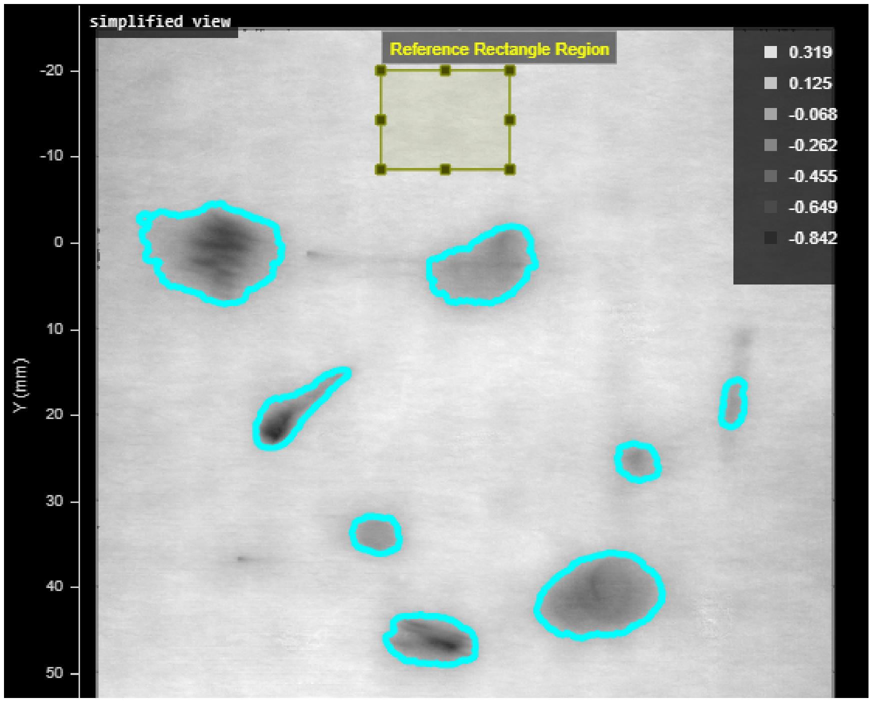
Parameters

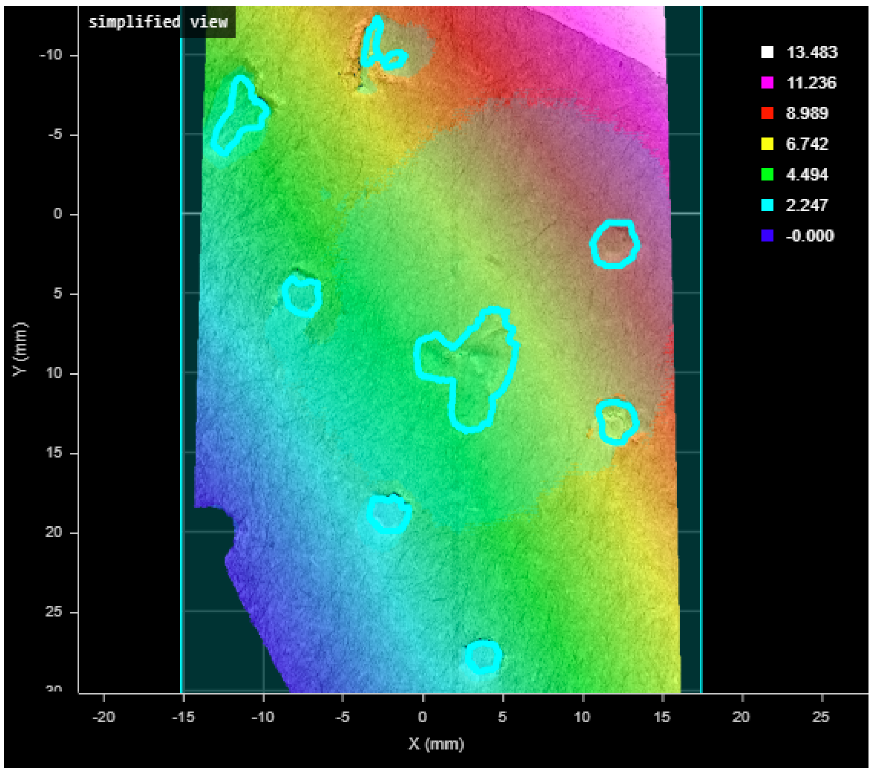
Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.

Parameter	Description
Use Intensity	If enabled, the tool uses intensity data instead of heightmap data. Only available if **Acquire Intensity** is enabled on the Scan page during the scan; for more information, see *Scan Modes* on page 88.
Use Measure Region	Indicates whether the tool limits blob detection to a user-defined region. If this option is not checked, the tool detects blobs in the entire active area. In the following, blobs are only detected in the rectangular measure region: 
Measure Region Type	When you enable **Use Measure Region**, the tool displays this and additional settings related to the type selected in this parameter. For details on flexible regions and their settings, see *Flexible Regions* on page 153. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 152.

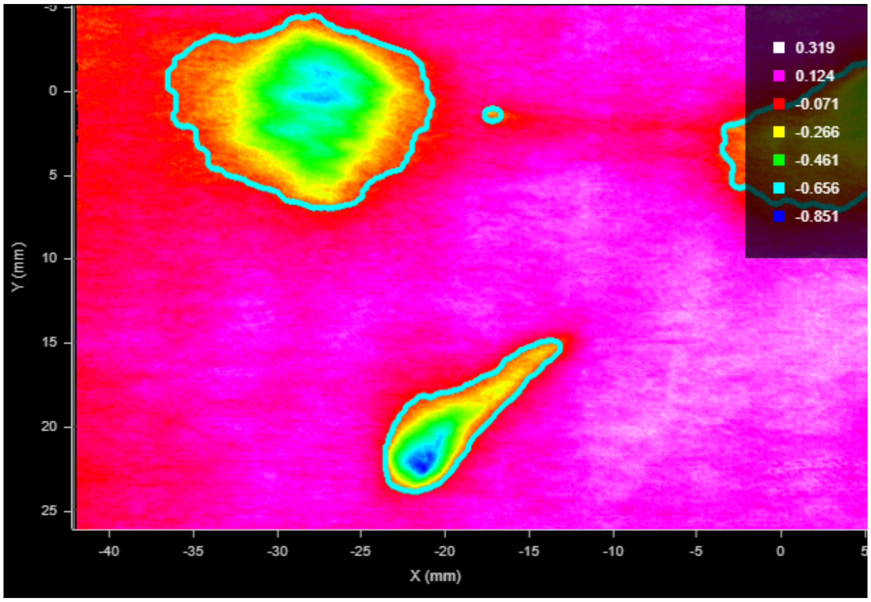
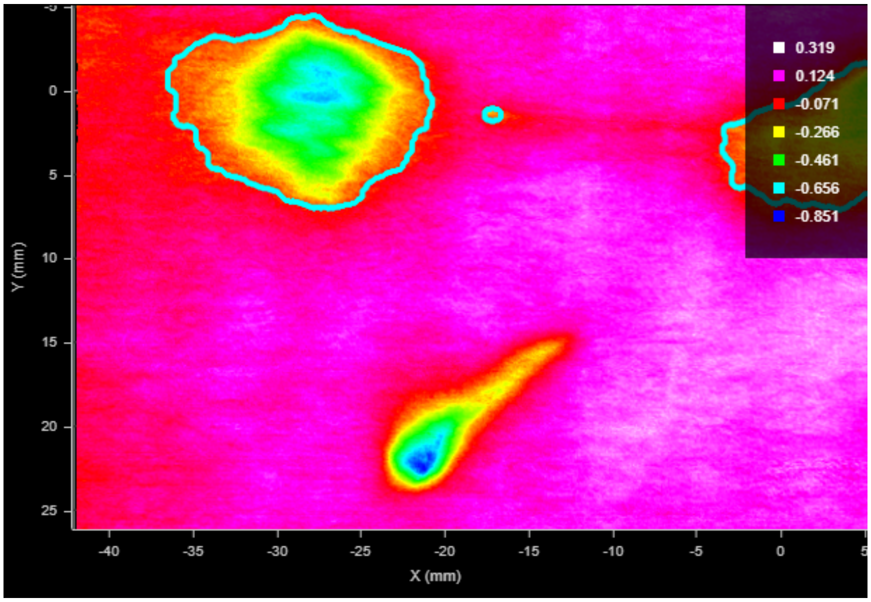
Parameter	Description
Reference Type	Provides three options: None, Reference Region, and Reference Plane. If the reference type is set to None, the **Height Threshold** setting is absolute (relative to zero). For the other options, see **Reference Region Type** and **Reference Plane** below. **None** If the reference type is set to None, the **Height Threshold** setting is absolute (relative to zero). **Reference Region** If the reference type is set to Reference Region, the tool uses the user-defined region specified in **Reference Region Type** (see below) to calculate an average height or intensity of the data in the reference region. In this case, **Height Threshold** and **Threshold Direction** are relative to this value. In the following, blobs are detected using a relative height threshold of -0.2 mm, relative to the average in the reference region:  **Reference Plane** If the reference type is set to Reference Plane, the tool uses the user-specified plane geometric feature specified in **Reference Plane** (see below) to correct for tilt of the target. In the following, note the gradient of the heightmap colors, indicating a height difference of roughly 9 millimeters between the lower and higher areas near the dents on the target's surface.

Parameter	Description
	

Parameter	Description
Reference Region Type	When you set **Reference Type** (see above) to Reference Region, the tool displays a drop-down that lets you choose the reference region type, as well as additional settings related to the type you select. (For details, see *Flexible Regions* on page 153.) The tool calculates an average height or intensity of the data in the reference region. **Height Threshold** is relative to this value. For example, in the following, blobs are detected using a relative height threshold of -0.2 mm, relative to the average in the reference region:  For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 152.

Parameter	Description
Reference Plane	If you set **Reference Type** (see above) to Reference Plane, the tool uses the plane geometric feature you choose in the **Reference Plane** drop-down to correct for the tilt of the target. Typically, you add and configure a Surface Plane tool to generate a plane (for more information, see *Plane* on page 409). For information on geometric features, see *Geometric Features* on page 164. For example, in the following, despite the overall tilt of the target, the tool detects the flaws on the surface. (Note the gradient of the heightmap colors, indicating a height difference of roughly 9 millimeters between the lower and higher areas near the dents on the target's surface.) 
Include Null Points	Indicates whether null points (points where no height or intensity value is available, due to dropouts or regions outside of the measurement range) are filled with the value in **Null Fill Value** as a general “background level” or to fill gaps to aid in isolating blobs. If **Use Intensity** is enabled, the value in **Null Fill Value** is an intensity.
Height Threshold Intensity Threshold	The threshold above or below which data is considered for being a blob. Use the **Threshold Direction** setting to determine whether data above or below the threshold is considered. If **Use Intensity** is enabled, this setting is named **Intensity Threshold**. Otherwise, it is named **Height Threshold**.

Parameter	Description
Open Kernel X Open Kernel Y	The X and Y kernel size, respectively, for morphological opening to remove small areas of data. Use these settings, for example, to remove bridges between areas to properly isolate them or to remove small areas entirely (perhaps caused by noise). Use different values of X and Y to use a non-rectangular filter to adapt the kernel to the kinds of unwanted data you see in the scan data.
Close Kernel X Close Kernel Y	The X and Y kernel size, respectively, for morphological closing to fill in holes smaller than the specified kernel size. Use these settings, for example, to fill small areas within potential blobs that may be caused by drop-outs. Use different values of X and Y to use a non-rectangular filter to adapt the kernel to the kinds of holes you see in the scan data.
Hierarchy	Provides options to let you find either external blobs only or both external and internal blobs. **External Blobs** Use this option to ignore smaller blobs in larger blogs: only the outermost blog is returned. **External + Internal Blobs** Use this option to include smaller blobs in larger blobs.
Use Area Filter Max Area Min Area	If **Use Area Filter** is enabled, the tool applies an area filter to potential blobs using the values in **Max Area** and **Min Area**.

Parameter	Description
Use Aspect Filter Max Aspect Min Aspect	If **Use Aspect Filter** is enabled, the tool applies an aspect filter (ratio of length and width) to the rotated bounding box that would encapsulate the area, using the values in **Max Aspect** and **Min Aspect**. For example, the following dent in a surface is included as a blob if these aspect values are set to 1 and 0.354, respectively (the rotated bounding box encapsulating would be 13.059 mm x 4.704 mm).  In the following, the same dent is excluded if **Min Aspect** is set to a value greater than 0.354. 

Parameter	Description
Use Circularity Filter Max Circularity Min Circularity	If **Use Circularity Filter** is enabled, the tool applies a circularity filter to potential blobs to measure how close to a circle the blob is, using the values in **Max Circularity** and **Min Circularity** . Circularity is determined from area within the contour of the blob and the perimeter of its contour. With increasing perimeter for the same area, circularity is reduced.
Use Convexity Filter Max Convexity Min Convexity	If **Use Convexity Filter** is enabled, the tool applies a convexity filter to potential blobs, using the values in **Max Convexity** and **Min Convexity** . Convexity is defined as the (Area of the Blob / Area of its convex hull), and "convex hull" of a shape is the tightest convex shape that completely encloses the shape.
Ordering	Orders the measurements, features, and surface data of the individual blobs output by the tool. Choose one of the following: • Area - Large to small • Area - Small to large • Position - X increasing • Position - X decreasing • Position - Y increasing • Position - Y decreasing
Number of Blob Outputs	Determines the number of blobs the tool outputs as measurements, features (center points of blobs), and surface data. Currently limited to 200 blobs.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

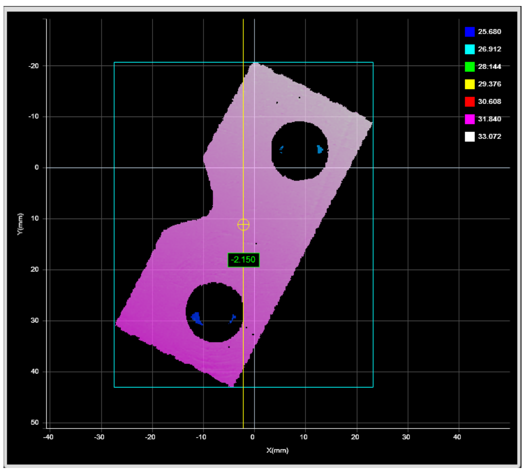
Bounding Box

The Bounding Box tool provides measurements related to the smallest box that contains the *scan data* from a part (for example, X position, Y position, width, length, etc.).

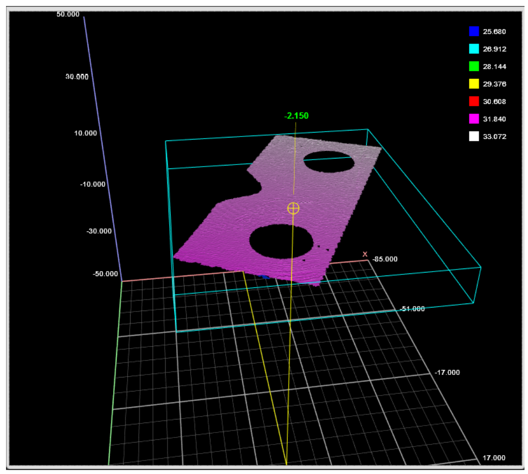
> If you need to measure the height of the target relative to the Z = 0 reference (such as if you want to measure the height of a box or other container), use the Surface Bounding Box Advanced tool; for more information, see See *Bounding Box Advanced* on page 306.

A bounding box can be vertical or rotated. A vertical bounding box provides the absolute position from which the Position centroids tools are referenced.

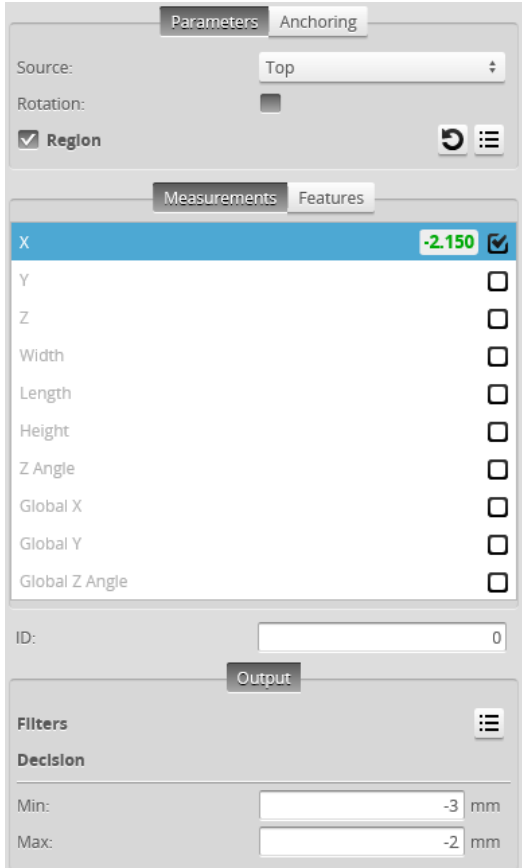
> The vertical bounding box X and Y correspond to the part frame of reference origin. For this reason all X and Y measurements (except Bounding Box Global X and Global Y) are referenced to this point when **Frame of Reference** on the **Part Detection** panel is set to **Part**. See *Part Detection* on page 111 for more information.



2D View



3D View

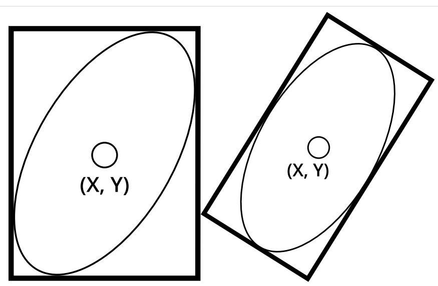


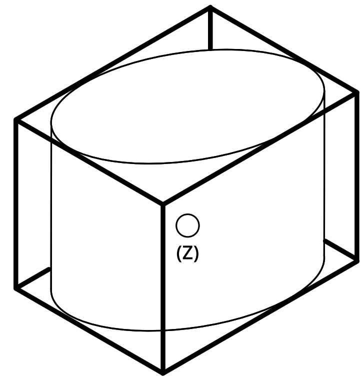
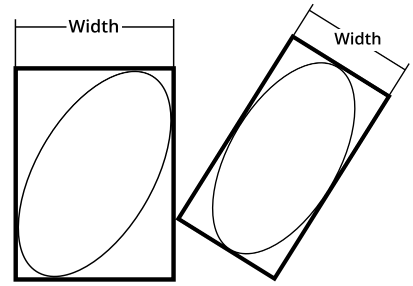
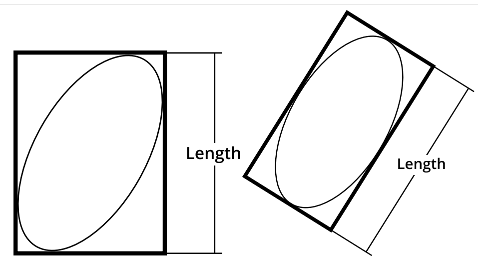
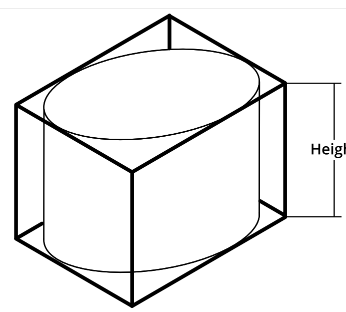
Measurement Panel

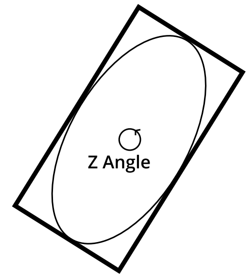
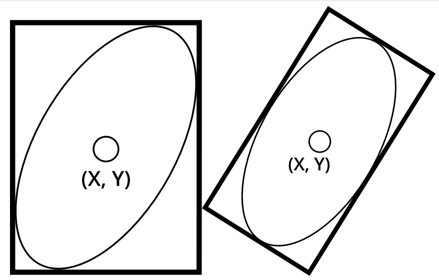
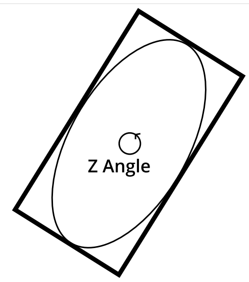
For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Determines the X position of the center of the bounding box that contains the part. The value returned is relative to the *part*.	
Y Determines the Y position of the center of the bounding box that contains the part. The value returned is relative to the *part*.	

Measurement	Illustration
Z Determines the Z position of the center of the bounding box that contains the part. The value returned is relative to the *part*.	
Width Determines the width of the bounding box that contains the part. When the **Rotation** setting is disabled, the bounding box is the smallest rectangle whose sides are parallel to the X and Y axes. Width is on the X axis. When **Rotation** is enabled, the width is the smaller side dimension.	
Length Determines the length of the bounding box that contains the part. When the **Rotation** setting is disabled, the bounding box is the smallest rectangle whose sides are parallel to the X and Y axes. Length is on the Y axis. When **Rotation** is enabled, the length is the longer side dimension.	
Height Determines the height of the bounding box that contains the part.	

Measurement	Illustration
Z Angle Determines the rotation around the Z axis and the angle of the longer side of the bounding box relative to the X axis. If **Rotation** is not enabled, the measurement returns 90.000 degrees. In order to use this measurement for angle anchoring, you must enable **Rotation**; for more information on anchoring, see *Measurement Anchoring* on page 169.	
Global X* Determines the X position of the center of the bounding box that contains the part *on the surface from which the part was extracted*.	
Global Y* Determines the Y position of the center of the bounding box that contains the part *on the surface from which the part was extracted*.	
Global Z Angle* Determines the rotation of the longer side of the bounding box around the Z axis *on the surface from which the part was extracted*. If part matching is enabled, the returned value represents the rotation of the part *before* part matching rotates it. If **Rotation** is not enabled, the measurement returns 90.000 degrees.	

*These measurements are mostly useful with parts extracted from a surface. For more information on parts, see *Part Detection* on page 111.

Features

Type	Description
Center Point	The center point of the bounding box.
Box Axis Line	The axis of the bounding box.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.

Parameter	Description
Rotation	A bounding box can be vertical or rotated. A vertical bounding box provides the absolute position from which the part's Position centroid measurements are referenced. Check the **Rotation** setting to select rotated bounding box.
Asymmetry Detection	Resolves the orientation of an object over 360 degrees. The possible values are: 0 – None 1 – Along Major Axis 2 – Along Minor Axis This setting is only visible if **Rotation** is checked.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

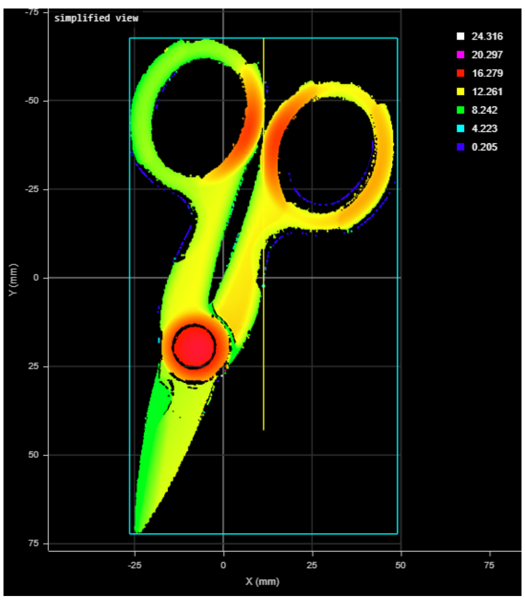
Bounding Box Advanced

> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

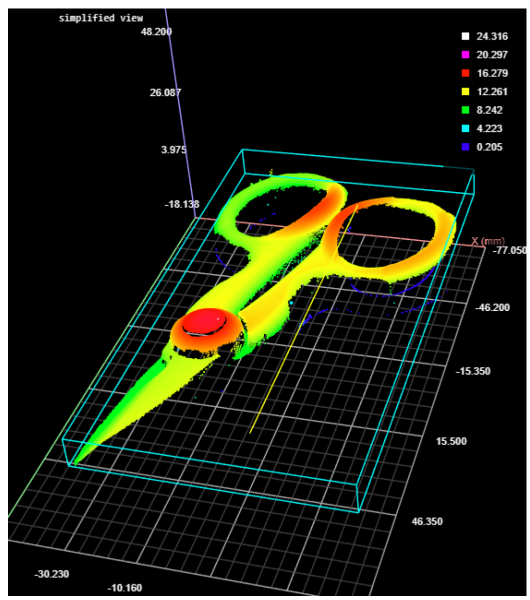
Like the Bounding Box tool (see *Bounding Box* on page 301), the Bounding Box Advanced tool provides measurements related to the smallest box that contains the *scan data* from a part (for example, X position, Y position, width, length, etc.). However, this version of the tool also lets you get the height of bounding box relative to the Z origin (typically the conveyor on which the target is sitting). This lets you determine, for example, the height of a box or other container on the conveyor as part of a product packaging process. New settings also let you easily filter out noise that can affect height, width, and length measurements.

A bounding box can be vertical or rotated. A vertical bounding box provides the absolute position from which the Position centroids tools are referenced.

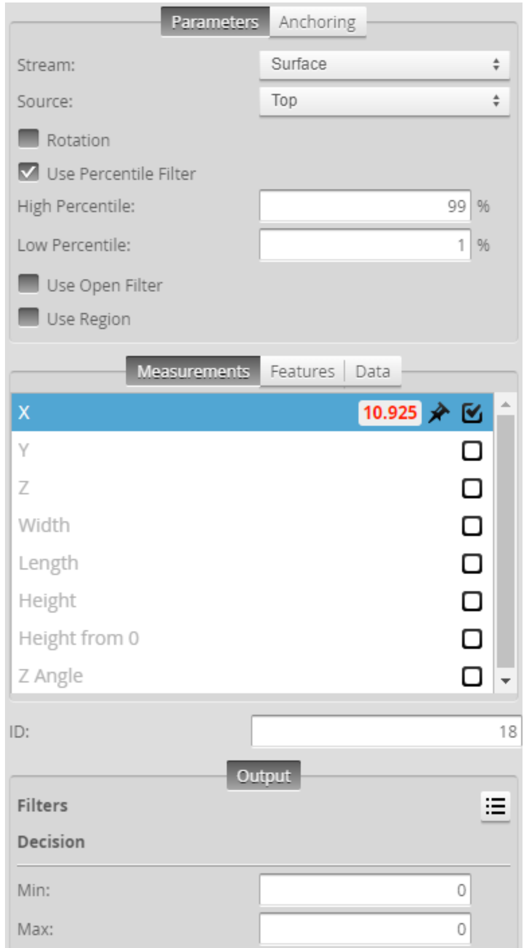
> The vertical bounding box X and Y correspond to the part frame of reference origin. For this reason all X and Y measurements (except Bounding Box Global X and Global Y) are referenced to this point when **Frame of Reference** on the **Part Detection** panel is set to **Part**. See *Part Detection* on page 111 for more information.



2D View



3D View

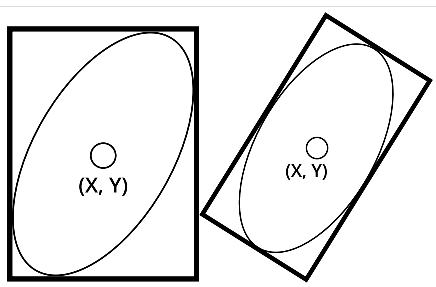


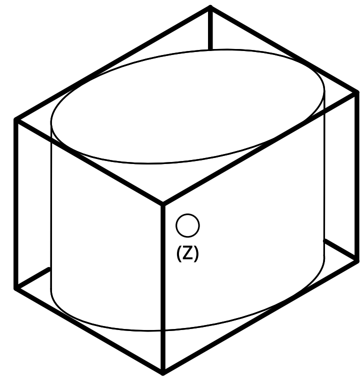
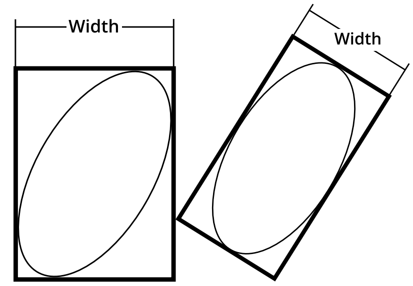
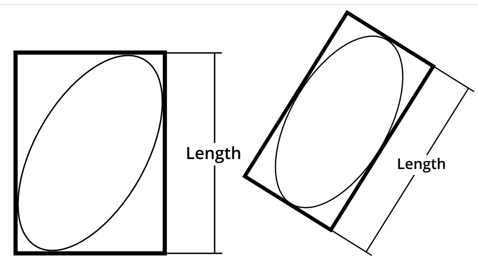
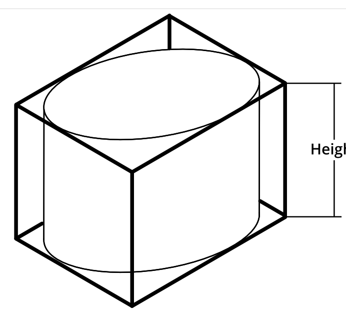
Measurement Panel

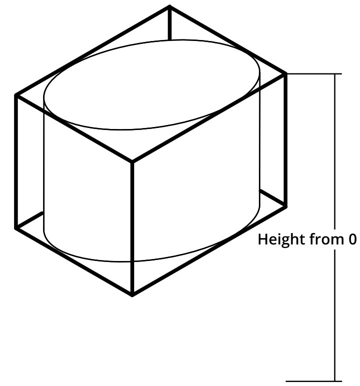
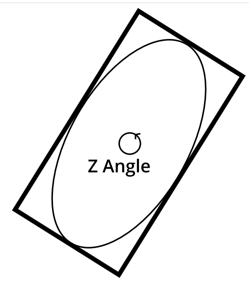
For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Determines the X position of the center of the bounding box that contains the part. The value returned is relative to the *part*.	
Y Determines the Y position of the center of the bounding box that contains the part. The value returned is relative to the *part*.	

Measurement	Illustration
Z Determines the Z position of the center of the bounding box that contains the part. The value returned is relative to the *part*.	
Width Determines the width of the bounding box that contains the part. When the **Rotation** setting is disabled, the bounding box is the smallest rectangle whose sides are parallel to the X and Y axes. Width is on the X axis. When **Rotation** is enabled, the width is the smaller side dimension.	
Length Determines the length of the bounding box that contains the part. When the **Rotation** setting is disabled, the bounding box is the smallest rectangle whose sides are parallel to the X and Y axes. Length is on the Y axis. When **Rotation** is enabled, the length is the longer side dimension.	
Height Determines the height of the bounding box that contains the part.	

Measurement	Illustration
Height from 0 Determines the distance from the top of the bounding box to the Z origin (Z = 0).	
Z Angle Determines the rotation around the Z axis and the angle of the longer side of the bounding box relative to the X axis. If **Rotation** is not enabled, the measurement returns 90.000 degrees. In order to use this measurement for angle anchoring, you must enable **Rotation**; for more information on anchoring, see *Measurement Anchoring* on page 169.	

Features

Type	Description
Center	The center point of the bounding box.

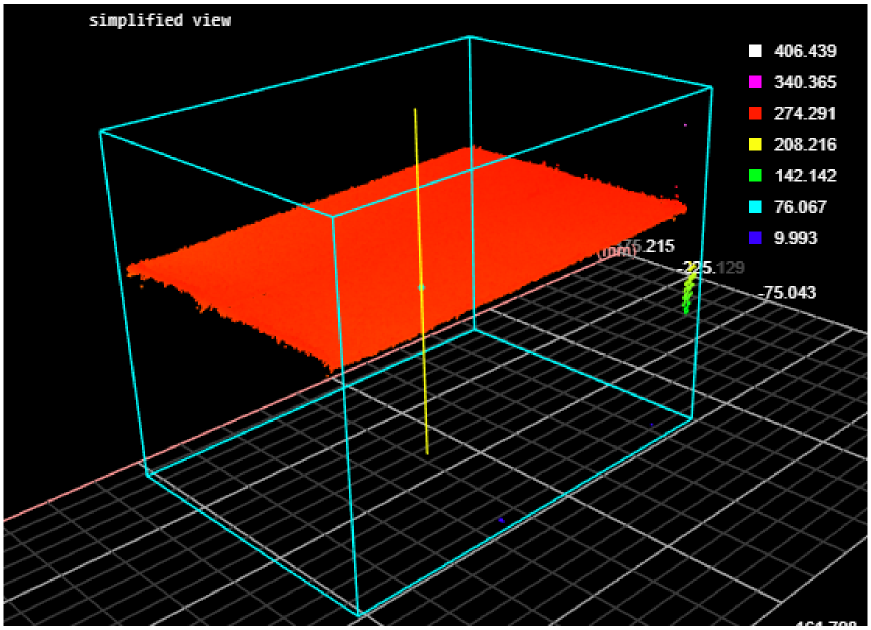
For more information on geometric features, see *Geometric Features* on page 164.

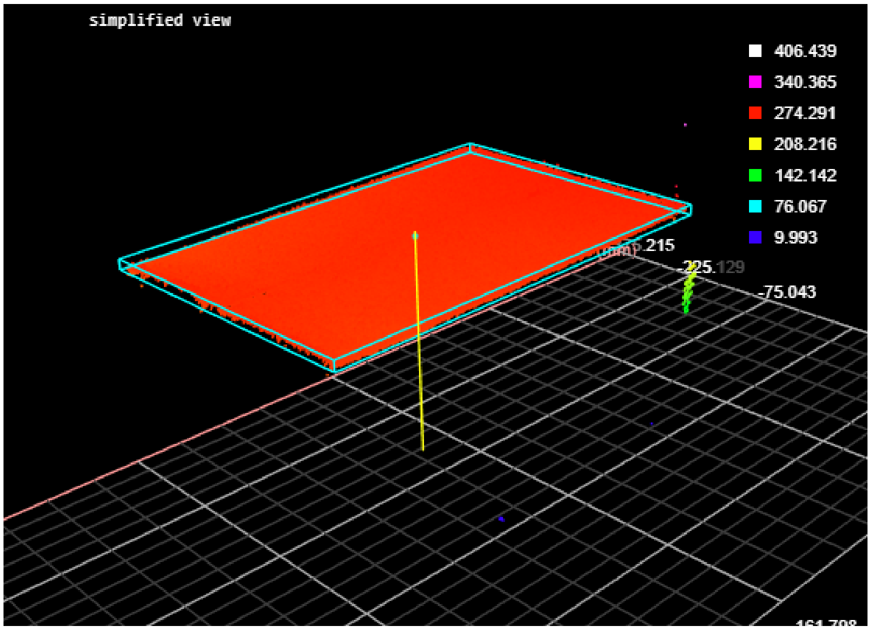
Data

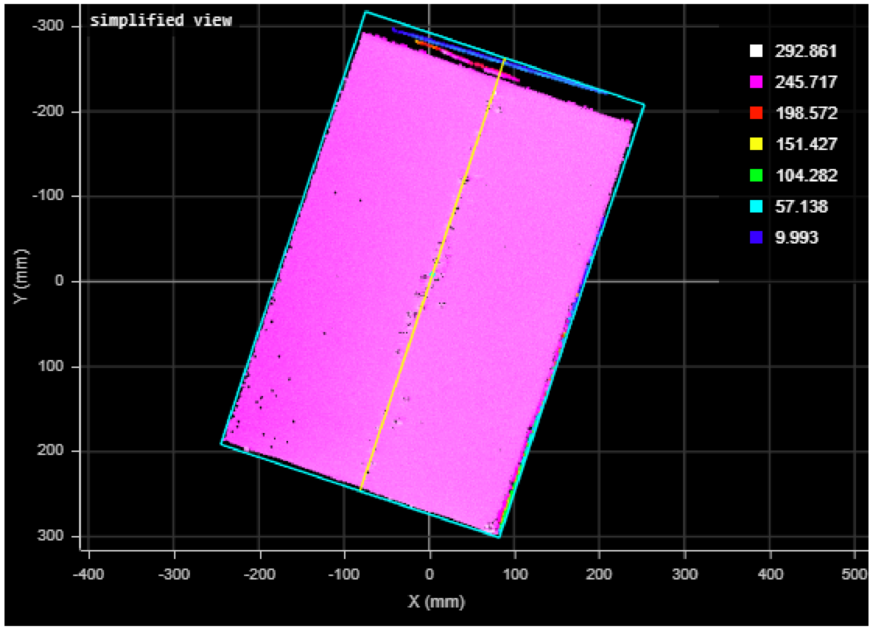
Type	Description
Diagnostics Surface	A surface useful for evaluating the impact of the open filter. For more information, see *Use Open Filter* on page 312.

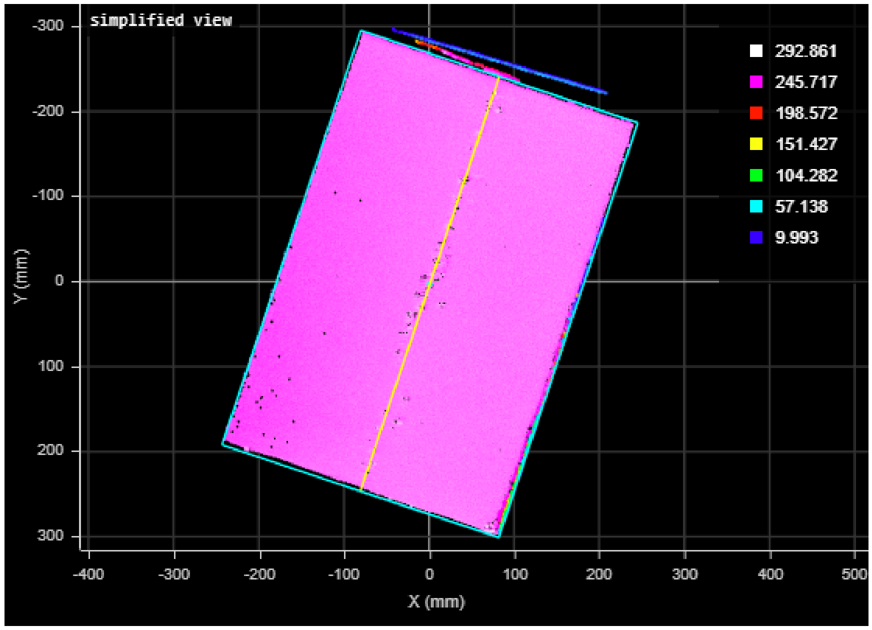
Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Rotation	A bounding box can be vertical or rotated. A vertical bounding box provides the absolute position from which the part's Position centroid measurements are referenced. Check the **Rotation** setting to select rotated bounding box.

Parameter	Description
Use Percentile Filter	Limits the bounding box to data points along the Z axis between the values you set in **High Percentile** and **Low Percentile**, which are displayed when you choose this option. Use this setting to obtain more "robust" height measurements. This setting is useful to exclude noise that would otherwise cause inaccurate height measurements. For example, in the following scan of a box, without excluding a small percentage of the highest data points, data points caused by noise to the upper right produces an inaccurate height measurement of the box of 406.457 mm.  When **High Percentile** is set to 99%, the highest 1 percent of data points is excluded from the placement of the bounding box, and an accurate height of the target box of 270.477 mm is returned.

Parameter	Description
	

Parameter	Description
Use Open Filter	When enabled, this setting lets you set the value of **Kernel Size** for an *open* morphological operation applied to the scan data on the XY plane, letting you achieve "robust" width and length measurements. This filter removes noise or small objects from scan data, while keeping the shape and size of the larger objects in the scan data. For example, in the following, noise along the edge at the top of the data viewer results in an inaccurate length measurement.  When the filter is set to an appropriately sized kernel (here, 11 points), the noise is excluded from the calculation of the bounding box, and an accurate length is returned.

Parameter	Description
	 Use the Diagnostics Surface on the **Data** tab to evaluate the impact of the open filter, to avoid removing too much data.
Use Region	When enabled, displays additional settings to let you set a region (see below).
Region Type Inner Circle Diameter Inner Ellipse Major Axis Inner Ellipse Minor Axis Sector Start Angle Sector Angle Range Mask Source Low Threshold High Threshold	When you enable **Use Region**, the tool displays additional settings related to the measure region type. For details on flexible regions and their settings, see *Flexible Regions* on page 153. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

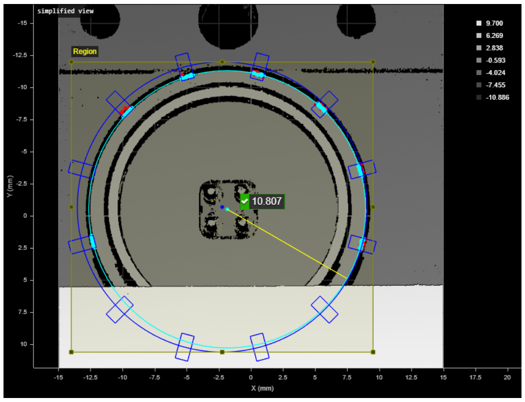
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

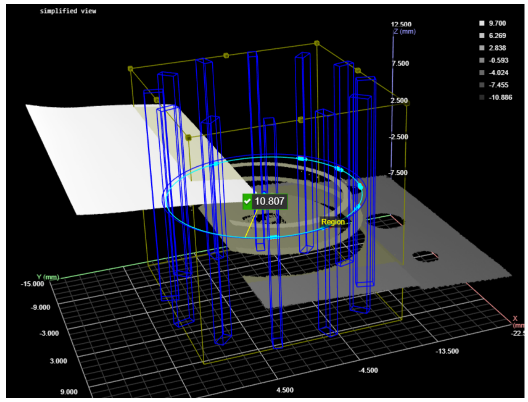
Circular Edge

The Circular Edge tool fits a circle to a circular edge in the scan data, using either height map or intensity data. The edge can be the outer edge of a disc-like feature or the inner edge of a hole. The tool can optionally work with partial data, as little as 1/4 of a circle, letting it work with rounded corners.

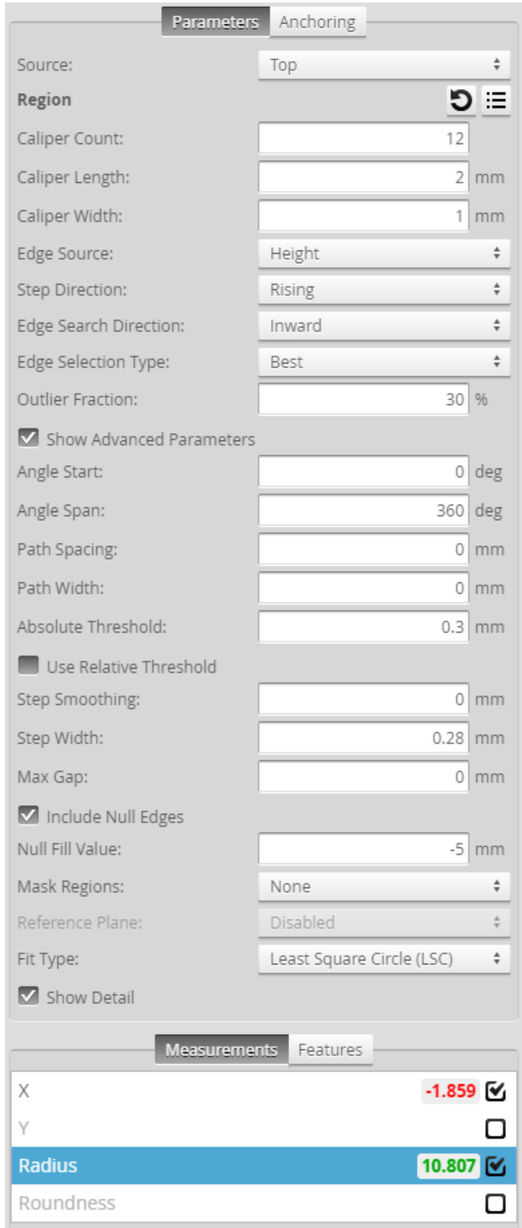
The tool lets you measure the position and radius of the circular feature and determine its roundness error. The feature is expected to be relatively round and not, for example, ovoid. In the following images, the outer edge of a circular feature is measured. The same tool could just as easily measure the characteristics of one of the holes at the top.



2D View



3D View



Measurement Panel

The tool uses one of four standard methods to calculate roundness. The choice of method affects the other measurements.

- Least Square Circle (LSC)
- Minimum Zone Circle (MZC)
- Maximum Inscribed Circle (MIC)
- Minimum Circumscribed Circle (MCC)

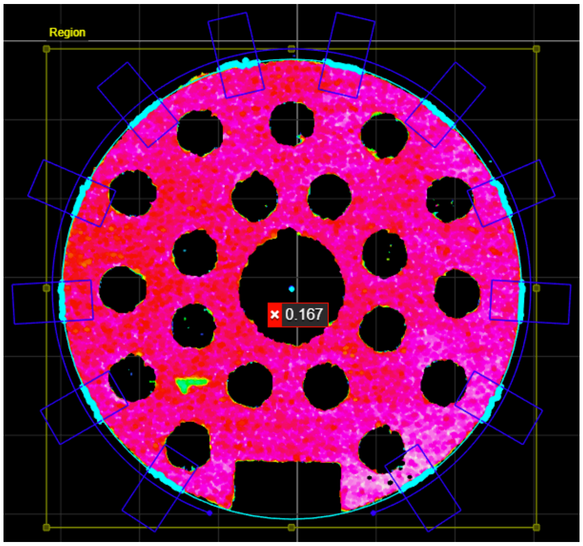
The tool can also generate circle and center point geometric features that Feature tools can take as input for measurement. For more information on Feature tools, see *Feature Measurement* on page 484.

Some of the tool parameters are hidden unless **Show Advanced Parameters** is checked.

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Calipers, Extracted Paths, and Edge Points

To fit a circle to the scan data, the Surface Circular Edge tool starts by overlaying evenly spaced calipers along a circular path constrained by the region of interest.



Rectangular calipers (dark blue) placed along circular path (dark blue), constrained by the region

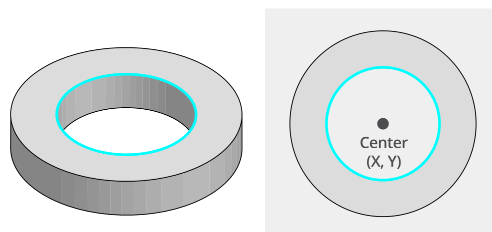
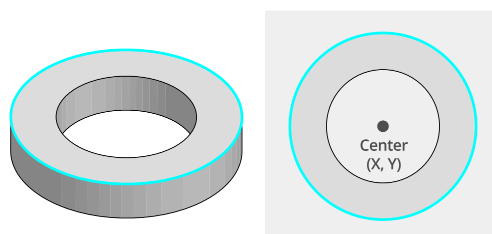
The circular path can optionally be partial, and starts at a defined orientation around the Z axis. The circular path can be as short as 1/4 of a circle, letting it work with rounded corners. Calipers extend vertically to fill the entire region of interest.

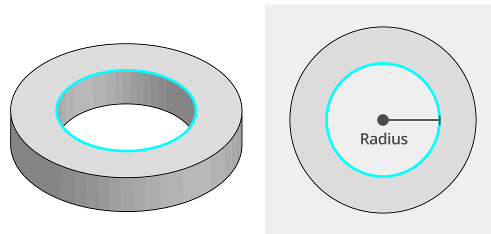
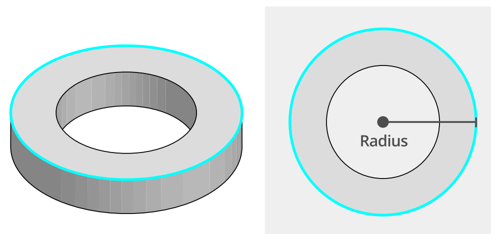
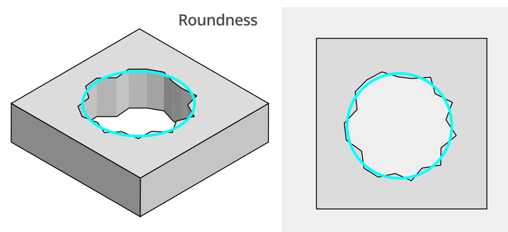
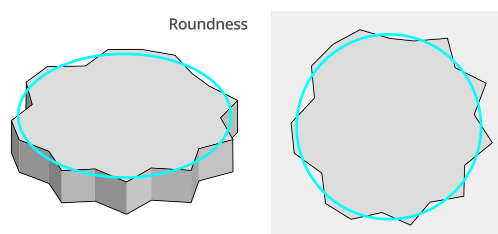
Internally, the tool extracts profiles from the data within each caliper, running from the end of the caliper closest to the center of the tool's region of interest to the end farthest from the center. The tool then searches for steps in each profile that meet the criteria set by the tool’s settings, such as minimum height, direction (whether it is rising or falling), and so on.

The tool places an edge point on each selected step. The tool then uses the edge points in all the calipers to fit a circle: the various characteristics of the fitted circle are then returned as measurements.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X **Y** Returns the X and Y position of the center of the fitted circle, respectively.	 

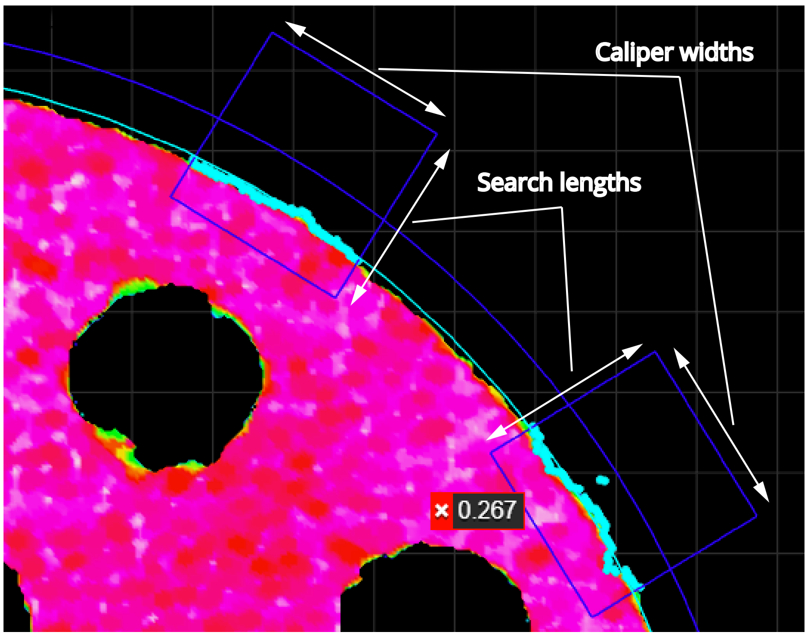
Measurement	Illustration
Radius Returns the radius of the fitted circle.	 
Roundness Returns the roundness or circularity of the edge points with respect to the reference circle of the selected roundness error method set in **Fit Type**.	 
Min Error **Max Error** These measurements return information on the points furthest inside and outside the fitted circle, respectively.	

Features

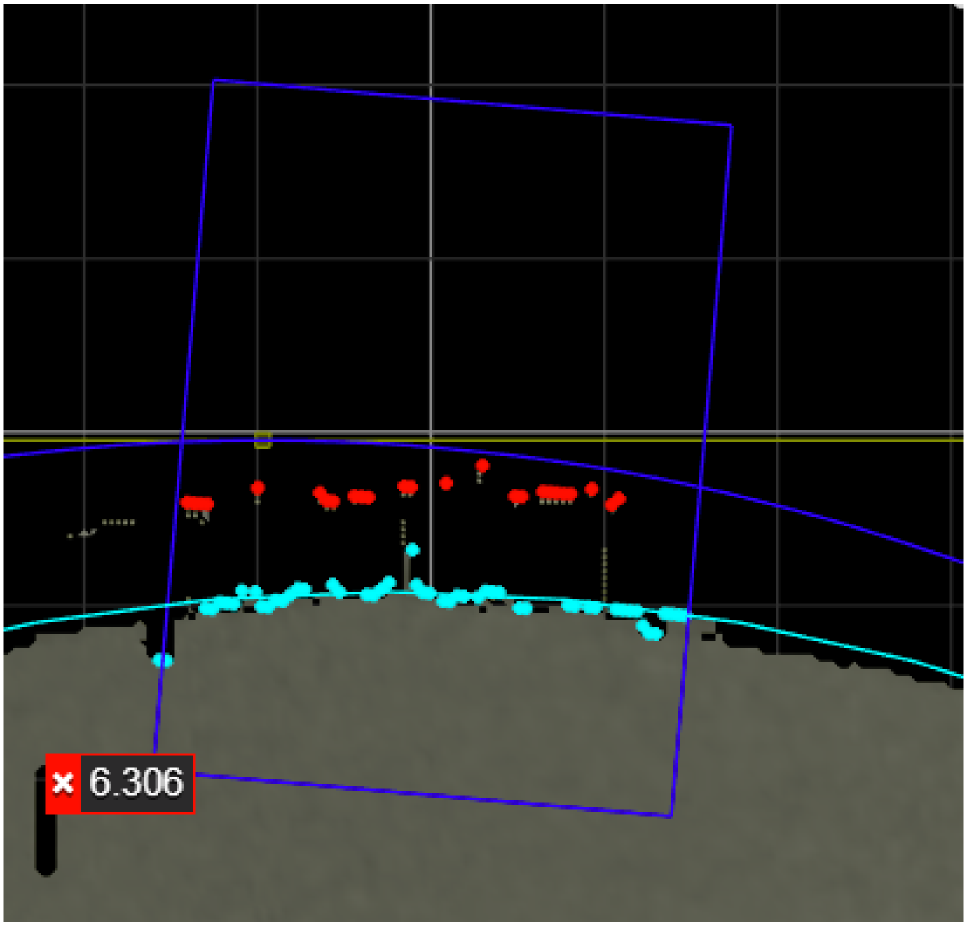
Type	Description
Center	The center of the fitted circle.
Circle	The fitted circle.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

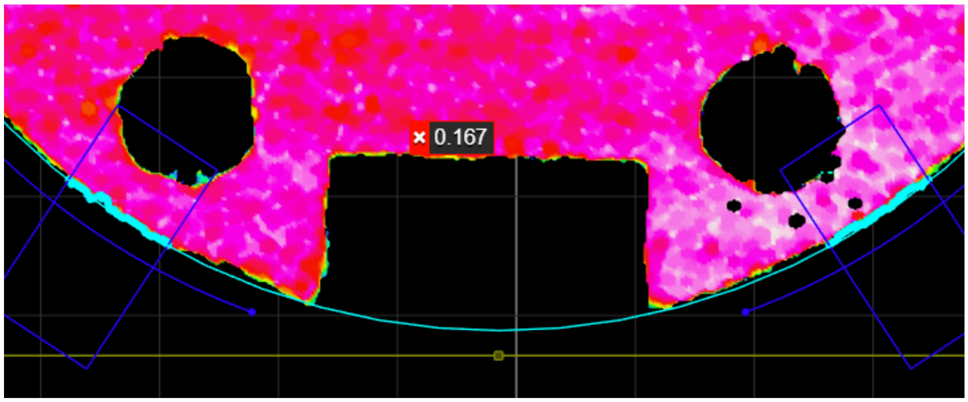
Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152. The region also constrains the circular path along which the tool places the calipers.
Caliper Count	The number of calipers the tool places along the circular path. Using a higher number of calipers increases the amount of data available to the tool, but also increases the amount of time the tool takes to run. Choose a balance between the runtime of the tool and the number of calipers needed to get enough edge points to properly fit the circle to the scan data.
Caliper Length Caliper Width	**Caliper Length** is the length of the calipers (extending perpendicular to a tangent on the circular caliper path, centered on the path). The length of the calipers determines the length of the extracted profiles the tool examines for steps. Longer calipers increase the amount of data the tool must analyze and therefore the time the tool takes to run; longer calipers can also include unwanted steps when the tool searches for the edge. **Caliper Width** is the width of the calipers (extending parallel to a tangent on the circular path). A wider caliper increases the time the tool takes to run. It does however increase the number of edge points, which may help the tool fit the circle. 
Edge Source	Specifies the type of data the tool uses. Either Height or Intensity. Use intensity data as the edge source when contrast differences on a flat area of a target, which would not be detected using height map data, are distinct, letting the tool use the detected edge to fit the circle.

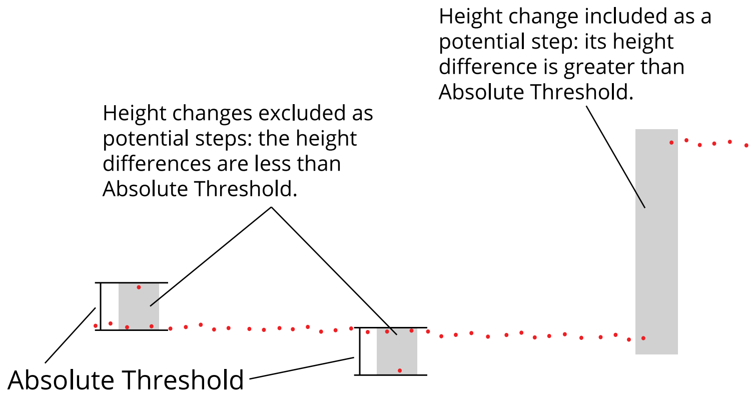
Parameter	Description
Step Direction	Determines whether the expected step in the data rises or falls, or moves from valid to null or null to valid. Note that this setting depends on the **Edge Search Direction** setting for its interpretation of what "rises" and "falls." One of the following: **Rising & Falling**: Searches for edge points on rising or falling edges. **Rising**: Searches for edge points only on rising edges. **Falling**: Searches for edge points only on falling edges.
Edge Search Direction	Specifies the search direction along the calipers. Either Inward (toward the center of the region of interest) or Outward.
Edge Selection Type	Determines which step the tool uses on each of the profiles internally extracted from the calipers when there are multiple steps. An edge point is placed on each chosen step, and is used to fit the circle. Steps must pass the criteria of the tool's settings, such as threshold and outlier exclusion. **Best**: Selects the greatest step in the search direction on each profile. **First**: Selects the first step in the search direction on each profile. **Last**: Selects the last step in the search direction on each profile.

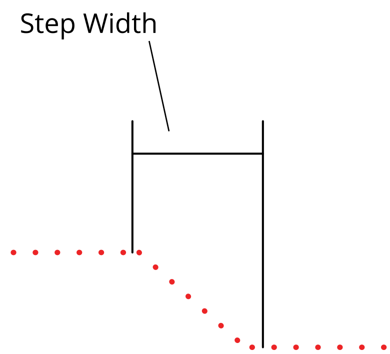
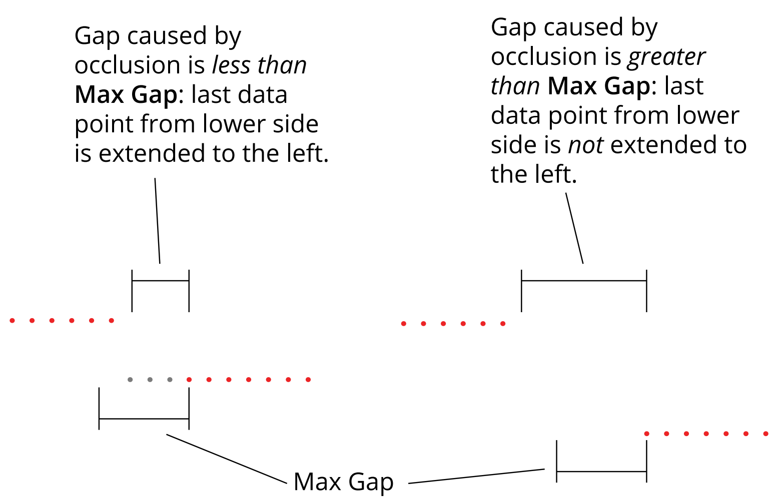
Parameter	Description
Outlier Fraction	The percentage of outlier points to exclude. Setting this to a small value can help the tool fit the circle better to the edge.  *__Outlier Fraction__ set to a low value: rejected outlier edge points are red.*
Show Advanced Parameters	When enabled, displays advanced settings. Note that most of these settings are applied *even when they are hidden*. For information on these settings, see *Advanced Parameters* on the next page.
Show Detail	When disabled, hides the calipers and caliper path, as well as the edge points.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

The following parameters are hidden when **Show Advanced Parameters** is unchecked. All advanced parameters, *except* **Reference Plane**, are applied when they are hidden. Mask regions are not rendered, even though they are applied.

Advanced Parameters

Parameter	Description
Angle Start Angle Span	These settings work together to let you set a partial path and exclude part of the data. In the following close-up image of a circular feature, the dark blue path starts to the right of the notch, continues counter-clockwise around the circular feature, and ends to the left of it.  **Angle Start** is the starting angle, around the Z axis on the XY plane, for the circular path along which calipers are placed. Setting this to 0 aligns the start angle with the positive direction of the X axis. **Angle Span** is the length of the circular path along which calipers are placed.
Path Spacing	Sets the spacing between paths in the calipers used to extract the profiles that determine the edge. A higher number of paths results in a higher number of edge points, which makes the fitting of the edge line more accurate. However, a higher number of edge points results in a greater tool execution time. When **Path Spacing** is set to 0, the resolution of the scan data is used as the basis for spacing.
Path Width	The size of the windows perpendicular to the path used to calculate an average for each data point on a path profile. Useful to average out noise along the path caused by reflections, and so on. If **Path Width** is set to 0, no averaging is performed (only the data point under the path is used). For averaging along the path, use **Step Smoothing** (see below).

Parameter	Description
Absolute Threshold	When **Use Intensity** is disabled, the setting specifies the minimum *height* difference between points on a path profile for that step to be considered for an edge point. The setting can be used to exclude smaller steps on a part that should not be considered for an edge, or to exclude height differences caused by noise. When used in conjunction with **Relative Threshold**, **Absolute Threshold** is typically set to a small value, greater than the general surface roughness.  When **Use Intensity** is enabled, the setting specifies the minimum difference in intensity. (**Acquire Intensity** must enabled in the **Scan Mode** panel.)
Use Relative Threshold	When this option is enabled, the **Relative Threshold** field is displayed.
Relative Threshold	The value for the relative threshold. The tool calculates a relative threshold by scaling the greatest height or intensity difference found on the path profiles by the percentage in **Relative Threshold**. This lets you configure the tool without knowing the actual step height in advance, and is useful for edges with varying step height. For a height or intensity difference to be considered a valid step, both **Absolute Threshold** and **Relative Threshold** must pass.
Step Smoothing	The size of the windows along the path used to calculate an average for each data point on a path profile. The setting is useful for averaging out noise. If **Step Smoothing** is set to 0, no averaging is performed (only the data point under the path is used). For averaging perpendicular to the path, use **Path Width** (see above).

Parameter	Description
Step Width	The distance, along a path profile, separating the points used to find steps on a path profile.  The setting is useful when you must detect a slope as an edge, rather than a sharply defined edge: setting **Step Width** to a value greater than the width of the edge ensures that the tool measures the height difference between the flat regions on either side of the edge. As a result, the height of the step is accurately measured, and the edge is correctly located. Setting **Step Width** wider than necessary can reduce the precision of edge location.
Max Gap	Fills in regions of missing data caused by an occlusion near the desired edge. Use this setting when continuity on the target is expected. When **Max Gap** is set to a non-zero value, the tool holds and extends the last data point on the low side next to an edge across a gap of null points, up to the distance specified in **Max Gap**. 

Parameter	Description
Include Null Edges	Indicates whether null points (points where no height or intensity value is available, due to dropouts or regions outside of the measurement range) are filled with the value in **Null Fill Value** as a general "background level." If **Use Intensity** (see above) is enabled, the intensity value in **Intensity Null Fill Value** is also used. A typical example is a discrete part produced by part detection of an object sitting on a flat background. The background is not visible in the part, so the tool assumes that any null region are at the background level. To find edges along a region of null points, you must use either this option and an appropriate value in **Null Fill Value** (and **Intensity Null Fill Value** if **Use Intensity** is enabled) or **Max Gap**. Otherwise, only edges within areas of contiguous data will be detected.
Null Fill Value	The height value (in mm) used to replace null points not filled by **Max Gap** when **Include Null Edges** is enabled.
Intensity Null Fill Value	The intensity value (0-255) used to replace null points when **Include Null Edges** and **Use Intensity** are enabled.
Mask Regions	Lets you enable up to five regions that you can use to mask data you want the tool to ignore. You can resize and reposition the mask regions using the mouse in the data viewer, or by configuring values manually in the **Mask Region** sections the tool displays in the tool settings for each region. You can only set the rotation of the mask regions manually by modifying the region's **Z angle** parameter. By default, when you add multiple mask regions, they are initially placed in the same position, one on top of the other.
Reference Plane	Uses the output of a Surface Plane tool as a reference plane. Useful to correct the scan data if the target is slightly tilted. When **Show Advanced Parameters** is unchecked and **Reference Plane** is set to a plane, the plane is *ignored*.

Parameter	Description
Fit Type	The method the tool uses to calculate the roundness of the feature. One of the following: **Least Square Circle (LSC)** **Minimum Zone Circle (MZC)**: If you choose this method, set the circle the tool uses with the **Which Circle** parameter. **Maximum Inscribed Circle (MIC)**: Typically used to measure the inner edge of a circular feature, such as a hole. **Minimum Circumscribed Circle (MCC)**: Typically used to measure the outer edge of a circular feature. If you load a job that contains an instance of the Circular Edge tool created using an earlier firmware version, an additional parameter (**LSC FIt Method**) is displayed. It provides two options: **Least Square Method**: This algorithm provides more accurate fit results than Iterative Approximation on partial circle data. The execution time is also better on average, so this method should be chosen in general. **Iterative Approximation**: Legacy algorithm for compatibility with 5.2 SR2 and earlier.
Which Circle	Tells the tool which circle (Inner or Outer) to use when Minimum Zone Circle is the fit method in **Fit Type**.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

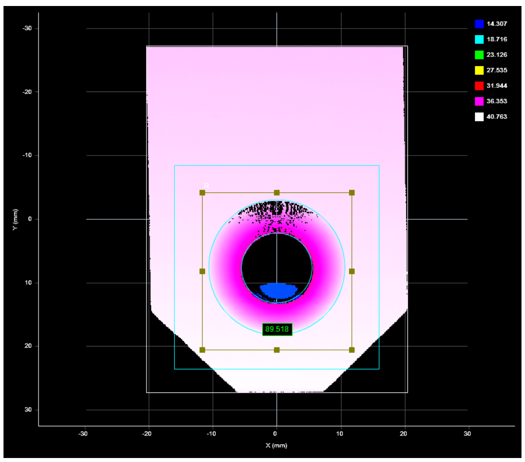
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

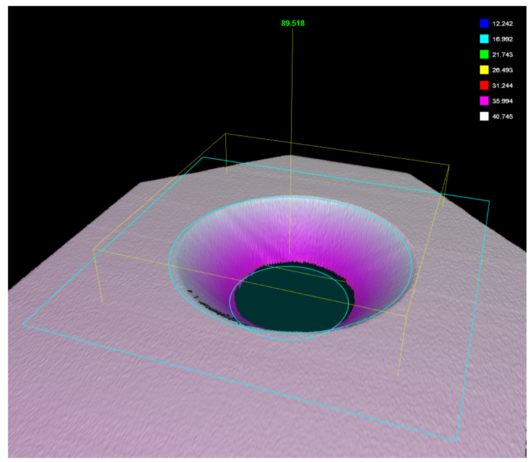
Countersunk Hole

The Countersunk Hole tool locates a countersunk circular opening within a region of interest on the surface and provides measurements to evaluate characteristics of countersunk holes, including the position (X, Y, and Z) of the center of the hole, outside radius of the hole, hole bevel angle, and the depth of the hole. The countersunk hole can be on a surface at an angle to the sensor. The tool also supports measuring holes drilled at an angle relative to the surrounding surface.

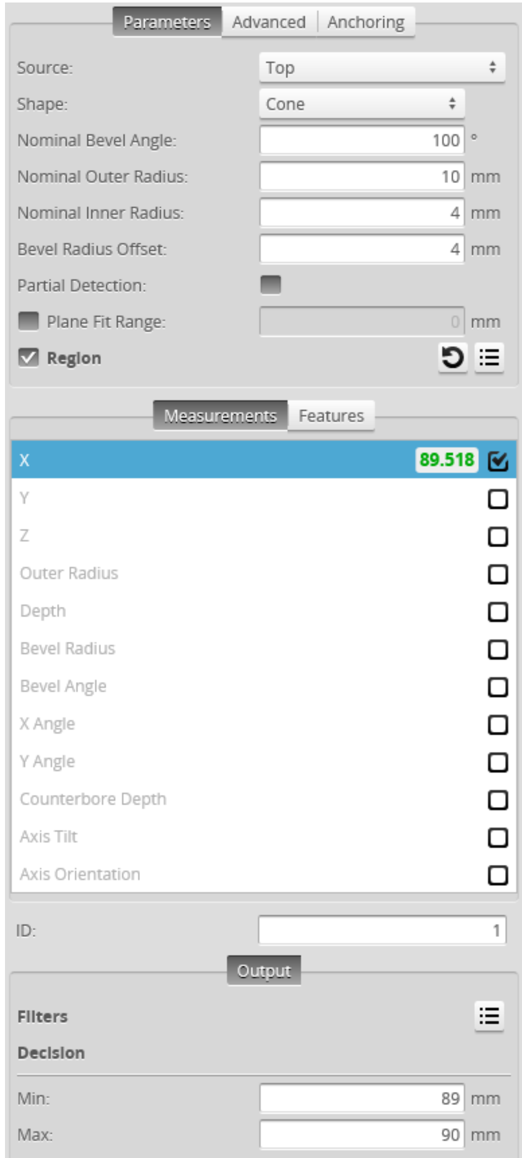
🗋	The tool does not search for or detect the feature. The tool expects that the feature, conforming reasonably well to the defined parameters, is present and that it is on a sufficiently uniform background.

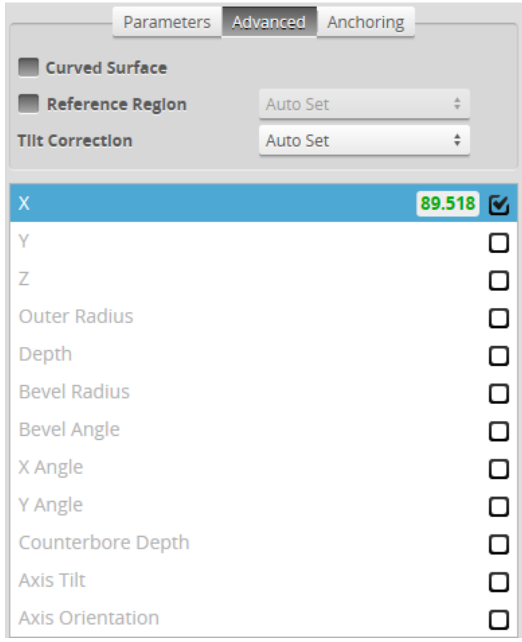


2D View



3D View


Parameters
Advanced
Anchoring
Source:
Top
Shape:
Cone
Nominal Bevel Angle:
100
°
Nominal Outer Radius:
10
mm
Nominal Inner Radius:
4
mm
Bevel Radius Offset:
4
mm
Partial Detection:
Plane Fit Range:
0
mm
Region
Measurements
Features
X
89.518
Y
Z
Outer Radius
Depth
Bevel Radius
Bevel Angle
X Angle
Y Angle
Counterbore Depth
Axis Tilt
Axis Orientation
ID:
1
Output
Filters
Decision
Min:
89
mm
Max:
90
mm

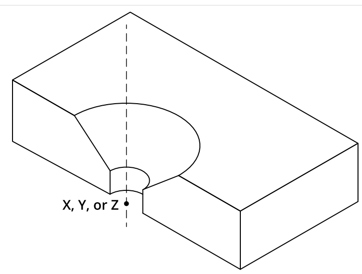
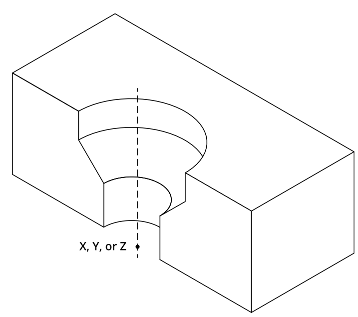


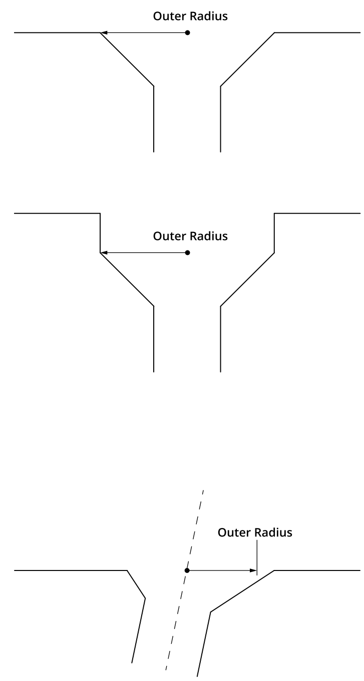
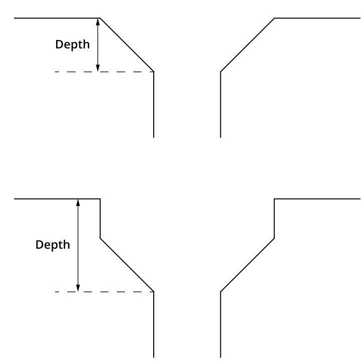
Measurement Panel

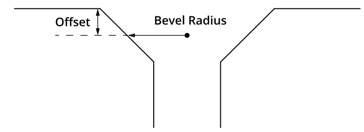
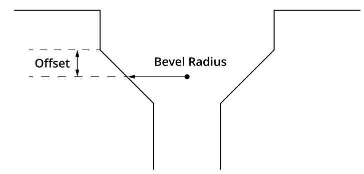
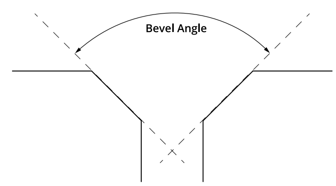
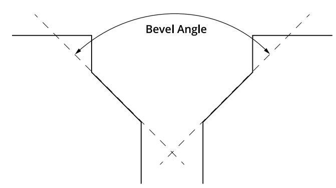
For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

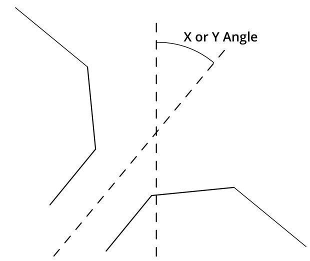
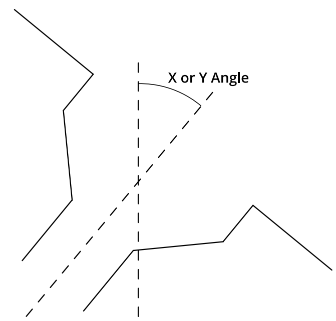
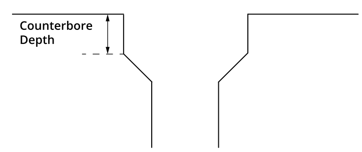
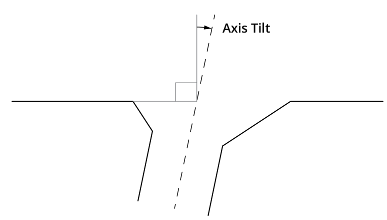
Measurements, Features, and Settings

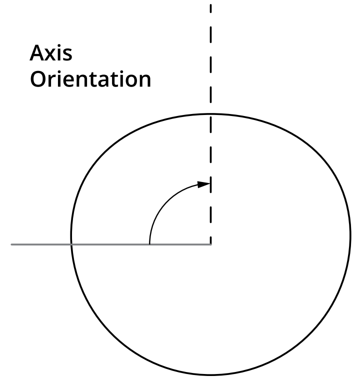
Measurements

Measurement	**Illustration**
X Determines the X position of the center of the countersunk hole.	
Y Determines the Y position of the center of the countersunk hole.	
Z Determines the Z position of the center of the countersunk hole.	

Measurement	Illustration
Outer Radius Determines the outer radius of the countersunk hole. When a hole is cut at an angle relative to the surrounding surface, the outer radius is calculated as if the hole were not cut at an angle. To convert the radius to a diameter, set the **Scale** setting in the **Output** panel (displayed after expanding the **Filters** section) to 2.	
Depth Determines the depth of the countersunk hole relative to the surface that the countersunk hole is on.	

Measurement	Illustration
Bevel Radius Determines the radius at a user-defined offset (**Offset** setting) relative to the surface that the countersunk hole is on. To convert the radius to a diameter, set the **Scale** setting in the **Output** panel (displayed after expanding the **Filters** section) to 2.	 
Bevel Angle Determines the angle of the hole's bevel.	 *Cone*  *Counterbore*

Measurement	Illustration
X Angle Determines the angle the hole relative to the X axis.	 *Cone*
Y Angle Determines the angle of the hole relative to the Y axis.	 *Counterbore*
Counterbore Depth Determines the depth of a counterbore.	
Axis Tilt Measures the tilt of the axis of the hole relative to the surface surrounding the hole. This measurement is not supported when **Shape** is set to **Counterbore**.	

Measurement	Illustration
Axis Orientation Measures the angle of the axis of the hole around the normal of the surface surrounding the hole, relative to the X axis. This measurement is not supported when **Shape** is set to **Counterbore**.	

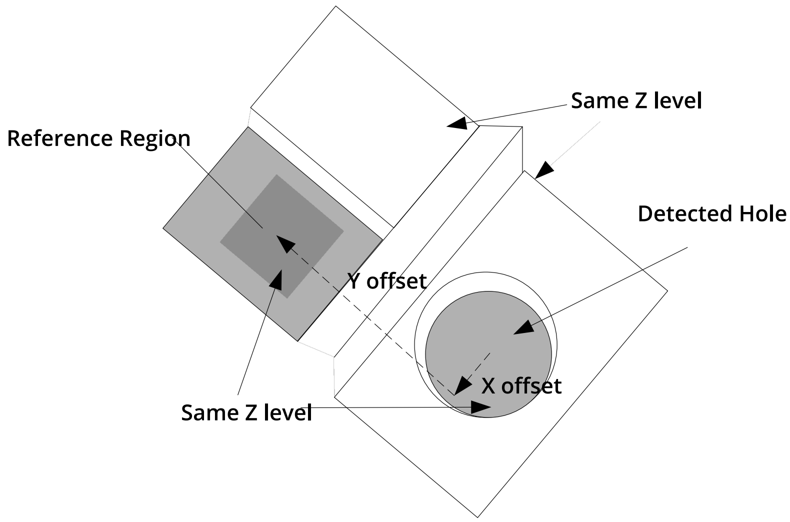
Features

Type	Description
Center Point	The center point of the countersunk hole. The Z position of the center point is at the Z position of the surrounding surface.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Shape	The shape of the countersunk hole. (See illustrations above.) 0 – Cone 1 – Counterbore
Nominal Bevel Angle	The expected bevel angle of the countersunk hole.
Nominal Outer Radius	The expected outer radius of the countersunk hole.
Nominal Inner Radius	The expected inner radius of the countersunk hole.
Bevel Radius Offset	The offset, relative to the surface that the countersunk hole is on, at which the bevel radius will be measured.
Partial Detection	Enable if only part of the hole is within the measurement region. If disabled, the hole must be completely in the region of interest for results to be valid.
Plane Fit Range	Excludes data beyond the specified distance from the plane surrounding the hole. You can use this setting to exclude surfaces close to the countersunk hole that step down from the plane surrounding the hole that could make measurement of the hole less reliable.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.

Parameter	Description
Curved Surface	Whether the surface that the countersunk hole is on is curved. When this setting is enabled, specify the orientation of the curvature in degrees in the **Curve Orientation** setting.
Curve Orientation	The orientation of the curvature in degrees. Only visible when **Curved Surface** is enabled.
Reference Regions	The tool uses the reference regions to calculate the Z position of the hole. It is typically used in cases where the surface around the hole is not flat.  When this option is set to **Autoset**, the algorithm automatically determines the reference region. When the option is not set to **Autoset**, you must manually specify one or two reference regions. The location of the reference region is relative to the detected center of the hole and positioned on the nominal surface plane. When **Reference Region** is disabled, the tool measures the hole's Z position using all the data in the measurement region, except for a bounding rectangular region around the hole.
Tilt Correction	Tilt of the target with respect to the alignment plane. **Autoset**: The tool automatically detects the tilt. The measurement region to cover more areas on the surface plane than other planes. **Custom**: You must enter the X and Y angles manually in the X Angle and Y Angle parameters (see below).
X Angle Y Angle	The X and Y angles you must specify when **Tilt Correction** is set to **Custom**. You can use the Surface Plane tool's X Angle and Y Angle measurements to get the angle of the surrounding surface, and then copy those measurement's values to the **X Angle** and **Y Angle** parameters of this tool.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.

Parameter	Description
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

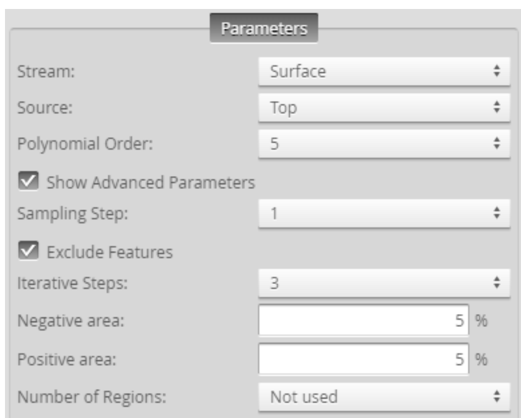
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Curvature

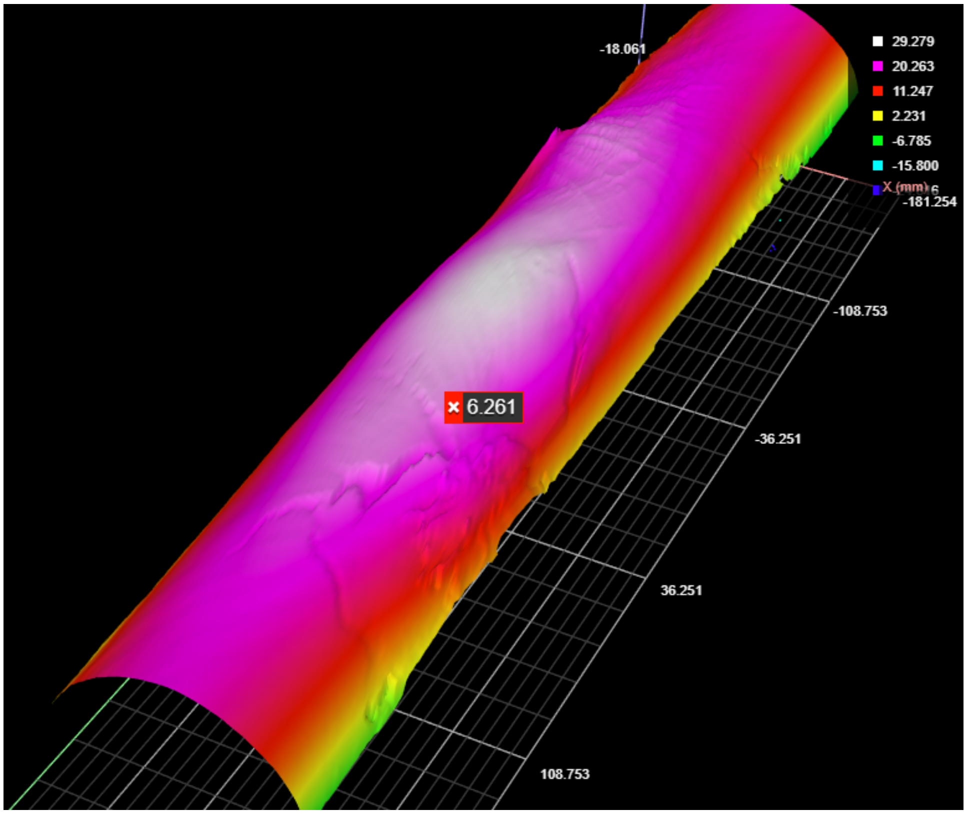
This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Surface Curvature tool removes curvature from curved surfaces while preserving surface features or defects, using a configurable polynomial order (the tool performs a 2D polynomial fit on X and Y to process surfaces). You can then use the tool's output apply measurements to the "flattened" surface.

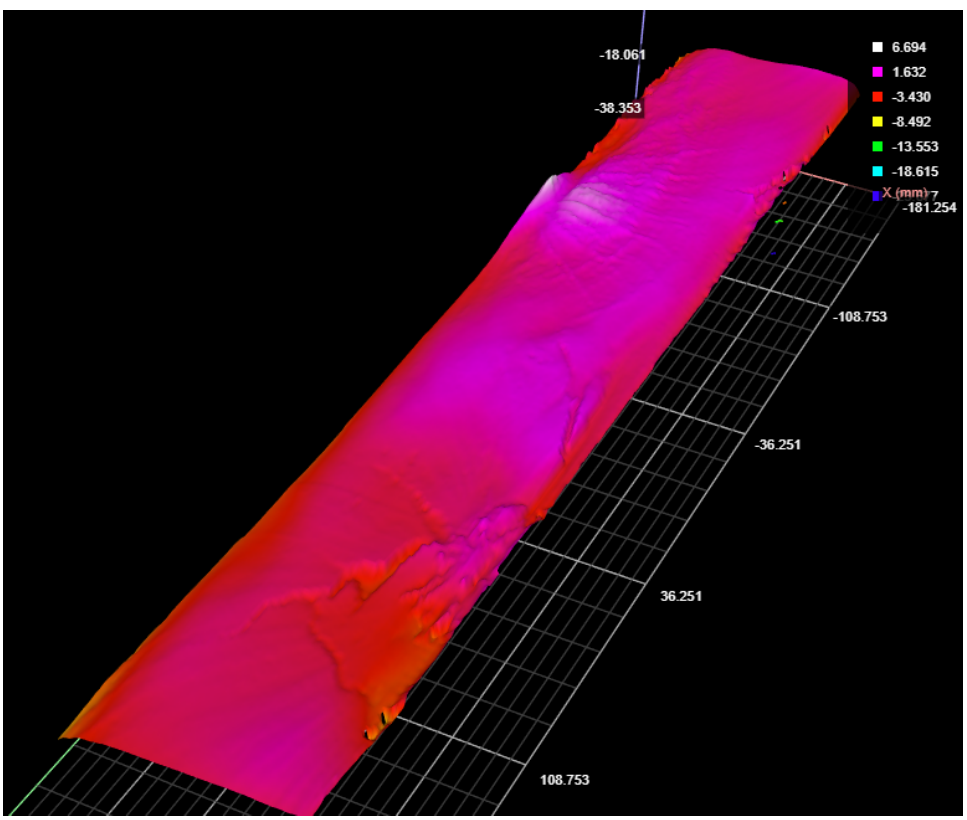


The tool does not support rotational scans (that is, polar "unwrapping").

In the following images, a curved surface (top) is flattened out (bottom), preserving the surface detail.

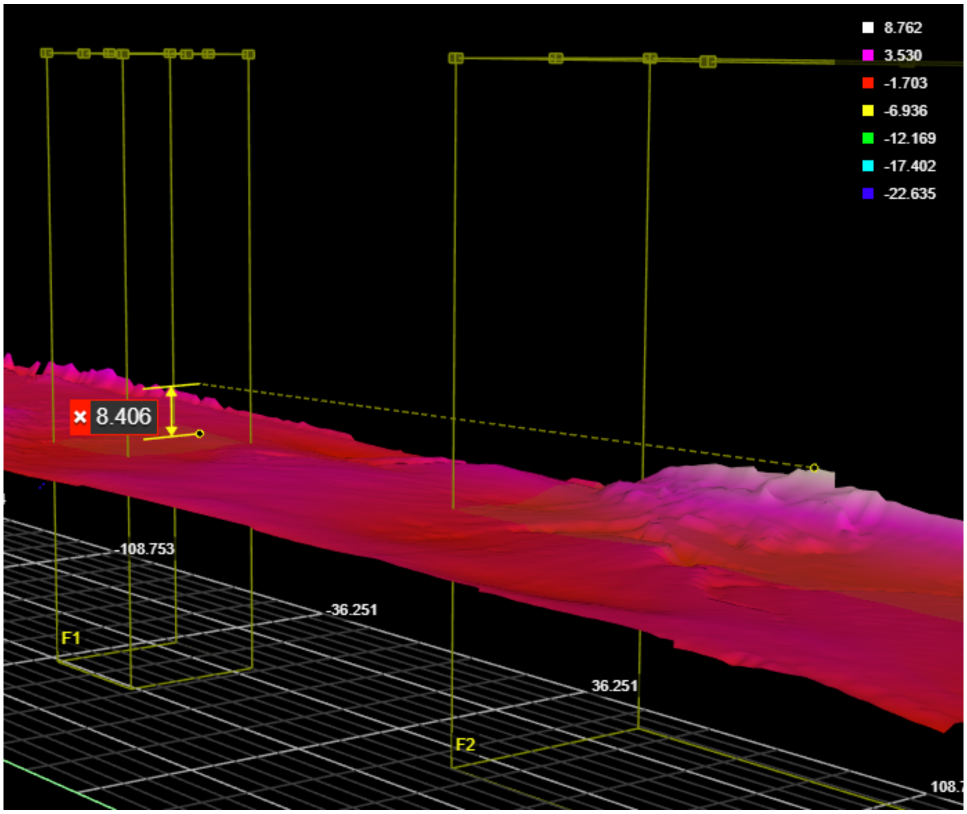


The original, curved scan of a target.



The "flattened" surface data (the tool's Difference Surface data output).

In the following image, a Surface Dimension tool's height measurement runs on the "flattened" output (the Surface Curvature tool's Difference Surface output) to determine the height of one of the raised areas:



Height of a raised feature relative to the previously curved surrounding surface.

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements and Settings

Measurements

Measurement
Processing Time The amount of time the tool takes to process.

Data

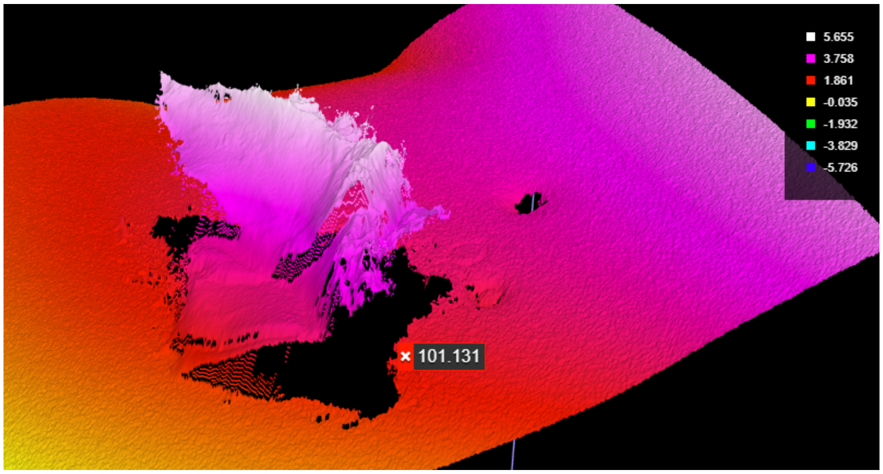
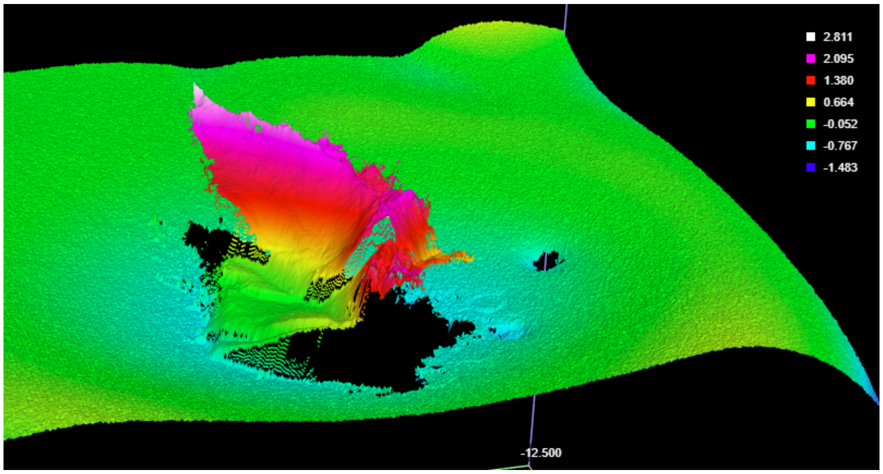
Type	**Description**
Fit Surface	The fitted polynomial the tool uses to flatten the original surface.
Difference Surface	The “flattened” surface: this is the original surface with the fitted polynomial removed.

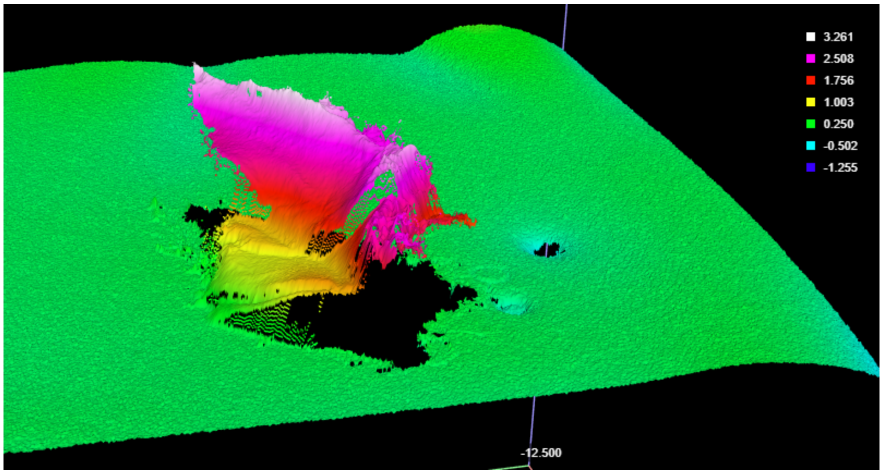
Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool. If you switch from one type of data to another (for example, from section profile data to surface data), currently set input features will become invalid, and you will need to choose features of the correct data type.
Polynomial Order	Selects the order (or degree) of the polynomial to be fit to the surface. A higher order results in a better fit but increases processing time.
Show Advanced Parameters	Enables a set of advanced parameters. (See *Advanced Parameters* below.)
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Advanced Parameters

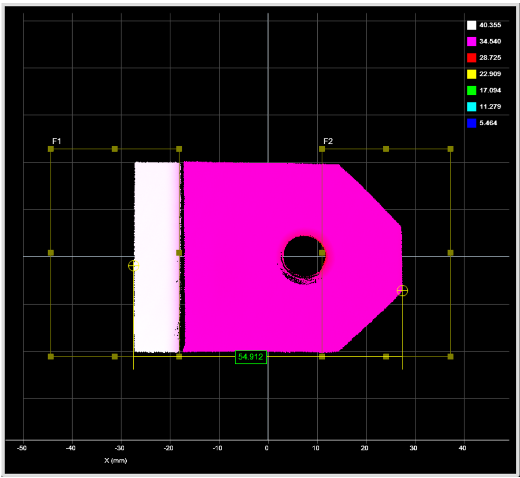
Parameter	Description
Sampling Step	The step in data points in both directions with which the surface is sampled. Choosing a higher sampling step reduces the processing time the tool requires, but reduces fit accuracy. Useful if the surface being processed has a large number of data points.

Parameter	Description
Exclude Features	Lets you exclude features or surface details from the polynomial fit. This can allow you to get a better fit on the surrounding surface. Checking this option enables the **Negative area**, **Positive area**, and **Iterative Steps** parameters. (See below.) For example, in the following scan data, we would like to accurately measure the circular divots and the small hole near the center of the data on the curved surface.  If the large feature to the left is *not* excluded for the polynomial fit, the fitted surface will and therefore the measurements on the smaller features will be inaccurate. In the following "flattened" scan data, without excluding the larger feature, the smaller features would be difficult to accurately measure:  When the larger feature is excluded from the polynomial fitting, the surrounding surface and the smaller features are more properly "flattened."

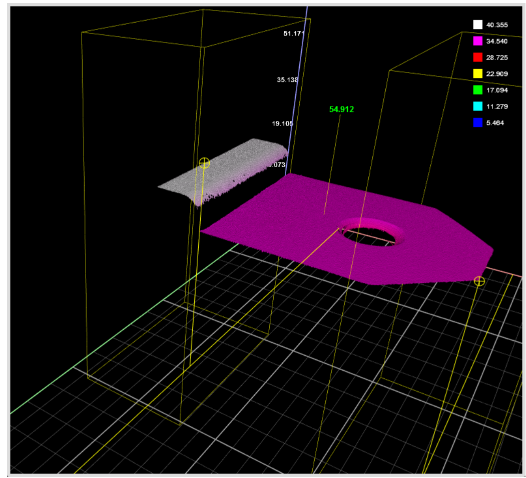
Parameter	Description
	
Iterative Steps	The number of times the tool repeats the feature exclusion calculation (see **Exclude Features**, above).
Negative area Positive area	These settings exclude the specified percentage of a histogram of the height values of the scan data from the botom up (**Negative area**) and from the top down (**Positive area**), respectively.
Number of Regions	Lets you specify and configure one or more regions that the tool will process. Use this parameter to limit the tool to specific areas on the target.

Dimension

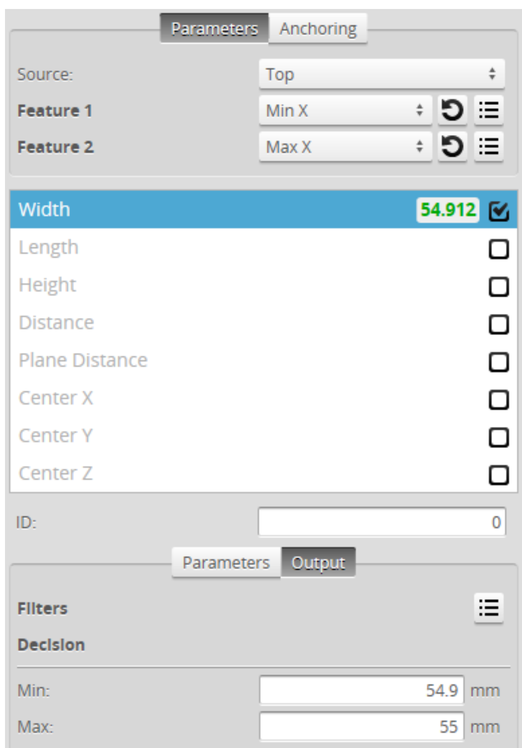
The Dimension tool returns various dimensional measurements of a part. You must specify two feature types (see below).



2D View

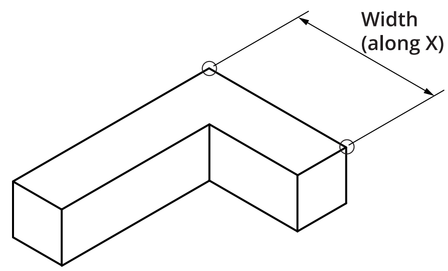
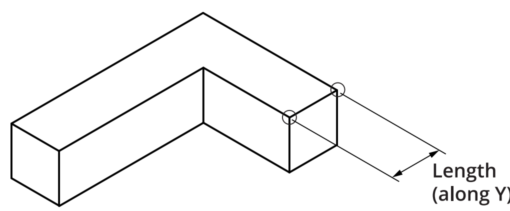
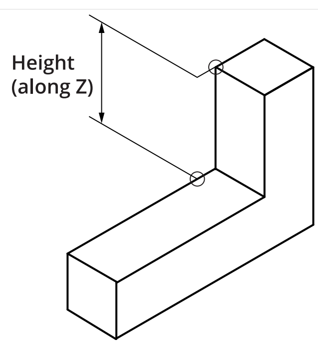


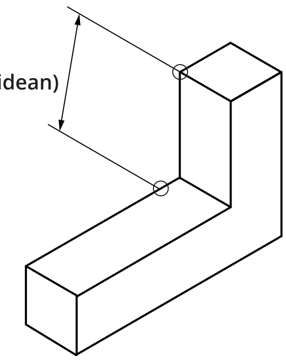
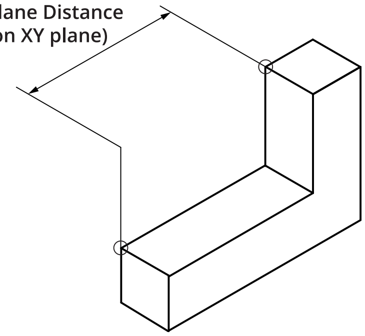
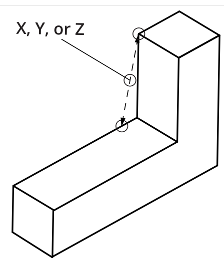
3D View



Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurement	**Illustration**
Width Determines the distance between the selected features along the X axis.	
Length Determines the distance between the selected features along the Y axis.	
Height Determines the distance between the selected features along the Z axis.	

Measurement	Illustration
Distance Determines the direct, Euclidean distance between the selected features.	
Plane Distance Determines the distance between the selected features. The position of the lowest feature point is projected onto the XY plane of the highest feature point.	
Center X Determines the X position of the center point between the selected features.	
Center Y Determines the Y position of the center point between the selected features.	
Center Z Determines the Z position of the center point between the selected features.	

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.

Parameter	Description
Feature 1 Feature 2	The **Feature 1** and **Feature 2** settings represent the two features the tool uses to perform measurements. For each, one of the following: • Average • Median • Centroid • Max X • Min X • Max Y • Min Y • Max Z • Min Z To set the region of a feature, adjust it graphically in the data viewer, or expand the feature using the expand button (☰) and enter the values in the fields. For more information on regions, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

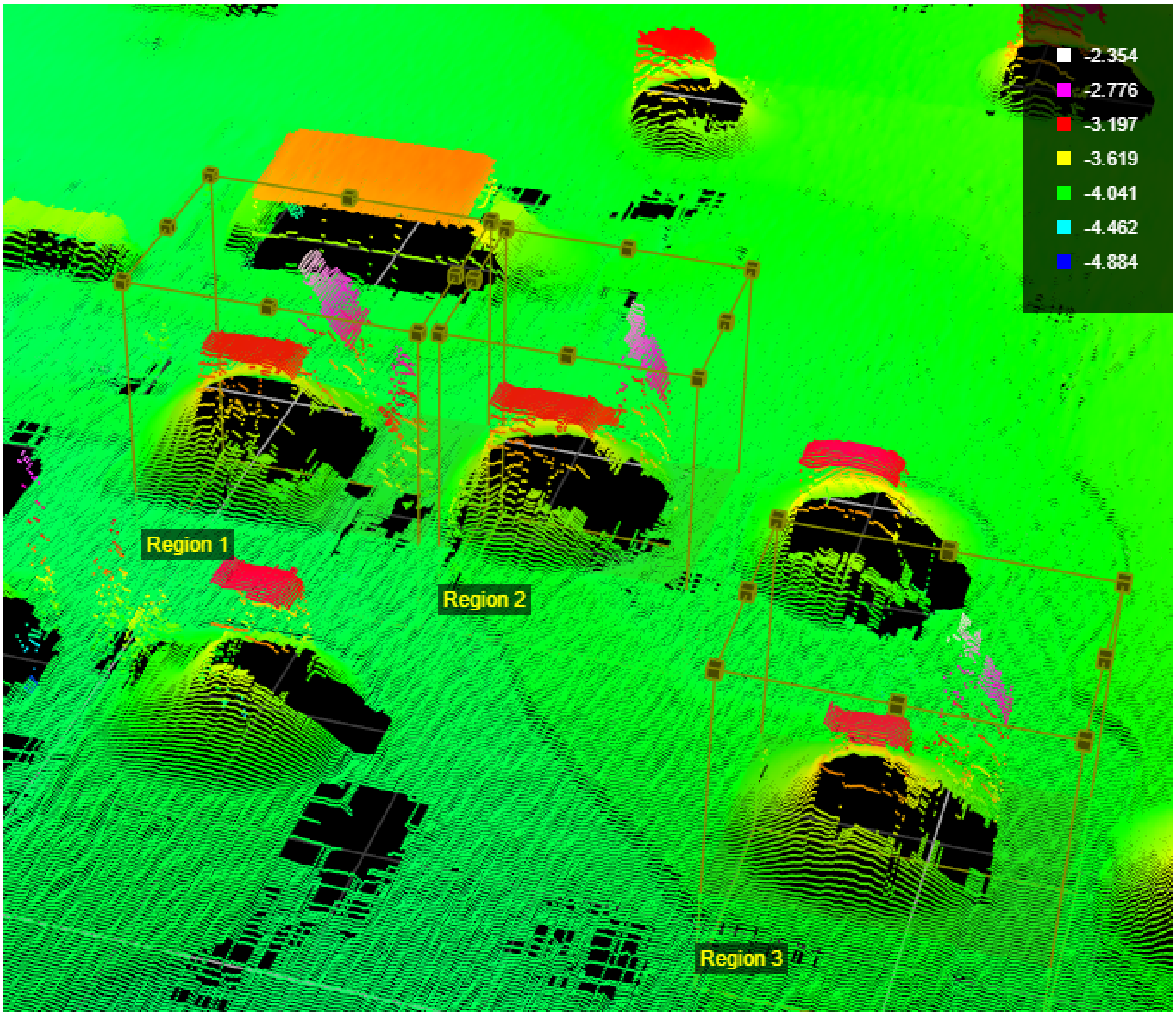
For more information on anchoring, see *Measurement Anchoring* on page 169.

Direction Filter

This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

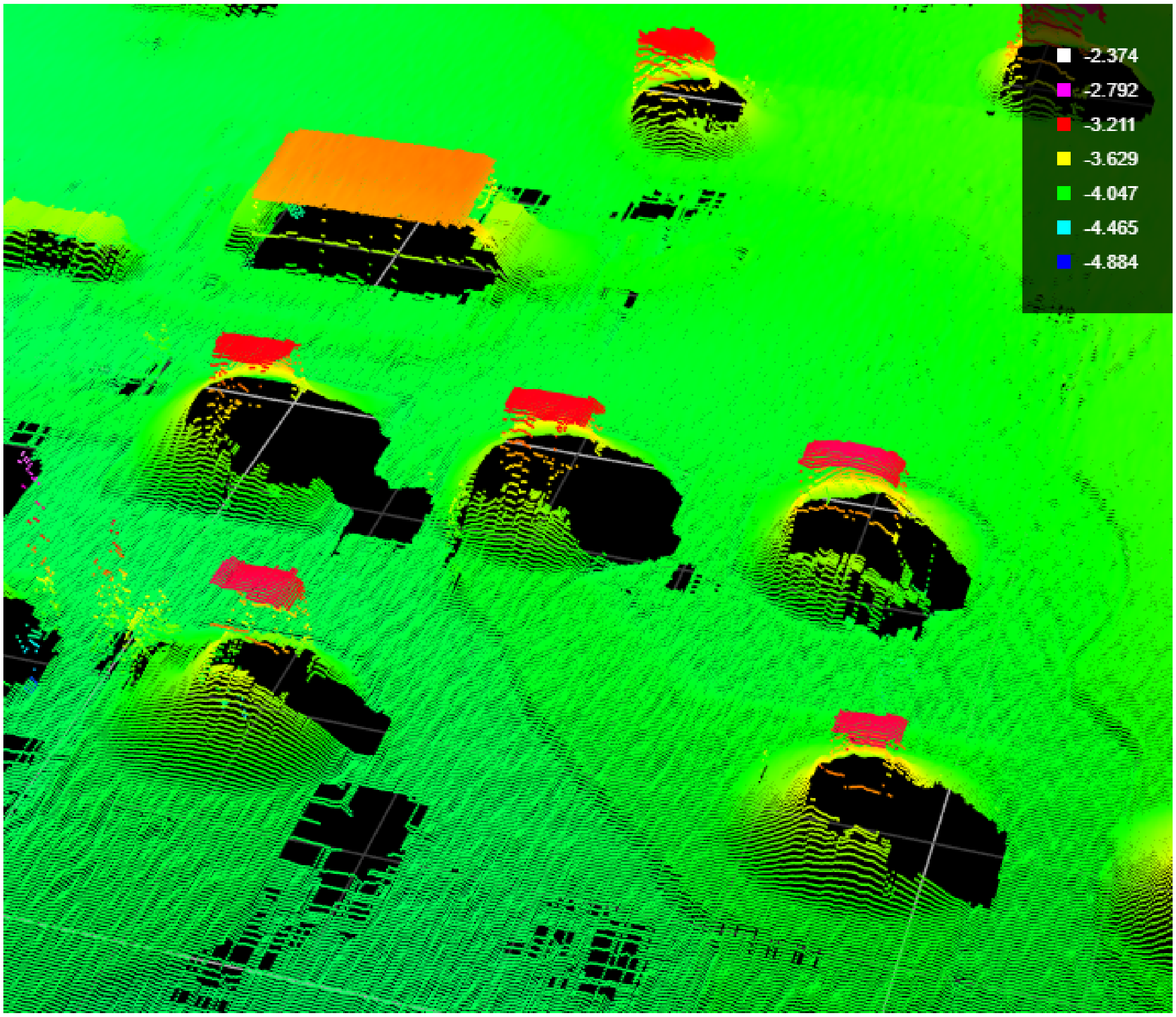
The Surface Direction Filter helps exclude unwanted data points based on their “orientation” (relative to surrounding data points) in 3D space, for example, data points resulting from reflections. The tool can provide better results than median or height based filters. The tool lets you define up to 16 regions, and for each region, configure the characteristics of the data points to exclude.

For example, in the following scan data, noise (in pink) appears to the right of three surface mount components on a PCB. In this case, the "direction" (specifically, the polar angle) of the noise is roughly 75 to 85 degrees, relative to Z.

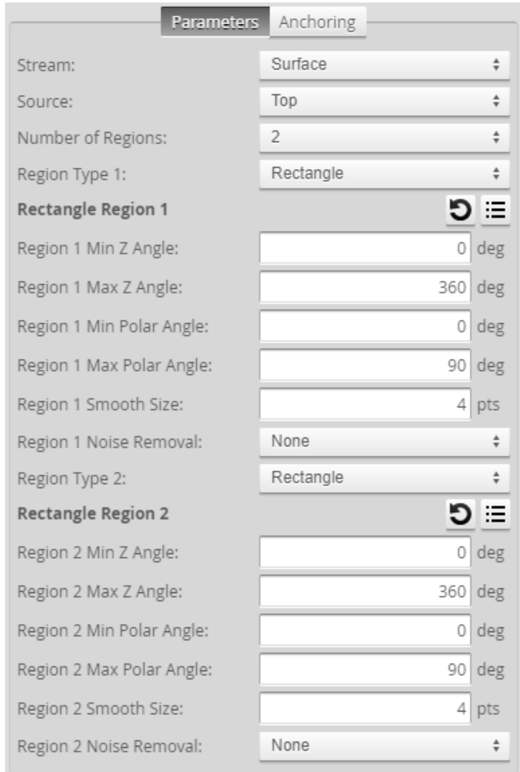


Surface before direction filtering.

In the following scan data, the tool has removed the noise.



Surface after direction filtering.



Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Data, and Settings

Measurements

Measurement
Processing Time The amount of time the tool takes to process.

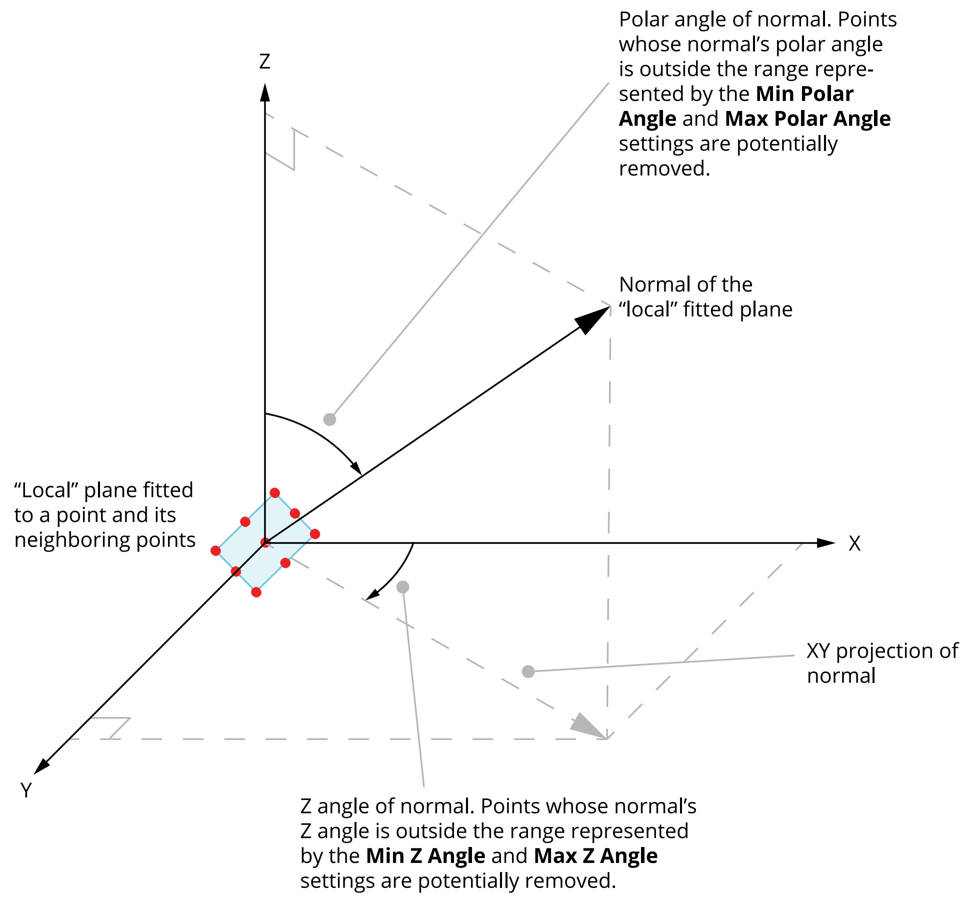
Data

Type	Description
Filtered Surface	The surface after filtering.

Parameters

Parameter	Description
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool. If you switch from one type of data to another (for example, from section profile data to surface data), currently set input features will become invalid, and you will need to choose features of the correct data type.
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Region Count	Only displayed on older instances of this tool. Newer instances use "flexible regions" (see the parameters below in this table). The number of regions the tool applies filtering to.
Region {n}	Only displayed on older instances of this tool. Newer instances use "flexible regions" (see the parameters below in this table). Lets you configure the size and position of region {n}. For the region-specific parameters, see *Region Filtering Parameters* on the next page.
Number of Regions Region Type {n} Inner Circle Diameter Inner Ellipse Major Axis Inner Ellipse Minor Axis Sector Start Angle Sector Angle Range Mask Source Low Threshold High Threshold	Only displayed on newer instances of this tool. When you enable **Use Region**, the tool displays additional settings related to the measure region type. For details on flexible regions and their settings, see *Flexible Regions* on page 153. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

The following illustrates the angle parameters that control which data points are excluded in scan data (see *Region Filtering Parameters* below):



The number of neighboring points shown above is for illustrative purposes only.

Region Filtering Parameters

Region {n} Min Z Angle Region {n} Max Z Angle	The minimum and maximum acceptable angles around the Z axis of the XY projection of the normal of the surface surrounding a data point, where 0 degrees is defined as positive X and positive rotation is clockwise around the Z axis.
Region {n} Min Polar Angle Region {n} Max Polar Angle	The minimum and maximum acceptable angles of the normal of the surface surrounding a data point with respect to the Z axis.
Region {n} Smooth Size	A mean filter applied to the surface data before calculating the normals in order to avoid abrupt normal changes due to noise.
Region {n} Noise Removal	Eliminates noise that can be introduced by the tool's normal calculation.

Anchoring

Anchor	**Description**
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

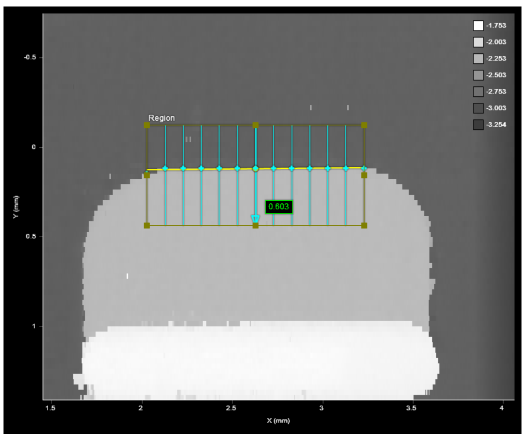
Edge

The Edge tool fits a line to a straight edge in the scan data, using either height map or intensity data. The tool's settings help fit the line when multiple potential edges are in the region of interest. After the tool locates an edge, it returns the position (X, Y, and Z) of the center of the edge line in the region of interest. The tool also returns its angle around the Z axis, the step height between the upper and lower surfaces adjacent to the edge, minimum and maximum error points to either side of the line, and a point count.

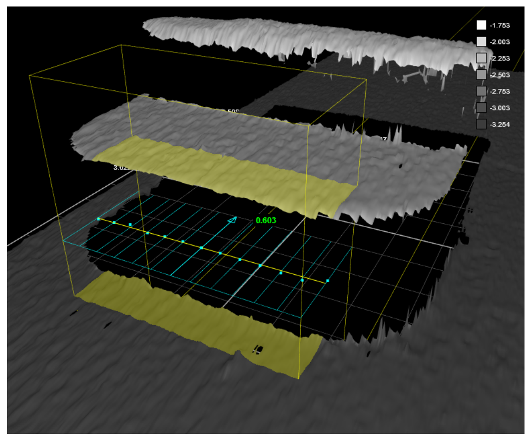
You can use the Z Angle measurement of the edge line with some tools to perform angle anchoring, compensating for minor part rotations around the Z axis, greatly increasing repeatability between part scans; for more information see *Measurement Anchoring* on page 169.

The minimum and maximum errors are useful for calculating a straightness value (using a script tool, for example; for more information, see *Script* on page 483).

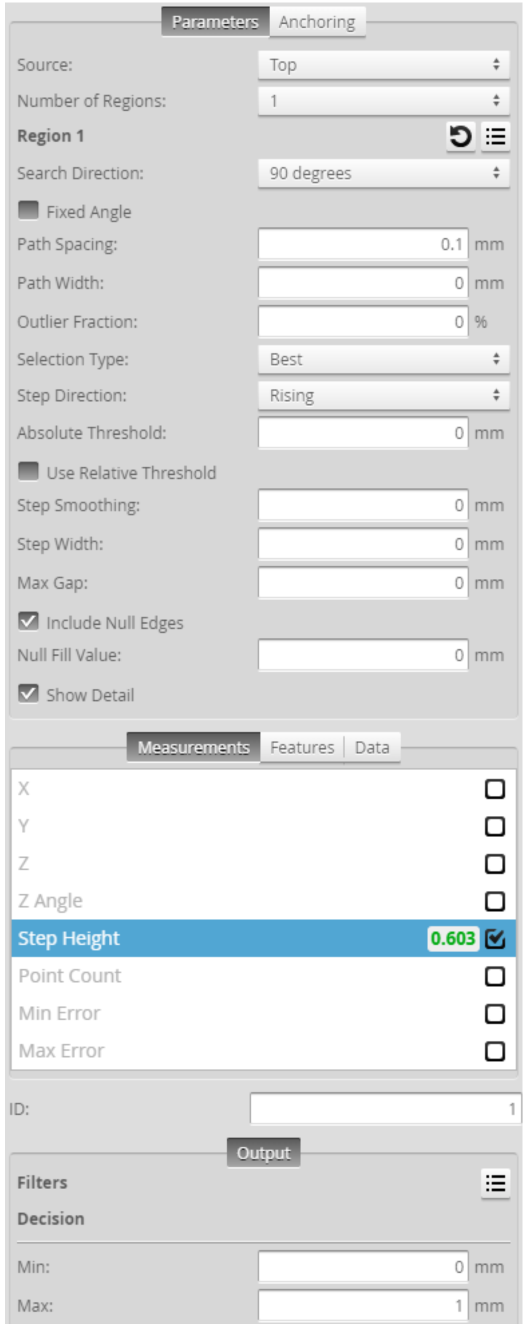
The tool can also generate edge line and center point geometric features that Feature tools can take as input for measurement. For more information on Feature tools, see *Feature Measurement* on page 484.



2D View



3D View

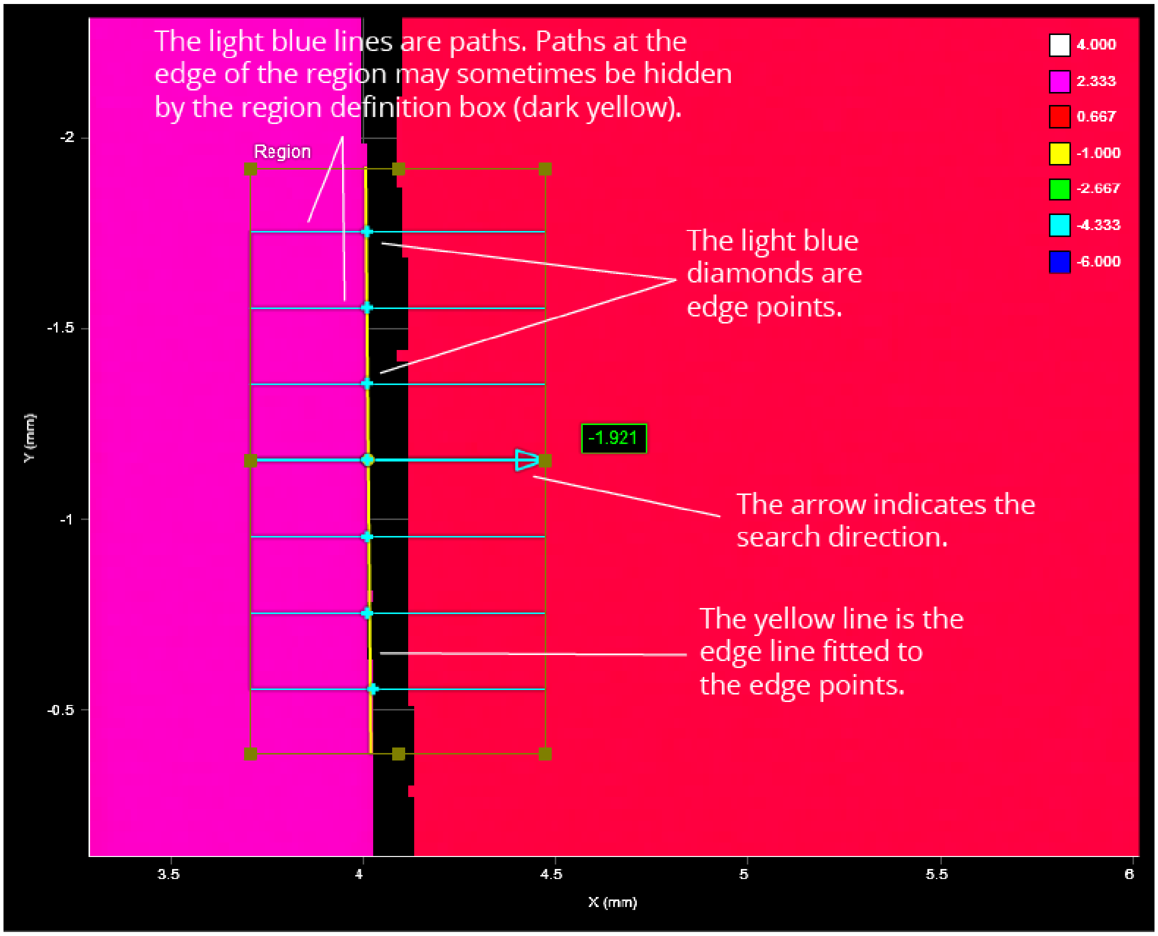


Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

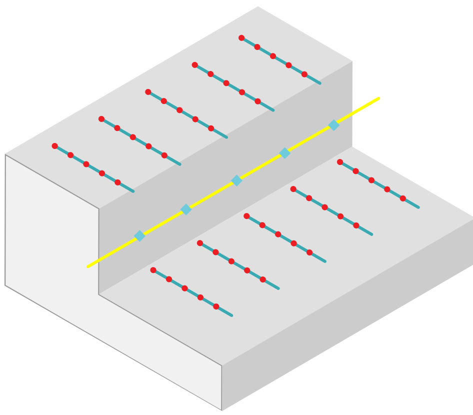
Paths and Path Profiles

To fit an edge line to the scan data, the Surface Edge tool overlays evenly spaced, parallel *paths* (light blue lines in the interface; see below) in the defined region of interest.



For each path, a profile is generated internally from the height map's data points that fall under or, optionally, near the path. The tool then examines each path profile for steps (changes in height) that meet the criteria set by the tool's settings, such as minimum height, direction (whether it is rising or falling), and so on.

Red dots are data points from the scan data that fall under paths (light blue lines).

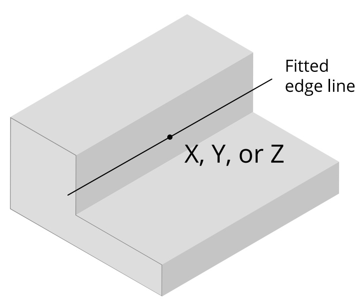


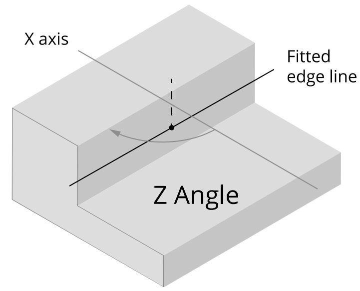
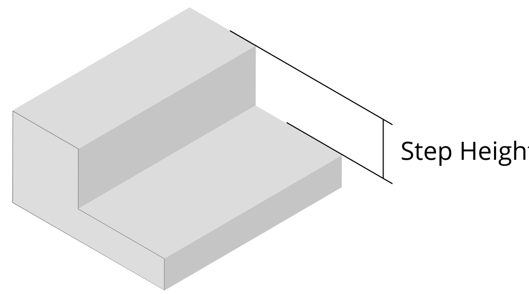
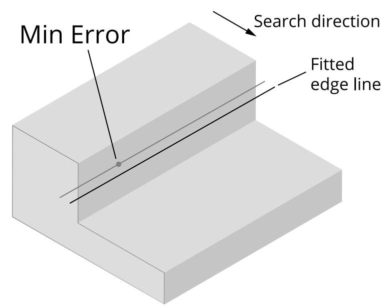
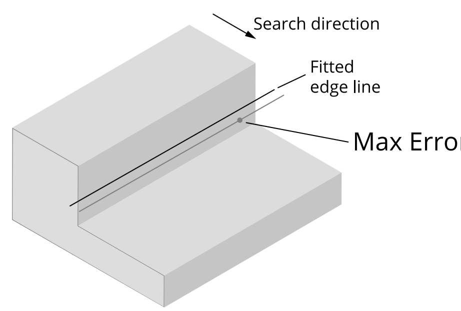
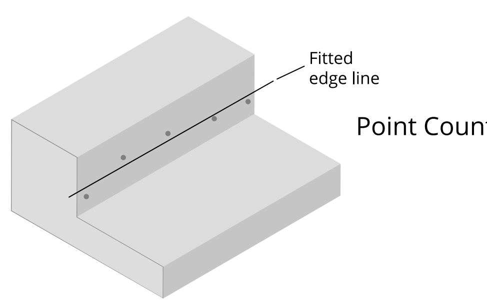
A single path profile extracted from a path.

For the step on each path profile that matches the settings, the tool places an edge point between the upper and lower area (light blue diamonds in the interface). The tool then fits a line to those edge points (yellow line in the interface). You can choose the orientation of the paths around the Z axis to accommodate different edge orientations.

Measurements, Features, Data, and Settings

Measurements

Measurement	Illustration
X Returns the X position of the center point of the fitted edge line.	
Y Returns the Y position of the center point of the fitted edge line.	
Z Returns the Z position of the center point of the fitted edge line.	

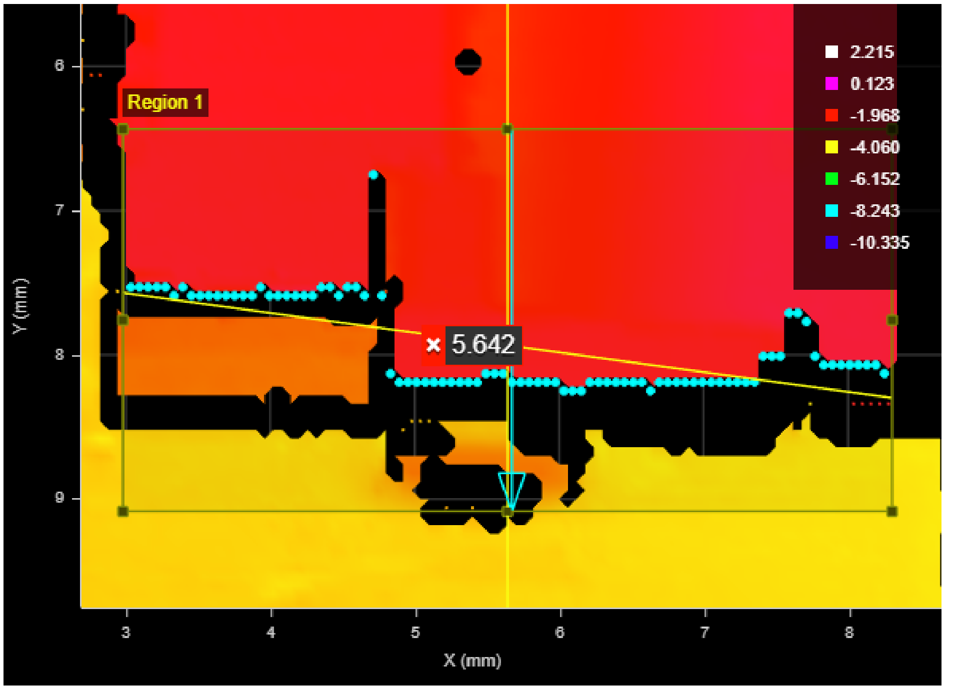
Measurement	Illustration
Z Angle Returns the rotation, around the Z axis, of the fitted edge line. Rotating the measurement region has no impact on the angle that is returned unless a different edge is detected. Useful for using minor variations in the rotation of an edge on target as an anchor for other measurements. For more information, see *Measurement Anchoring* on page 169.	
Step Height Returns the height of the step, calculated by averaging the step heights of all of the path profiles. (When **Use Intensity** is enabled, the value returned is the difference in intensity.)	
Min Error **Max Error** These measurements return the distances of the point furthest before the line (Min Error) and the point furthest after the line (Max Error), based on the search direction specified in the tool. The measurements ignore points excluded using the	 
Point Count The number of points used to fit the line. Useful for determining if the number of points is above an acceptable minimum.	

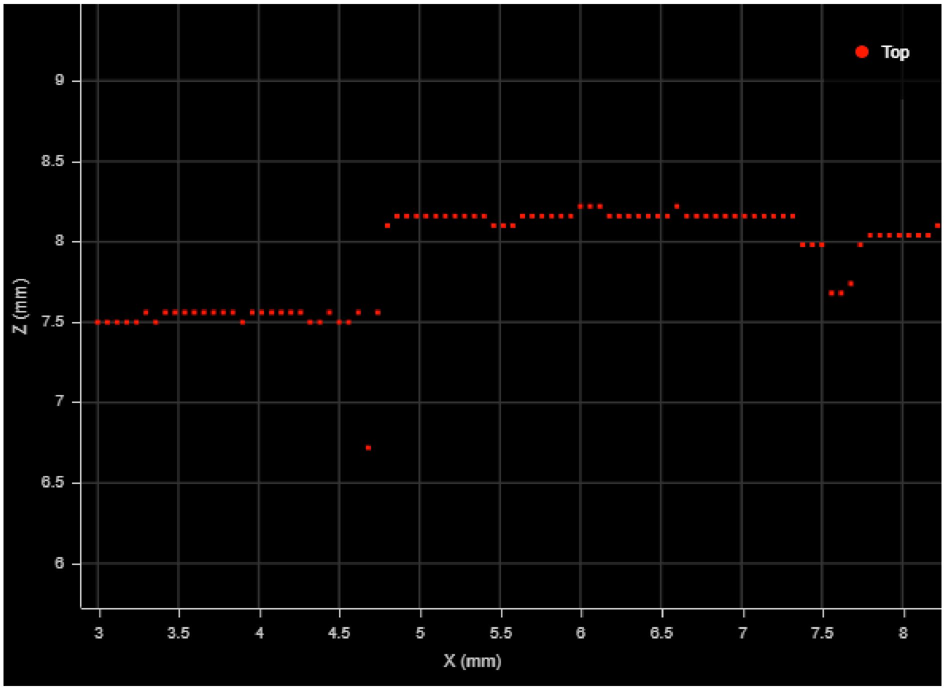
Features

Type	Description
Edge Line	The fitted edge line.
Center Point	The intersection point of the fitted edge line and a line through the center of the region at the path angle.

For more information on geometric features, see *Geometric Features* on page 164.

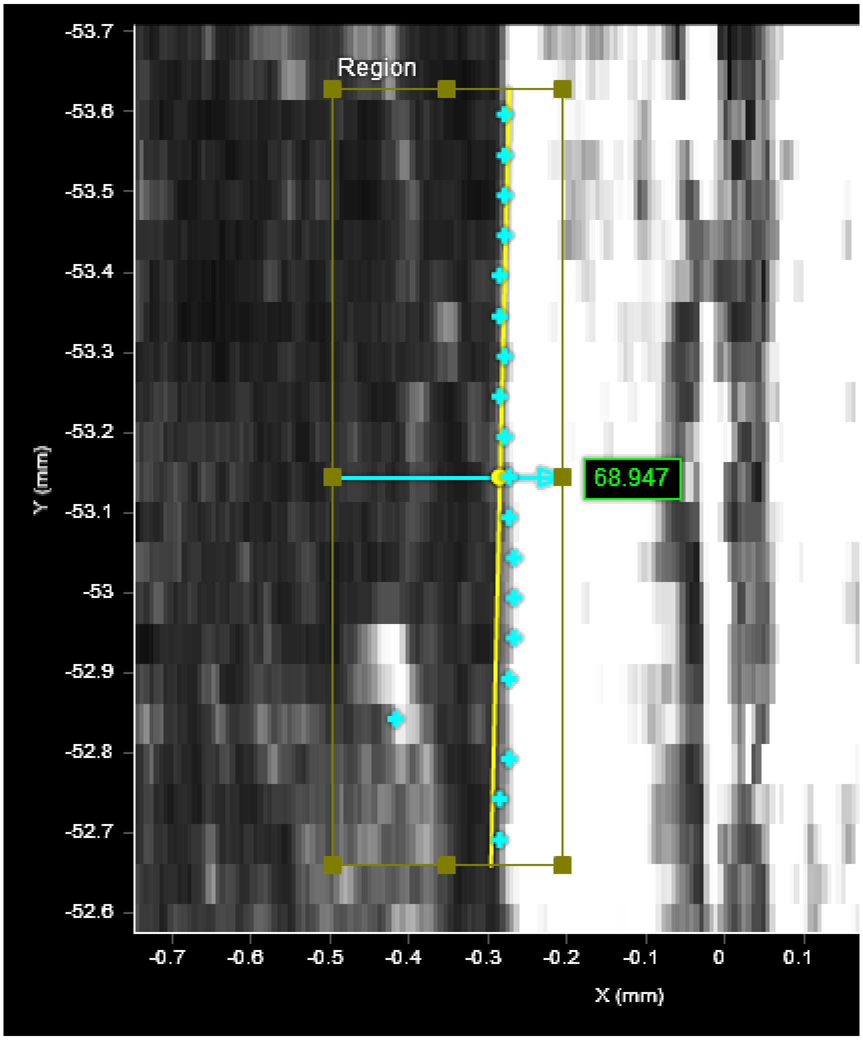
Data

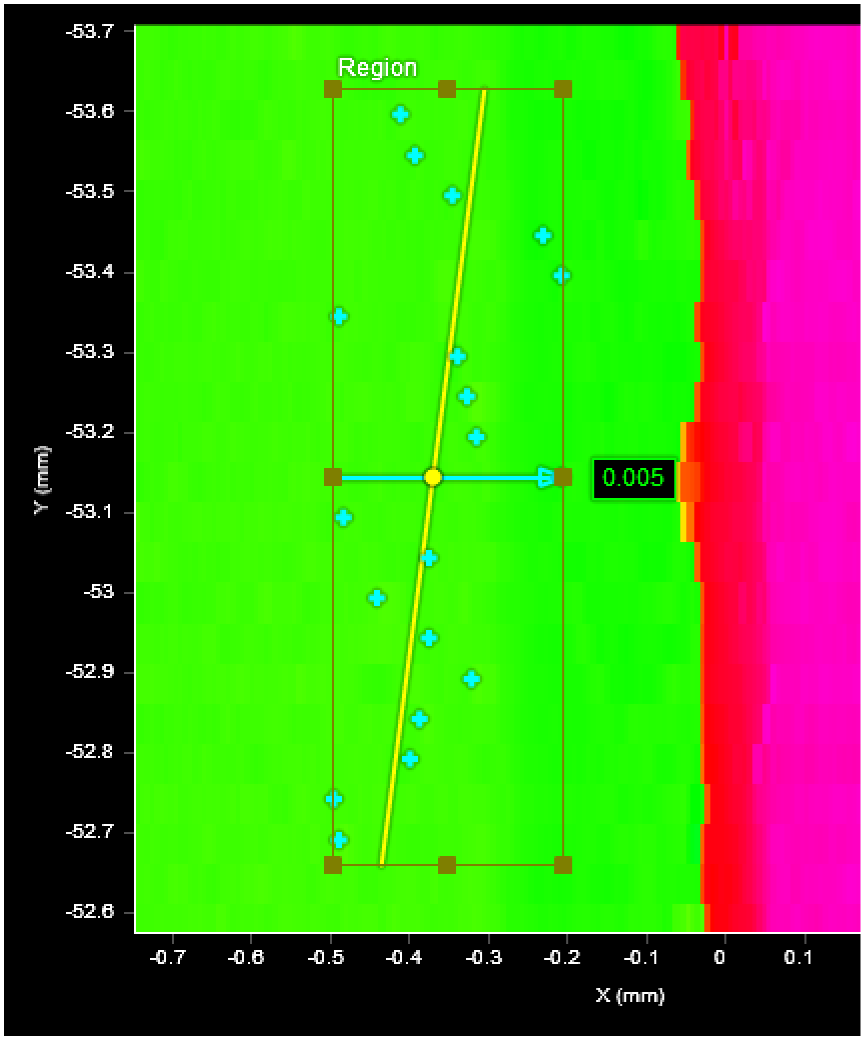
Type	Description
Profile Point Cloud Profile Region {n}	A point cloud profile (Profile Point Cloud) and one or more uniform spacing profiles (Profile Region {n}) representing the edge, respectively, made up of the tool's edge points. The XY positions of the edge points on the surface (cyan dots below) are represented as the XZ positions of the profile points, where X => X and Y => Z. Given the following edge, the resulting profile is shown further below:  The profile is mirrored vertically when compared to the edge: Note how the single edge point toward the top of Region 1 in the surface data above is at the bottom of the extracted profile (below).

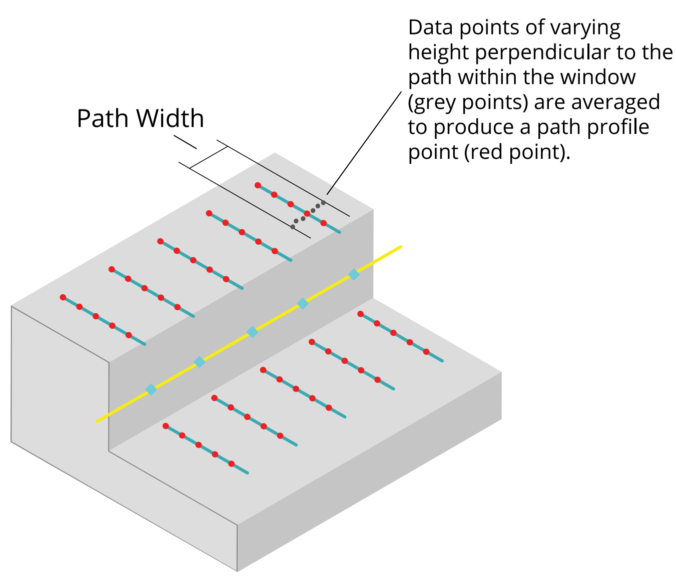
Type	Description
	

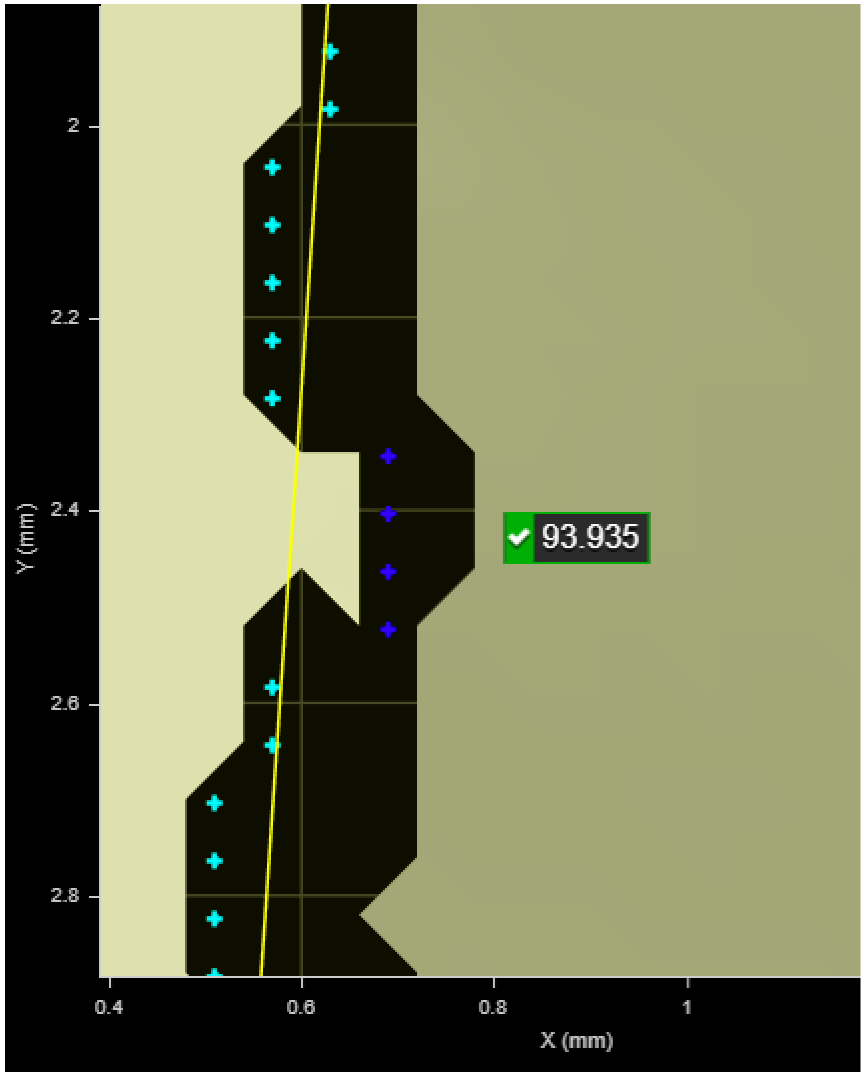
Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.

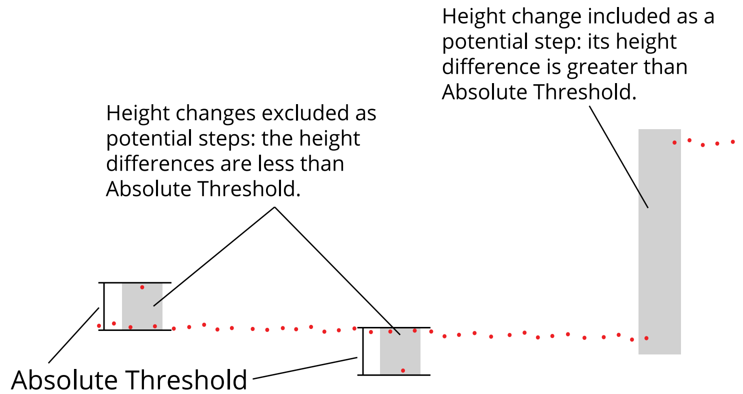
Parameter	Description
Use Intensity	(This setting is only available when **Acquire Intensity** is enabled in the **Scan Mode** panel; for more information, see *Scan Modes* on page 88.) Uses intensity data rather than height data to find an edge. Useful when color differences on a flat area of a target, which would not be detected using height map data, are distinct, letting you use the detected "line" as an anchor source or perform geometric feature measurements.  ***Use Intensity** enabled (intensity view): Surface Edge tool finds the edge using intensity data.*

Parameter	Description
	 ***Use Intensity** disabled (heightmap view of the same area): Surface Edge tool unable to find edge using height data.*
Number of Regions	The number of regions the tool will use to fit the line. You must configure each region (see *Region {n}* below). Using multiple regions allows you to fit a line to an edge that is not straight along its entire length or that is not continuous.
Region {n}	The region or regions the tool uses to fit a line. For more information, see *Regions* on page 152. The **Search Direction** setting applies to all of the regions. You can configure the **Z Angle** of each region independently to accommodate the particularities of the feature or target (for example, to exclude unwanted scan data next to one of the regions in the fitting of the line to the edge).

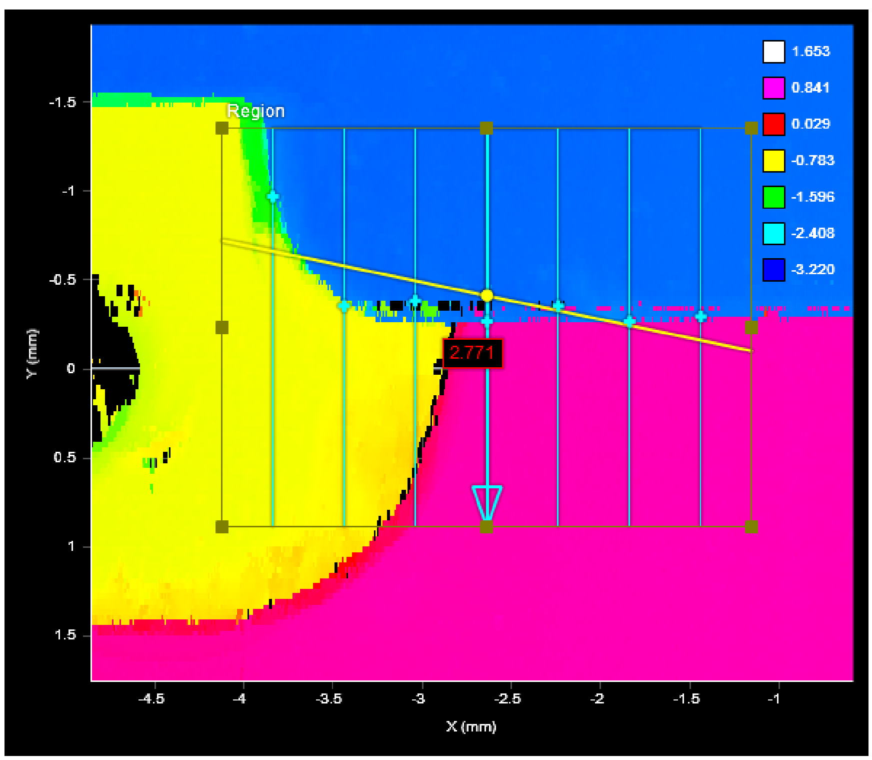
Parameter	Description
Search Direction	The search direction for steps, specified as an orientation around the Z axis, relative to the X axis. Can be 0, 90, 180, or 270 degrees. Choose a value that is roughly perpendicular to the edge on the target. The direction is indicated by a light blue arrow in the data viewer.
Fixed Angle	When this option is enabled, the value in **Fixed Angle Value** replaces the value the Z Angle measurement returns. Useful when the angle of the feature is known and noise in the scan data could otherwise cause the measurement to return an incorrect angle.
Fixed Angle Value	The value the tool uses to locate the edge and returns for the Z Angle measurement. You must enable **Fixed Angle** to set this value.
Path Spacing	Sets the spacing between paths in the measurement region used to extract the profiles that determine the edge. A higher number of paths results in a higher number of edge points, which makes the fitting of the edge line more accurate. However, a higher number of edge points results in a greater tool execution time. When **Path Spacing** is set to 0, the resolution of the scan data is used as the basis for spacing. No paths are displayed in the data viewer in this case.
Path Width	The size of the windows perpendicular to the path used to calculate an average for each data point on a path profile. Useful to average out noise along the path caused by reflections, and so on.  If **Path Width** is set to 0, no averaging is performed (only the data point under the path is used). For averaging along the path, use **Step Smoothing** (see below).

Parameter	Description
Outlier Fraction	The percentage of outlier points to exclude. Setting this to a small value can help the tool fit the line better to the edge.  ***Outlier Fraction** set to a low value: rejected outlier edge points are dark blue.*
Selection Type	Determines which step the tool uses on each path profile when there are multiple steps in the profile. An edge point is placed on each chosen step. Steps must pass the criteria of the tool's **Absolute Threshold**, **Step Direction**, and **Relative Threshold** settings. **Best**: Selects the greatest step on each path profile. **First**: Selects the first step on each path profile. **Last**: Selects the last step on each path profile.
Step Direction	Determines whether the expected step rises or falls along the path. Either **Rising**, **Falling**, or **Rising or Falling**.

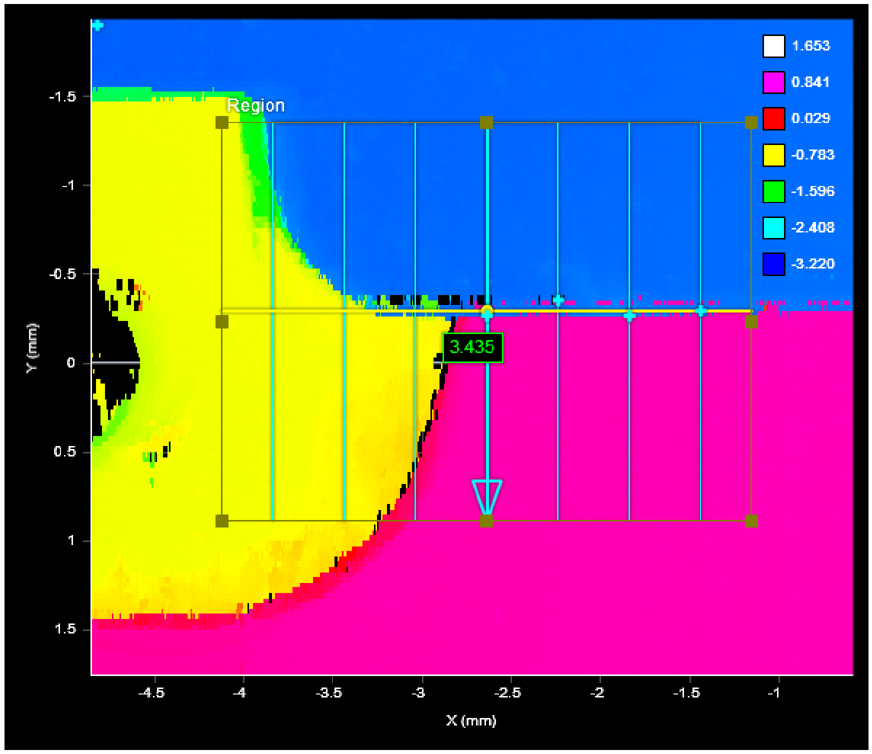
Parameter	Description
Absolute Threshold	When **Use Intensity** is disabled, the setting specifies the minimum *height* difference between points on a path profile for that step to be considered for an edge point. The setting can be used to exclude smaller steps on a part that should not be considered for an edge, or to exclude height differences caused by noise. When used in conjunction with **Relative Threshold**, **Absolute Threshold** is typically set to a small value, greater than the general surface roughness.

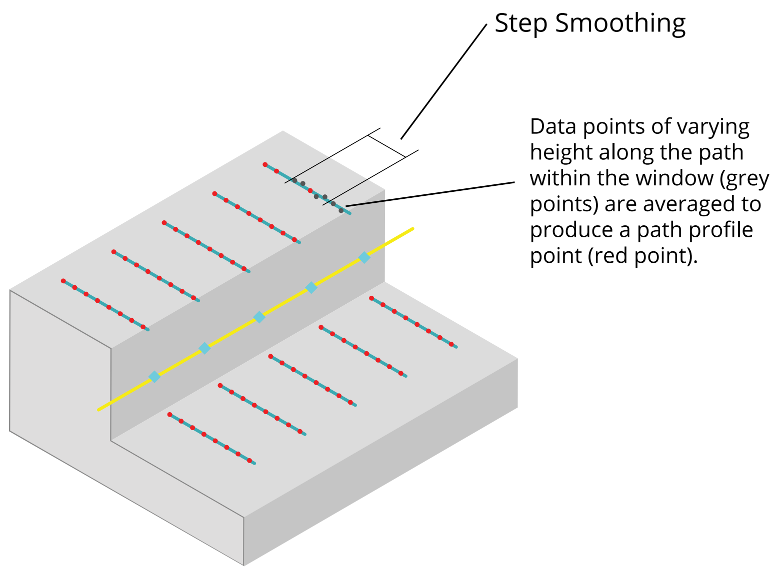
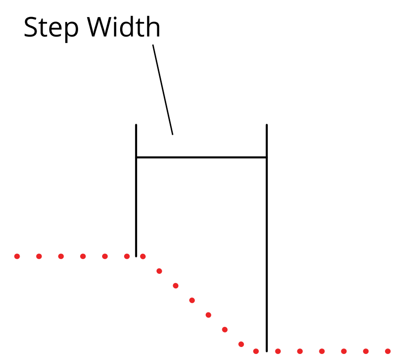


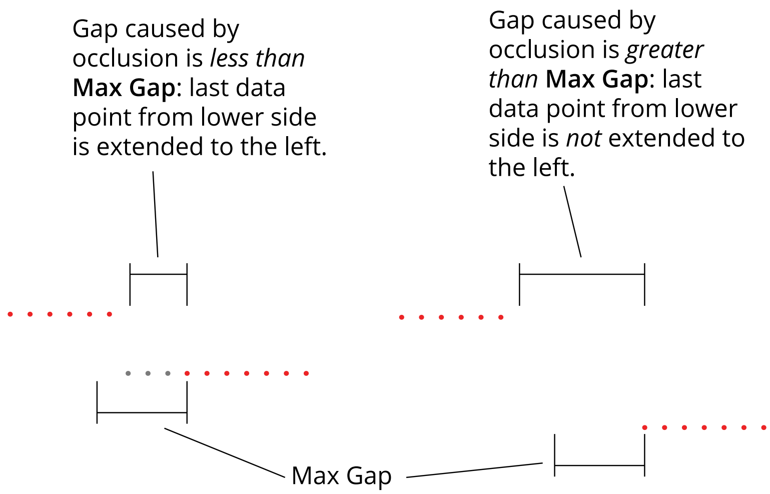
In the image below, when **Absolute Threshold** is left at the default of 0, all steps are included as possible candidates for an edge, and will be used to fit an edge line. The resulting edge line is angled upward to the left.



When **Absolute Threshold** is set to 3 with the same data (see image below), steps going from the yellow to pink regions (roughly 1.37 mm) and from the blue to yellow regions (roughly 2 mm) are

Parameter	Description
	excluded. Only steps from the blue to pink regions (roughly 3 mm) are included.  When **Use Intensity** is enabled, the setting specifies the minimum difference in intensity. (**Acquire Intensity** must enabled in the **Scan Mode** panel.)
Use Relative Threshold	When this option is enabled, the **Relative Threshold** field is displayed.
Relative Threshold	The value for the relative threshold. The tool calculates a relative threshold by scaling the greatest height or intensity difference found on the path profiles by the percentage in **Relative Threshold**. This lets you configure the tool without knowing the actual step height in advance, and is useful for edges with varying step height. For a height or intensity difference to be considered a valid step, both **Absolute Threshold** and **Relative Threshold** must pass.

Parameter	Description
Step Smoothing	The size of the windows along the path used to calculate an average for each data point on a path profile. The setting is useful for averaging out noise.  If **Step Smoothing** is set to 0, no averaging is performed (only the data point under the path is used). For averaging perpendicular to the path, use **Path Width** (see above).
Step Width	The distance, along a path profile, separating the points used to find steps on a path profile.  The setting is useful when you must detect a slope as an edge, rather than a sharply defined edge: setting **Step Width** to a value greater than the width of the edge ensures that the tool measures the height difference between the flat regions on either side of the edge. As a result, the height of the step is accurately measured, and the edge is correctly located. Setting **Step Width** wider than necessary can reduce the precision of edge location.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Max Gap</td><td>Fills in regions of missing data caused by an occlusion near the desired edge. Use this setting when continuity on the target is expected. When Max Gap is set to a non-zero value, the tool holds and extends the last data point on the low side next to an edge across a gap of null points, up to the distance specified in Max Gap.

</td></tr>
<tr><td>Include Null Edges</td><td>Indicates whether null points (points where no height or intensity value is available, due to dropouts or regions outside of the measurement range) are filled with the value in Null Fill Value as a general “background level.” If Use Intensity (see above) is enabled, the intensity value in Intensity Null Fill Value is also used.

A typical example is a discrete part produced by part detection of an object sitting on a flat background. The background is not visible in the part, so the tool assumes that any null region are at the background level.

To find edges along a region of null points, you must use either this option and an appropriate value in Null Fill Value (and Intensity Null Fill Value if Use Intensity is enabled) or Max Gap. Otherwise, only edges within areas of contiguous data will be detected.</td></tr>
<tr><td>Null Fill Value</td><td>The height value (in mm) used to replace null points not filled by Max Gap when Include Null Edges is enabled.</td></tr>
<tr><td>Intensity Null Fill Value</td><td>The intensity value (0-255) used to replace null points when Include Null Edges and Use Intensity are enabled.</td></tr>
<tr><td>Show Detail</td><td>When disabled, hides the light blue path lines and edge points.</td></tr>
<tr><td>Filters</td><td>The filters that are applied to measurement values before they are output. For more information, see Filters on page 167.</td></tr>
<tr><td>Decision</td><td>The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see Decisions on page 166.</td></tr>
</table>

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

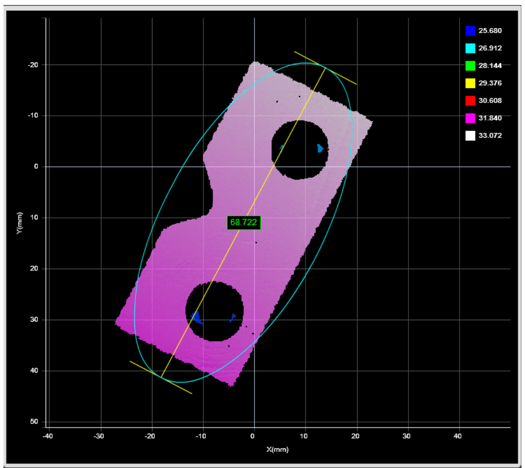
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

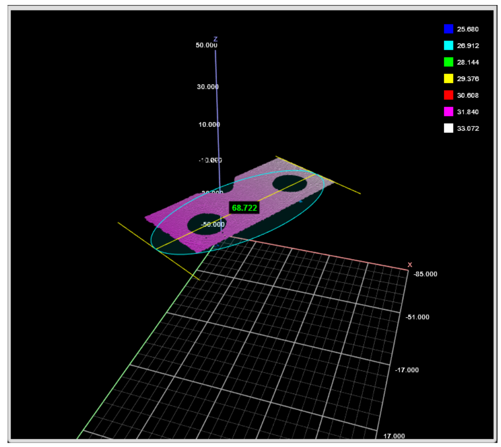
Ellipse

The Ellipse tool provides measurements for the major and minor axis lengths of an ellipse roughly aligned to the part's shape in the XY plane, and also for the ratio of the major and minor axis lengths and for the orientation angle of the ellipse. The tool is typically used to find the general orientation of a part, for example, potatoes on a conveyor that are longer in one dimension than the other.

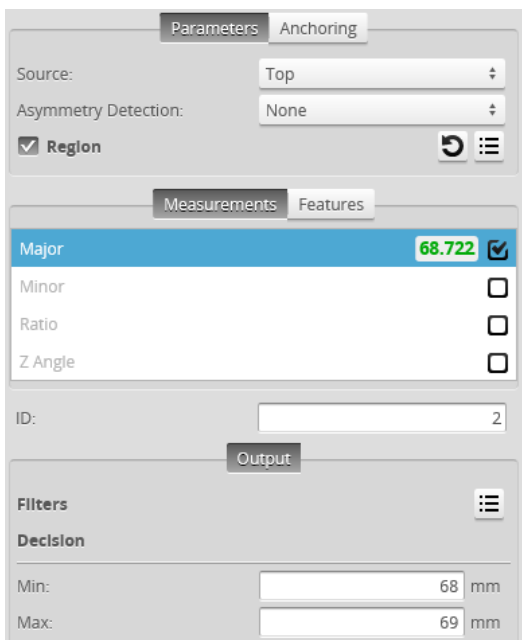
Note that the ellipse fit is not the minimum area ellipse around the data. (Technically, it is the ellipse with matching moments as the data.) For surfaces with no holes, this results in an ellipse approximately the same size and orientation of the part. But for surfaces with holes, the resulting ellipse can be larger than the part.



2D View



3D View

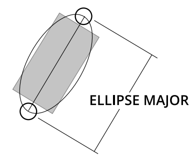
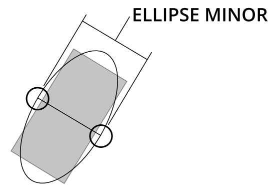
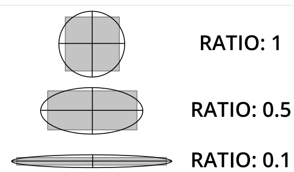
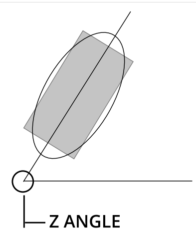


Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
Major Determines the major axis length of an ellipse fitted to the part's area in the XY plane.	
Minor Determines the minor axis length of an ellipse fitted to the part's area in the XY plane.	
Ratio Determines the minor/major axis ratio of an ellipse fitted to the part's area in the XY plane.	
Z Angle Determines the orientation angle of an ellipse fitted to the part's area in the XY plane.	

Features

Type	Description
Center Point	The center point of the fitted ellipse.
Major Axis	A line representing the major axis of the fitted ellipse.
Minor Axis	A line representing the minor axis of the fitted ellipse.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Asymmetry Detection	Resolves the orientation of an object over 360 degrees. The possible values are: 0 – None 1 – Along Major Axis 2 – Along Minor Axis
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

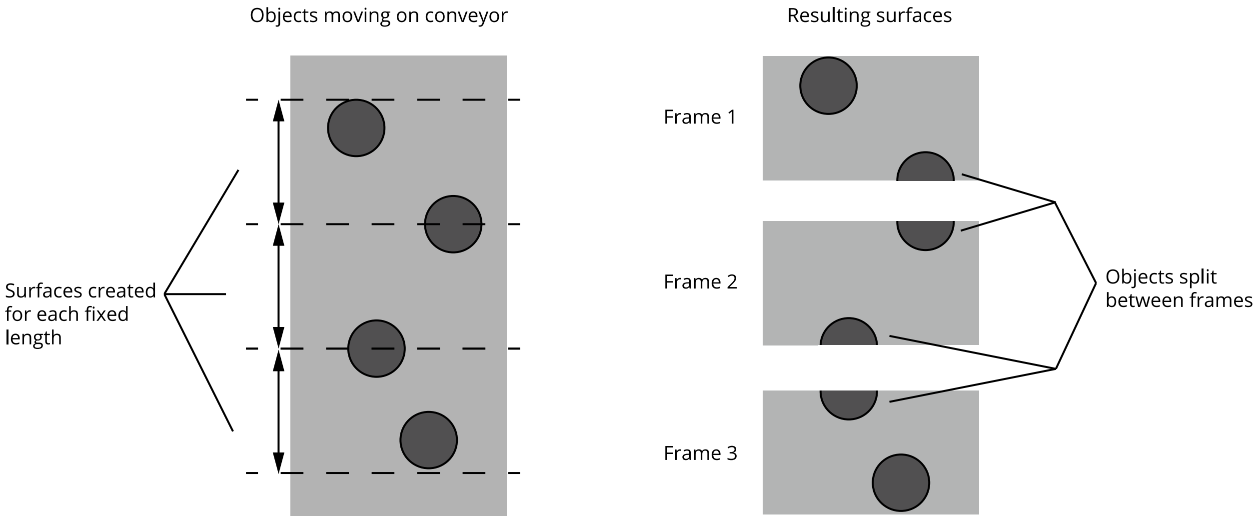
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

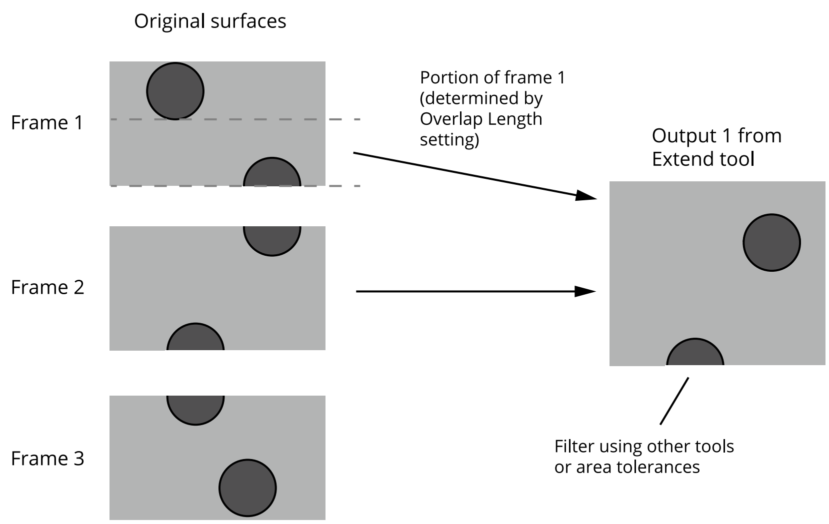
Extend

🗋	This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

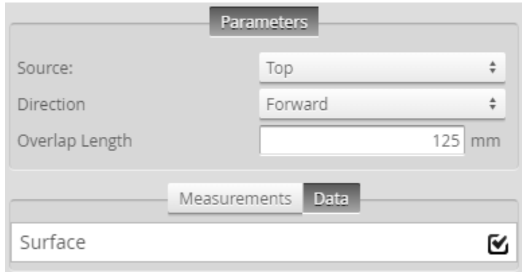
The Extend tool creates a new surface by appending part of the previous frame's data to the current frame's data. The tool outputs the new surface data, which can be used as input by other tools.



The following shows how the tool combines data:



Data is only appended in one direction. Partial objects in the resulting surface output from the tool must be filtered out using downstream tools, for example, excluding them based on the expected area.



Measurement Panel

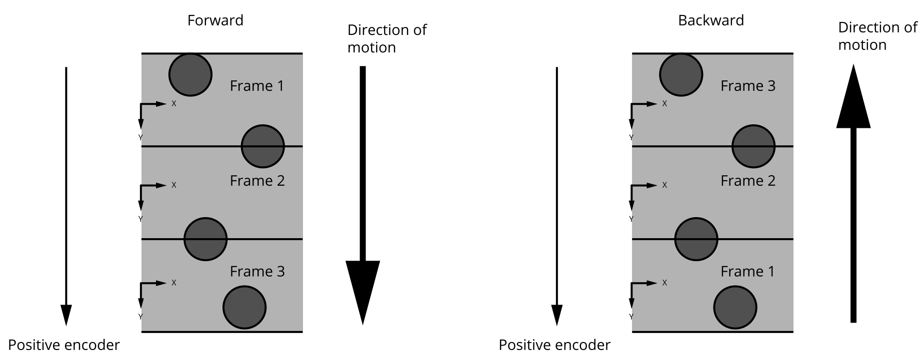
For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Data and Settings

Data

Type	Description
Extended Surface	Data containing an extended surface, available for use as input in the **Stream** drop-down in other tools.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Direction	Determines whether the previous frame's data is appended above or below the current frame's data.  One of the following. Note that these settings depend on whether the trigger source has been set to Encoder (see *Trigger Settings* on page 90). • **Auto**: Choose this when Encoder is selected as the trigger source, in which case the tool will know the direction of travel relative to encoder increase / decrease. • **Forward**: Choose this when the trigger source is not set to Encoder and the direction of motion is the same as the increase of the encoder. • **Backward**: Choose this option when the trigger source is not set to Encoder and the direction of motion is the opposite of the increase of the encoder.

Parameter	Description
Overlap Length	The amount, in millimeters, of the previous frame's data to append to the current frame's data. The combination will be output as tool data. Choose the overlap length to accommodate the size of your scan targets.
Mode	Determines the mode of the tool. One of the following: • **Normal**: The tool automatically chooses this operation after you have chosen another operation. • **Lock**: Lets you lock the current processing and outputs of the tool. Useful when you need to add another tool that will use this tool's output (for example, a Surface Section tool). If you do not lock the tool, as soon as you add the other tool, the output is cleared, which means you must re-execute the combined output again to configure the additional tool. Be sure to unlock the tool after you have configured any other tools.

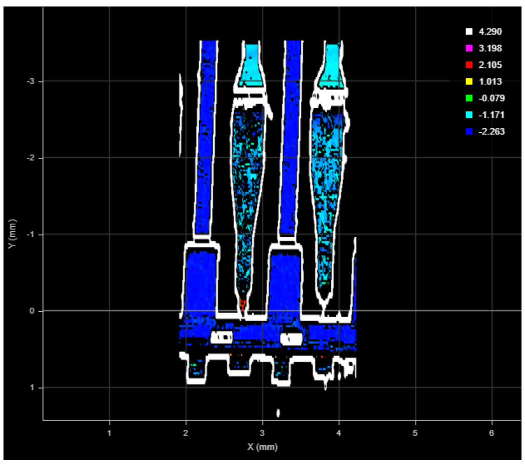
Filter

> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

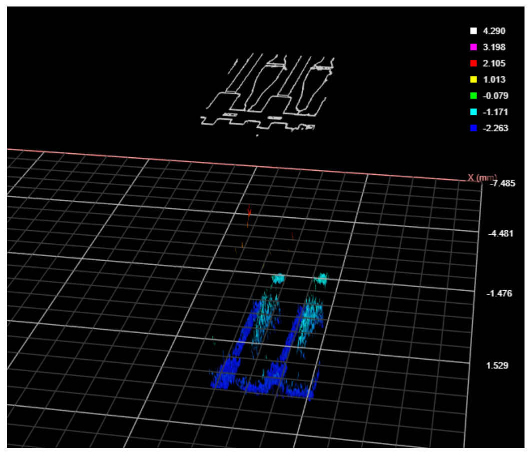
The Filter tool provides several common vision processing filters that you can apply to surface data, as well as a two "cropping" filters that output a subset of the surface data, letting you pre-process scan data to get more repeatable measurements. You can enable up to seven of the filters at once, in any order. Filters in the tool are chained together. Any Surface or Feature tool can use the resulting filtered surface data as input, via the tool's **Stream** drop-down.

For a list of the filters, see *Filters* on page 377.

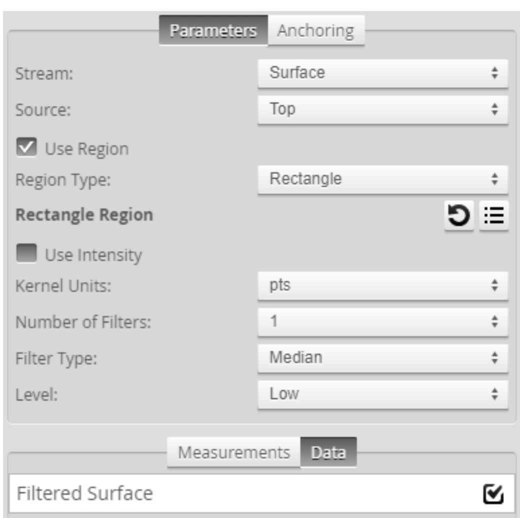
The Filter tool provides no measurements or decisions, as its only purpose is to output processed surface data.



2D View (Sobel Magnitude)



3D View (Sobel Magnitude)



Tool Setup

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Settings and Available Filters

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151. Can only accept Surface scan data (that is, cannot accept data from other tools).
Region	Only displayed on older instances of this tool. Newer instances use "flexible regions" (see the parameters below in this table). The region whose data the tool will apply filters to. Only data within the region is output to other tools.
Use Region	When enabled, displays additional settings to let you set a region (see below).
Number of Regions Region Type {n} Inner Circle Diameter Inner Ellipse Major Axis Inner Ellipse Minor Axis Sector Start Angle Sector Angle Range Mask Source Low Threshold High Threshold	Only displayed on newer instances of this tool. When you enable **Use Region**, the tool displays additional settings related to the measure region type. For details on flexible regions and their settings, see *Flexible Regions* on page 153. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 152.
Use Intensity	If enabled, the tool uses intensity data instead of heightmap data. Only available if Acquire Intensity is enabled on the Scan page during the scan; for more information, see *Scan Modes* on page 88.
Kernel Units	Specifies whether filters use data points (**pts**) or millimeters (**mm**).
Number of Filters	Specifies the number of filters you want to chain together. You can specify up to seven filters.
Filter Type	For each filter, specifies the type of filter. For more information on the available filters, see *Filters* on the next page.

Parameter	Description
Level	The kernel size used by the Median filter. High is a 5x5 square kernel. Low is a 3x3 square kernel.
Threshold	The threshold that the filter uses. (Not available on all filters.)
Symmetry	One of the following: Symmetrical, Horizontal, or Vertical. (Not available on all filters.)
Kernel Size	The kernel size that the filter uses. (Not available on all filters.)

The following filters are available in the Filter tool.

Filters

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Median</td><td>A median filter.</td></tr>
<tr><td>Gaussian</td><td>A Gaussian filter.</td></tr>
<tr><td>Open</td><td>Erosion followed by dilation.</td></tr>
<tr><td>Close</td><td>Dilation followed by erosion.</td></tr>
<tr><td>Erode</td><td>Applies an erosion filter. Lets you specify the direction of the erosion; one of the following:
• Horizontal
• Vertical
• Symmetrical</td></tr>
<tr><td>Dilation</td><td>Applies a dilation filter. Lets you specify the direction of the dilation; one of the following:
• Horizontal
• Vertical
• Symmetrical</td></tr>
<tr><td>Morph Gradient</td><td>Applies a morphological gradient. The difference between dilation and erosion.</td></tr>
<tr><td>Sobel Magnitude</td><td>Applies a Sobel magnitude filter.
Lets you specify the direction of the filter; one of the following:
• Horizontal
• Vertical
• Symmetrical</td></tr>
<tr><td>Laplacian</td><td>Applies a Laplacian filter. Useful for detecting areas of distinct edges. Uses the following kernel:
<table><tr><td>0</td><td>-1</td><td>0</td></tr><tr><td>-1</td><td>4</td><td>-1</td></tr><tr><td>0</td><td>-1</td><td>0</td></tr></table></td></tr>
<tr><td>Negative</td><td>Inverts the height or intensity values in the scan data.</td></tr>
<tr><td>Equalize</td><td>Normalizes the norm or value range of an array.</td></tr>
</table>

Name	Description
Binarize	Sets height values to a fixed value for each point that is present in the data. Can be used with a region Z offset to threshold points above/below a Z value. With intensity data, sets any point over
Percentile	Limits the scan data to points between the values you set in **High Percentile** and **Low Percentile**, which are displayed when you choose this option.
Relative Threshold	Crops scan data based on user-specified minimum and a maximum heights. Use **Reference Region** to set the heights relative to a reference region.
Crop only	Crops the scan data to the user-defined region.
Mask With Input	Uses the surface input into the tool as a mask on the data. Any points in the filtered data will be set to null if the input surface is null at the same location. For example, the Gaussian filter can extend data along the edges, adding data in areas that contain null values. This filter would remove data that the Gaussian filter introduces, preserving the null values. This filter should follow any filter that introduces this kind of unwanted data.

Data

Type	Description
Filtered Surface	The filtered data, available for use as input in the **Stream** drop-down in other tools.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Flatness

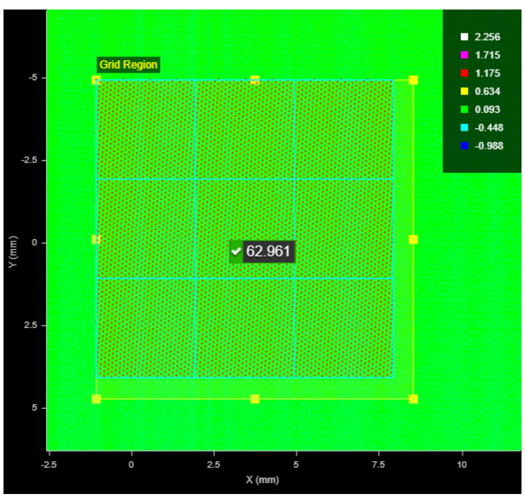
> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Flatness tool returns various measurements related to the flatness of one or more regions on the surface of your target. The tool is ideal for general fit and finish inspection.

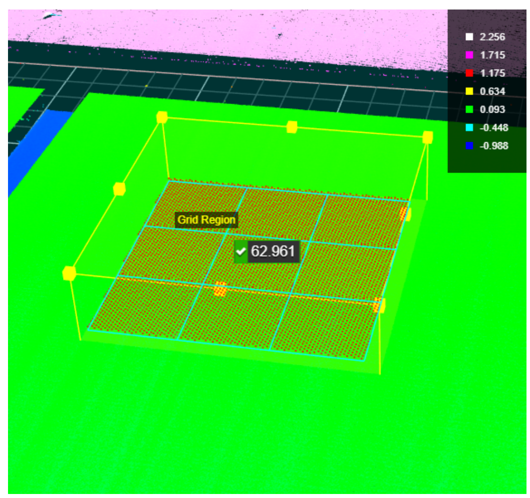
The tool lets you set a grid over a specific region, or more flexibly with multiple individual regions manually. In each case, "local" minimum and maximum heights, as well as flatness indicators (maximum - minimum), are returned (for grid cells or individual regions, depending on the tool's settings). In addition, "global" minimum, maximum, and flatness measurements, that combine data from all flatness measurement areas, can also be returned. The tool measures the maximum and minimum distances from a different best-fit plane for each local measurement, and from another plane fit to all data for the "global" measurements.

You can control how many data points the tool uses in its calculations to account for noise or smooth data, or otherwise exclude unwanted data.

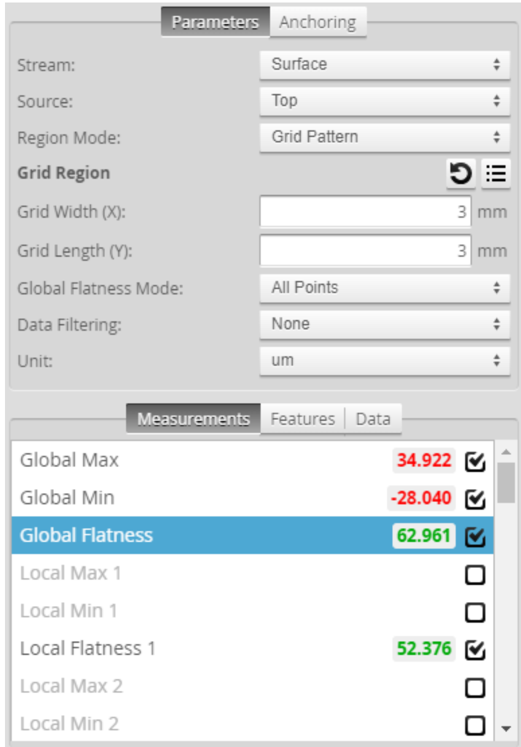
> When you configure the tool to use a grid that contains more than 15 cells, only the first 15 local measurements (which correspond to the first 15 cells of the grid) are displayed in the web interface. Flatness results for cells beyond 15 are however available in the tool data.



2D View



3D View



Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, Data, and Settings

Measurements

Measurement
Global Max **Global Min** **Global Flatness** The maximum distance, minimum distance, and flatness (maximum - minimum) calculated using the valid data points from *all* the cells in the grid (when **Region Mode** is set to **Grid Pattern**), or *all* the individual regions (when **Region Mode** is set to **Flexible**).
Local Max {n} **Local Min {n}** **Local Flatness {n}** The maximum distance, minimum distance, and flatness (maximum - minimum) calculated using the valid data points from a specific grid cell (when **Region Mode** is set to **Grid Pattern**), or an individual regions (when **Region Mode** is set to **Flexible**). Clicking a local measurement in the list of measurements selects the corresponding cell or region in the data viewer.

Features

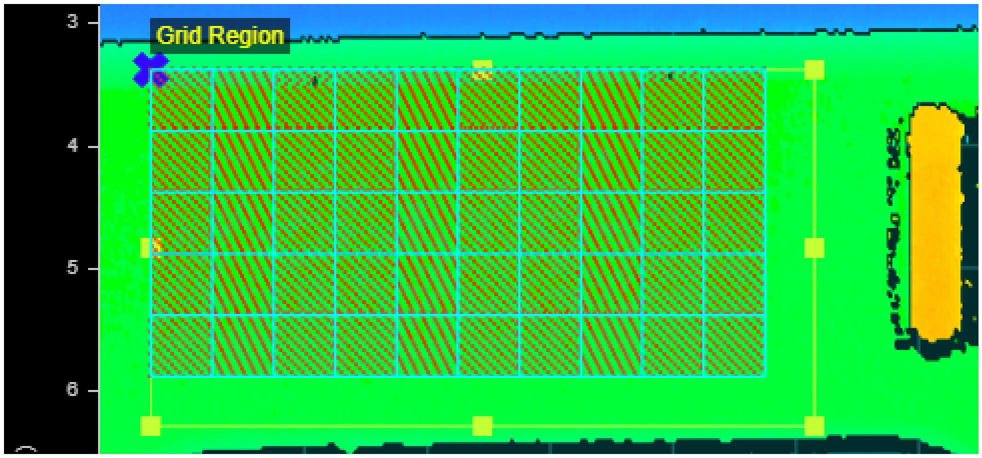
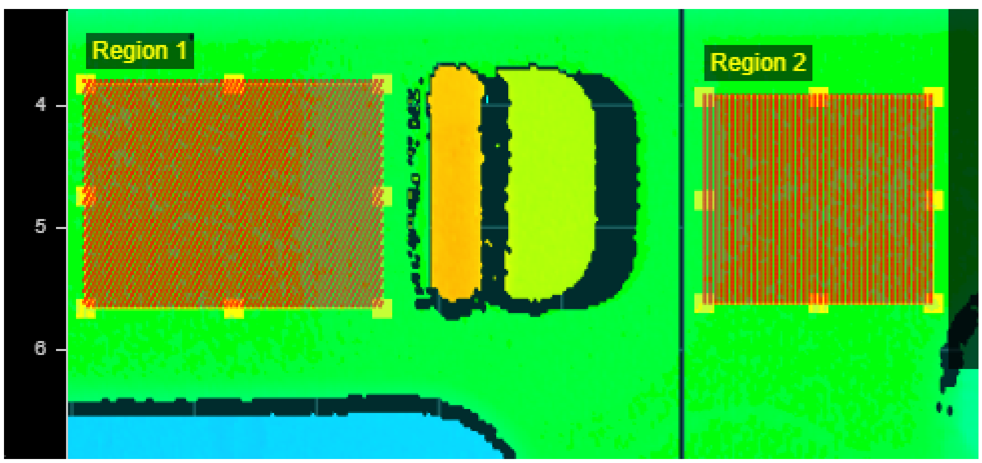
Type	Description
Global Plane	The plane fitted to the valid data points from *all* the cells in the grid (when **Region Mode** is set to **Grid Pattern**), or *all* the individual regions (when **Region Mode** is set to **Flexible**).
Local Plane {n}	The plane fitted to the valid data points from grid cell {n} (when **Region Mode** is set to **Grid Pattern**), or those from region {n} (when **Region Mode** is set to **Flexible**). Clicking a local plane in the list of features selects the corresponding cell or region in the data viewer.

Data

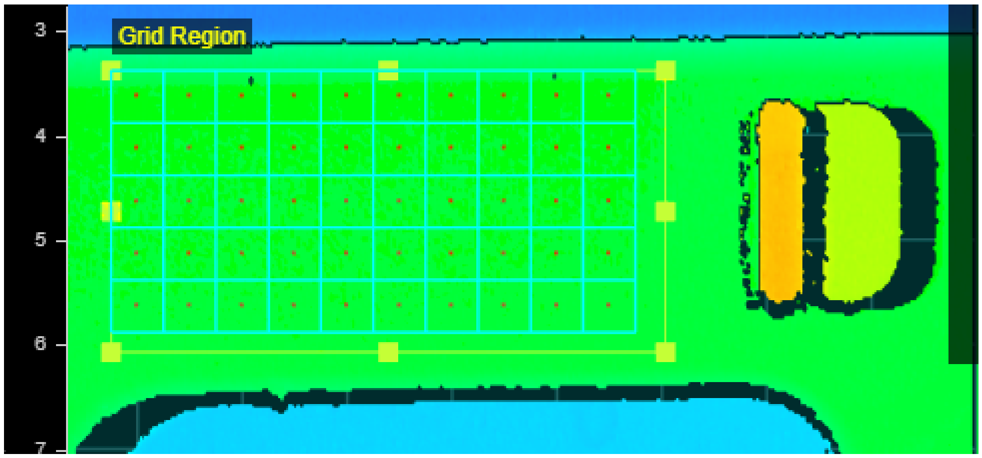
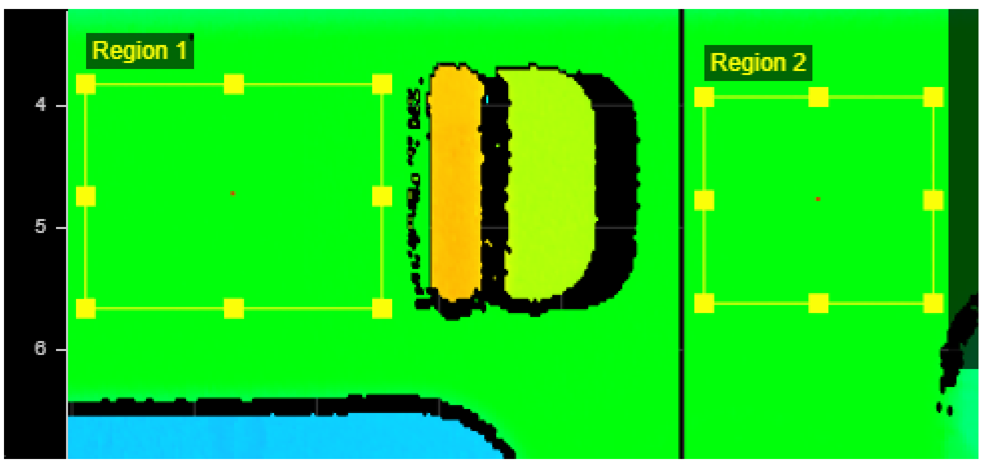
Type	Description
Output Measurement	Data containing the measurement results. The web interface only displays up to 15 local measurements. However, if you define the grid and cell size so that you have more than 15 flatness measurement areas, these are included in the tool data. A sample included in the SDK package shows how you can use this output data in an application.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.

Parameter	Description
Region Mode	Determines how flatness measurement areas are set up on the target. One of the following: **Grid Pattern**: The tool determines flatness in a grid you define on the target. This option enables settings that let you set the size and location of a region that contains the grid (**Grid Region** setting), as well as the width and length of the grid cells (**Grid Width** and **Grid Length**). The combination of the values of these settings determines the number of cells in the grid region.  **Flexible**: The tool determines flatness using one or more (up to 15) regions that you define individually on the target. 
Grid Region *(used with Grid Pattern region mode)*	Determines the size of the grid region. (See details under **Grid Pattern** in **Region Mode** above.)
Grid Width (X) Grid Length (Y) *(used with Grid Pattern region mode)*	These settings determine the size of the cells in the grid. (See details under **Grid Pattern** in **Region Mode** above.)

Parameter	Description
Region Number *(used with Flexible region mode)*	Only displayed on older instances of this tool. Newer instances use "flexible regions" (see the parameters below in this table). The number of regions.
Region {n} *(used with Flexible region mode)*	Only displayed on older instances of this tool. Newer instances use "flexible regions" (see the parameters below in this table). When **Region Mode** is set to **Flexible**, for each region, the tool displays a region definition.
Number of Regions Region Type {n} Inner Circle Diameter Inner Ellipse Major Axis Inner Ellipse Minor Axis Sector Start Angle Sector Angle Range Mask Source Low Threshold High Threshold	Only displayed on newer instances of this tool. When you enable **Use Region**, the tool displays additional settings related to the measure region type. For details on flexible regions and their settings, see *Flexible Regions* on page 153. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 152.

Parameter	Description
Global Flatness Mode	Chooses which points the tool uses to calculate global flatness. One of the following: **All Points**: The tool uses all points in the measurement area (all flexible regions or the grid pattern in the region). **Single Average Point**: The tool uses an average of the points in the measurement area. When you choose this option, the global measurements require at least four data points to calculate the plane and statistics. This means that if you set **Region Mode** to **Flexible**, you must choose a minimum of four regions; if you set Region Mode to Grid Pattern, the size of the grid and the cells must result in at least four cells.  
Data Filtering	Lets you filter scan data before the tool performs its calculations. **Percentile** - Limits the data to points between the values you set in **High Percentile** and **Low Percentile**, which are displayed when you choose this option. **None** - The tool performs no filtering.
Unit	Lets you choose which units the tool uses for measurement results. One of the following: • um (micrometers) • mm (millimeters)

Parameter	Description
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

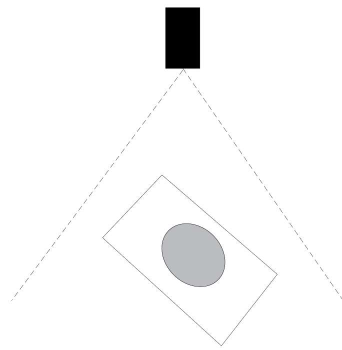
Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Hole

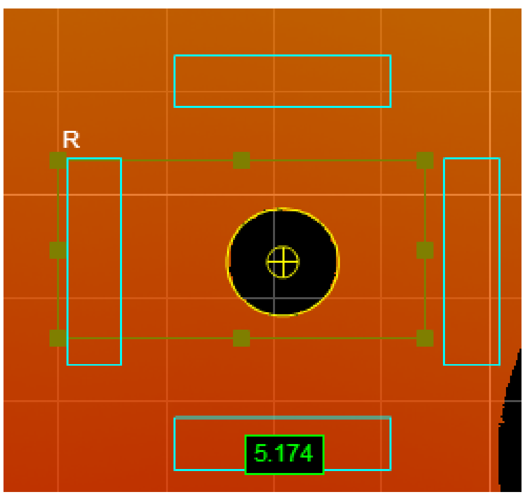
The Hole tool measures a circular opening within a region of interest on the surface and returns its position and radius.



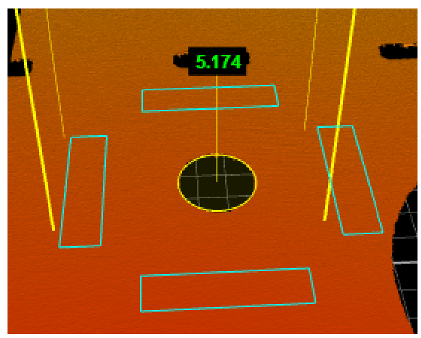
> The tool does not search for or detect the feature. The tool expects that the feature, conforming reasonably well to the defined parameters, is present and that it is on a sufficiently uniform background.

The hole can be on a surface at an angle to the sensor.

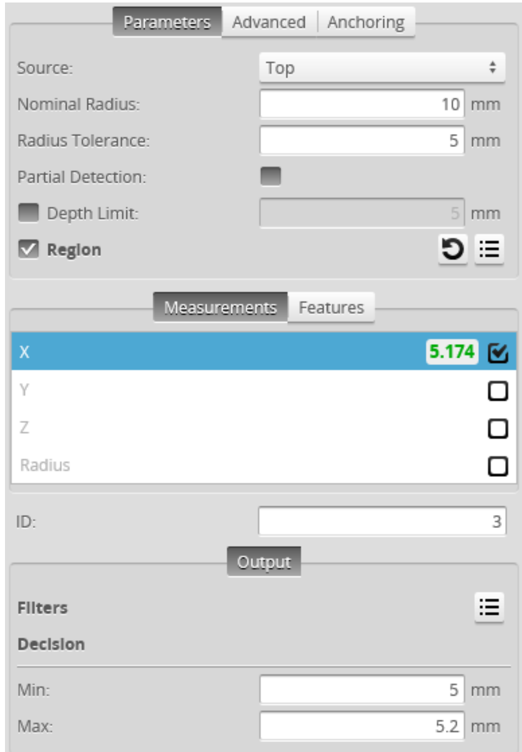
The tool uses a complex feature-locating algorithm to find a hold and then return measurements. See "Hole Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel.

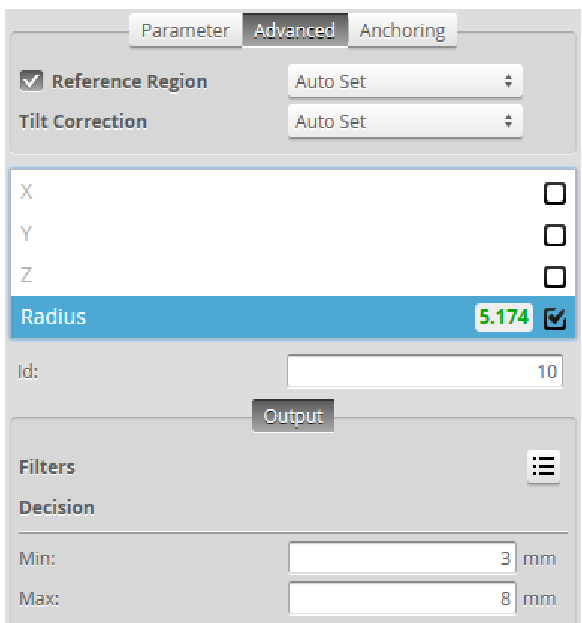


2D View



3D View



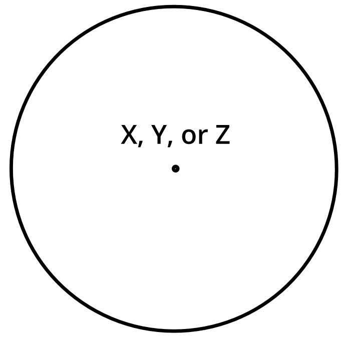
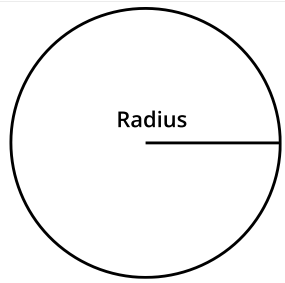


Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Determines the X position of the hole center. **Y** Determines the Y position of the hole center. **Z** Determines the Z position of the hole center.	
Radius Determines the radius of the hole.	

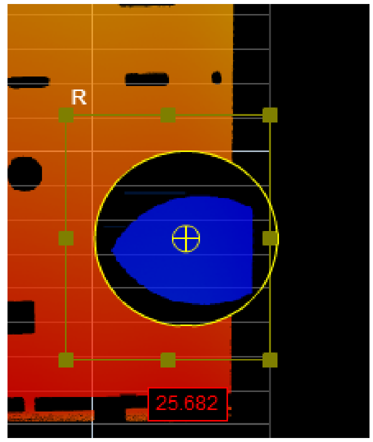
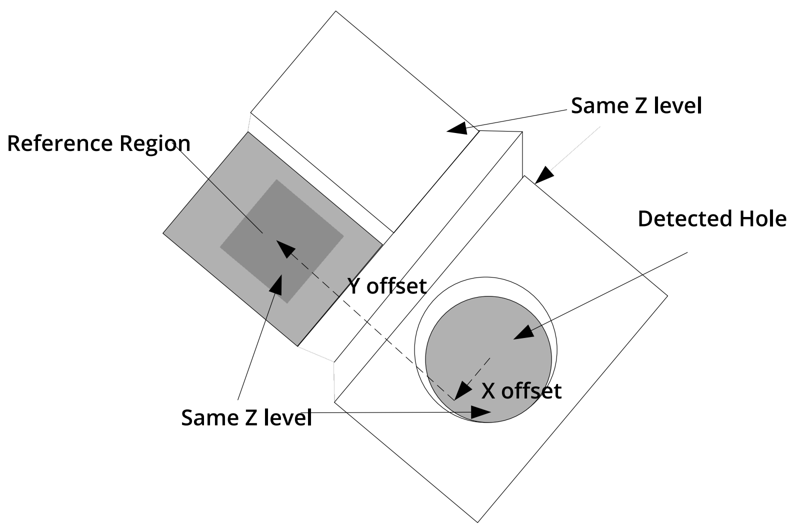
Features

Type	Description
Center Point	The center point of the hole. The Z position of the center point is at the Z position of the surrounding surface.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Nominal Radius	Expected radius of the hole.
Radius Tolerance	The maximum variation from the nominal radius (+/- from the nominal radius).

Parameter	Description
Partial Detection	Enable if only part of the hole is within the measurement region. If disabled, the hole must be completely in the region of interest for results to be valid. 
Depth Limit	Data below this limit (relative to the surface) is excluded from the hole calculations.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Reference Region	The tool uses the reference regions to calculate the Z position of the hole. It is typically used in cases where the surface around the hole is not flat.  When this option is set to **Autoset**, the algorithm automatically determines the reference region. When the option is not set to **Autoset**, you must manually specify one or two reference regions. The location of the reference region is relative to the detected center of the hole and positioned on the nominal surface plane. When **Reference Region** is disabled, the tool measures the hole's Z position using all the data in the measurement region, except for a bounding rectangular region around the hole.

Parameter	Description
Tilt Correction	Tilt of the target with respect to the alignment plane. **Autoset**: The tool automatically detects the tilt. The measurement region to cover more areas on the surface plane than other planes. **Custom**: You must enter the X and Y angles manually in the X Angle and Y Angle parameters (see below).
X Angle Y Angle	The X and Y angles you must specify when **Tilt Correction** is set to **Custom**. You can use the Surface Plane tool's X Angle and Y Angle measurements to get the angle of the surrounding surface, and then copy those measurement's values to the **X Angle** and **Y Angle** parameters of this tool. For more information, see Plane.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

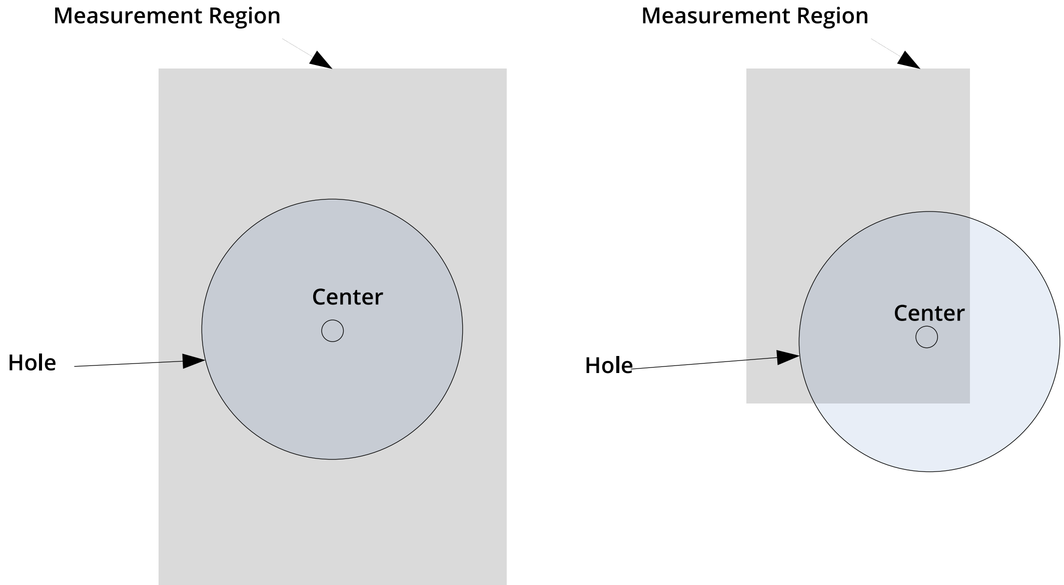
Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Measurement Region

The center of the hole must be inside the measurement region, even if the Partial Detection option is enabled.

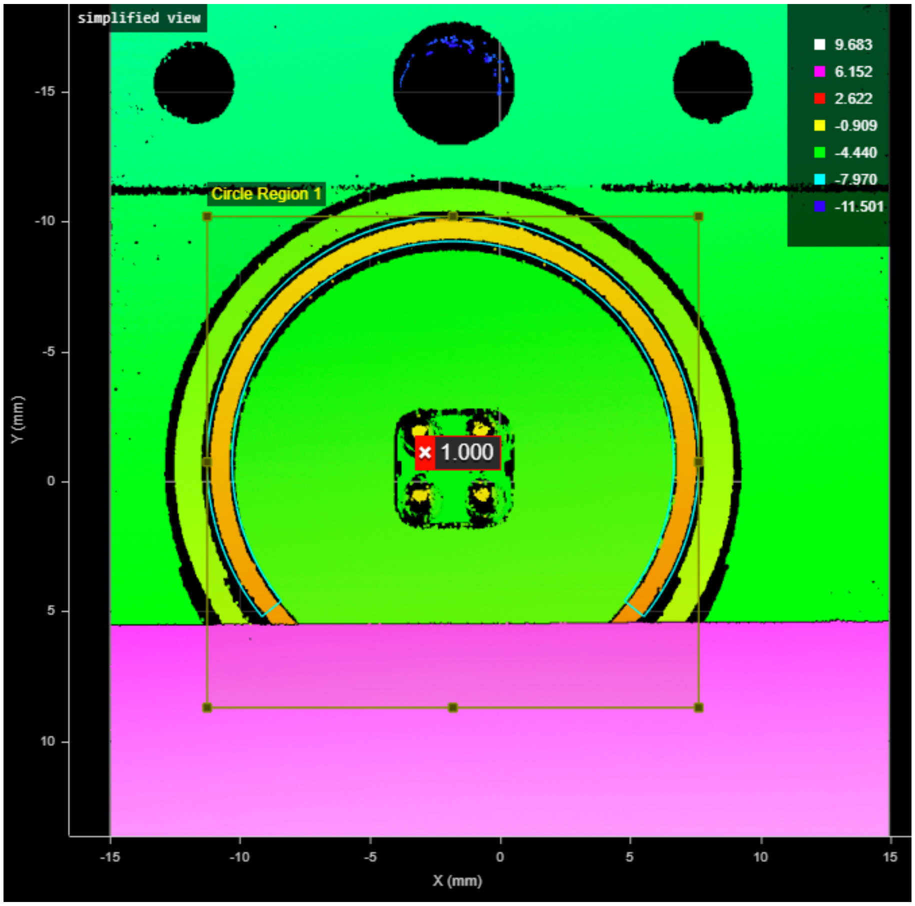


Mask

The Surface Mask tool lets you extract up to 16 regions of data from a surface. Each region's size, position, and shape (circular, elliptical, polygonal, and rectangular) can be individually configured, and regions can overlap. The tool can also exclude inner data of circular and elliptical regions, letting you extract rings of surface data.

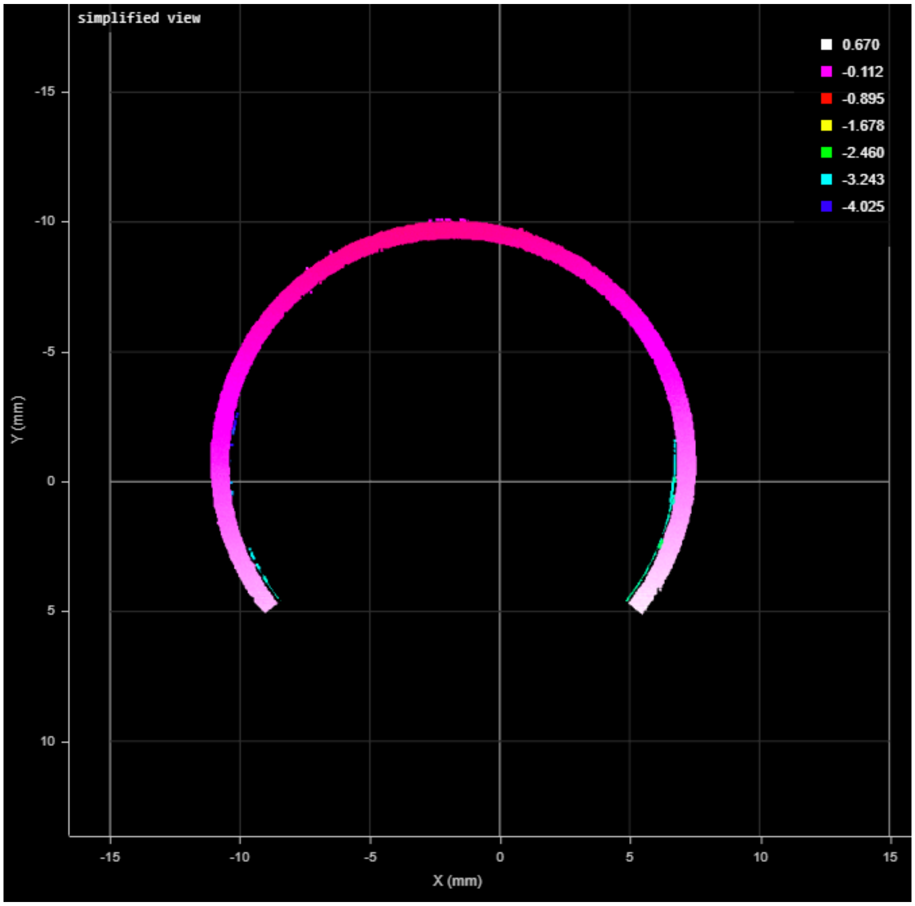
The resulting surface can then be further processed or measured by other tools.

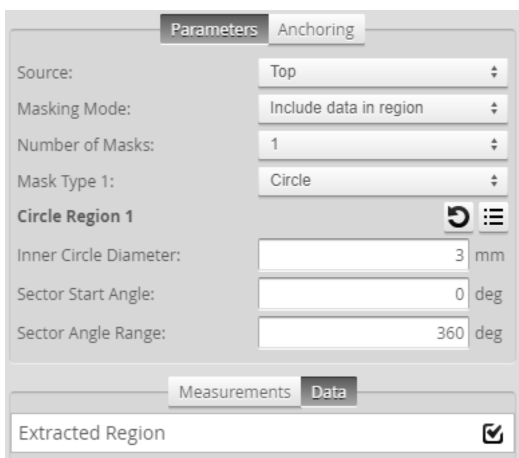
For example, given the following scan data:



A circle region box containing a partial ring (cyan)

The image below shows the extracted data. The extracted surface data can then be further processed by other tools, or measurements can be applied to the surface data.





Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements and Settings

Measurements

Measurement
Processing Time The amount of time the tool takes to process.

Data

Type	Description
Extracted Region	The surface containing the extracted region or regions.

Parameters

Parameter	Description
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool. If you switch from one type of data to another (for example, from section profile data to surface data), currently set input features will become invalid, and you will need to choose features of the correct data type.
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Masking Mode	The masking mode the tool uses. One of the following: **Include data in region**: Data in the mask is included **Exclude data in region**: Data in the mask is excluded.

Parameter	Description
Number of Masks Mask Type {n} / Region Type {n} Inner Circle Diameter Inner Ellipse Major Axis Inner Ellipse Minor Axis Sector Start Angle Sector Angle Range Mask Source Low Threshold High Threshold	When you enable **Use Region**, the tool displays additional settings related to the measure region type. For details on flexible regions and their settings, see *Flexible Regions* on page 153. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

OCR

This tool requires GoMax or PC-based acceleration.

For more information on GoMax, see the GoMax user manual.

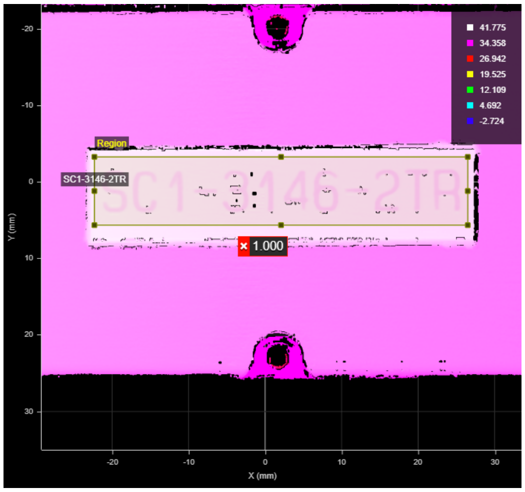
For more information on PC-based acceleration, see *Gocator Acceleration* on page 529.

The Surface OCR (optical character recognition) tool lets you extract a string of text from surfaces, using either heightmap or intensity scan data. The tool is font-independent and already trained. The tool therefore lets you implement OCR without the need for a separate 2D camera system.

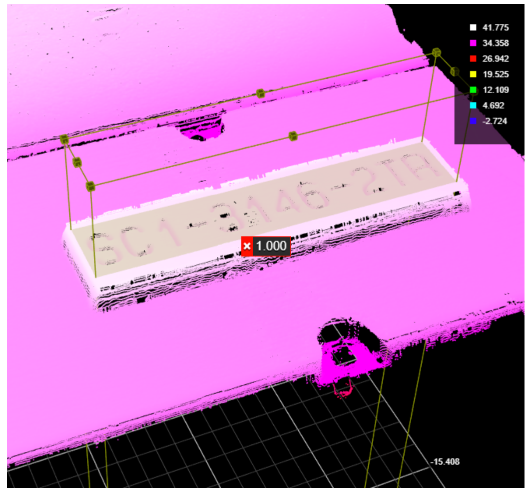
You can use the String Encoding tool to extract the string and pass it as output to a PLC; for more information, see *String Encoding* on page 442. The extracted string is also available via the SDK; for information on the SDK, see *GoSDK* on page 836 and the SDK reference documentation.

The tool does not support multi-line character recognition, and the text must be rotated so that it is human-readable from left to right along the X axis.

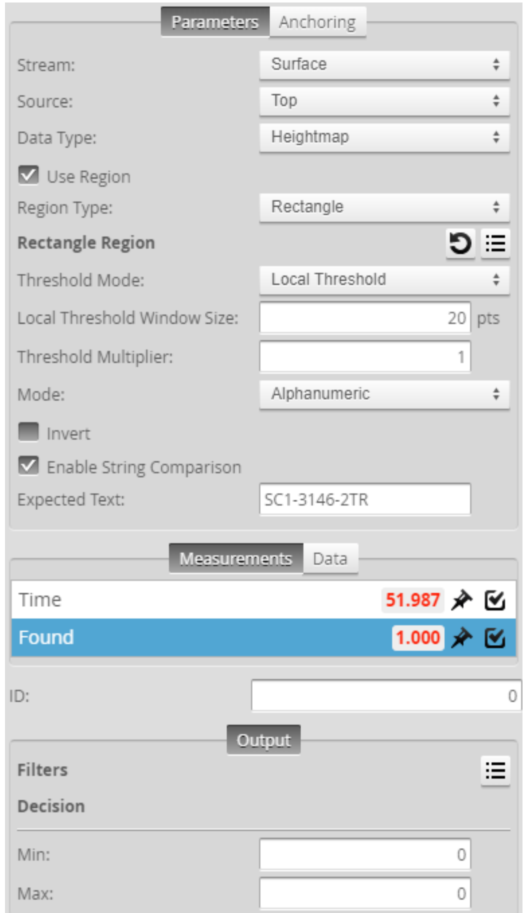
	When configuring the tool, use the Diagnostic Image data output, on the **Output** tab, to help set the thresholding parameters correctly.



2D View



3D View



Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements and Settings

Measurements

Measurement
Time The amount of time the tool takes to process.
Found Whether the extracted text is identical to the text in **Expected Text**.

Data

Type	**Description**
Diagnostic Image	The data the tool uses to perform optical character recognition.
Output String	A string containing the recognized text. (This data is not currently visualized in the data viewer.)

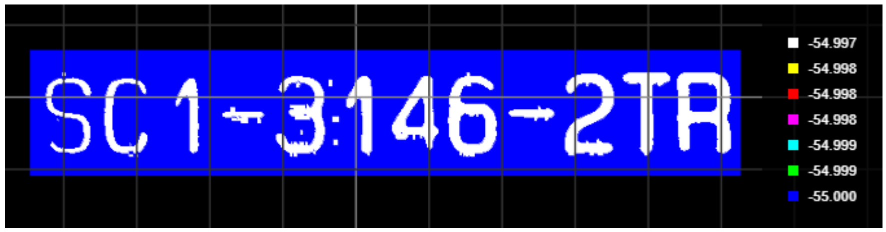
Parameters

Parameter	Description
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool.
Source	The sensor that provides data for the tool's measurements.
Data Type	The type of data the tool uses (**Heightmap** or **Intensity**).
Use Region	Only displayed on older instances of this tool. Newer instances use "flexible regions" (see the parameters below in this table). Indicates whether the tool uses a user-defined region. If this option is not checked, the tool uses data from the entire active area.
Region	Only displayed on older instances of this tool. Newer instances use "flexible regions" (see the parameters below in this table). The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Number of Regions Mask Type {n} / Region Type {n} Inner Circle Diameter Inner Ellipse Major Axis Inner Ellipse Minor Axis Sector Start Angle Sector Angle Range Mask Source Low Threshold High Threshold	Only displayed on newer instances of this tool. When you enable **Use Region**, the tool displays additional settings related to the measure region type. For details on flexible regions and their settings, see *Flexible Regions* on page 153. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 152.

Parameter	Description
Threshold Mode	Determines the threshold the tool uses to identify characters relative to the background data. One of the following: **Default** – The default used by tesseract with OTSU adaptive thresholding method. Use this mode if the scan data has been pre-processed to remove any tilt of the surface on which you want to perform OCR, for example using Surface Transform; for more information, see *Transform* on page 465. **Local Threshold** – The tool varies the threshold for each pixel based on the minimum and maximum values within a moving window over the region, using the specified window size and multiplier (see below). This method can compensate for intensity and height gradients. **Manual Threshold** – The tool uses a single, fixed threshold for the entire region (see Manual Threshold below).
Local Threshold Window Size	The window size the tool uses for local thresholding. The window size should generally be larger than the size of the characters being detected. Displayed when **Threshold Mode** is set to **Local Threshold**.
Threshold Multiplier	The multiplier the tool uses for local thresholding. Typically set to a value close to 1. Displayed when **Threshold Mode** is set to **Local Threshold**.
Manual Threshold	The manual threshold the tool uses, expressed as a percentage, converted to a 0-255 range, relative to minimum and maximum values within the region. Displayed when **Threshold Mode** is set to **Manual Threshold**.
Mode	Limits the characters the tool will recognize. Choose the mode based on the expected types of characters in the target. One of the following: **Alphanumeric** – Only attempts to recognize alphanumeric characters. **Numeric** – Only attempts to recognize numeric characters. **Whitelist** – Only attempts to recognize the characters in the **Whitelist** parameter that this option displays. **Blacklist** – Will not attempt to recognize characters in the **Blacklist** parameter that this option displays.
Whitelist Blacklist	The whitelist or blacklist of characters that the tool will attempt to recognize or ignore, respectively. These parameters are case sensitive. The list of characters is a simple string of characters. One of these parameters is displayed when **Mode** is set to **Whitelist** or **Blacklist**.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Invert</td><td>Swaps intensity values in the data the tool uses to perform OCR. Use this if you need to perform OCR on light text on a dark background. (The OCR library the tool uses expects dark text on a light background.)

For heightmap data, the tool swaps the "high" and "low" values. For example, in the second image below, the height values used for the text and the surrounding surface (the highest and lowest values in the heightmap legend to the right) are swapped when compared to the first, non-inverted data.



This parameter is not available when Threshold Mode is set to Default.</td></tr>
<tr><td>Enable String Comparison</td><td>Enables string comparison.</td></tr>
<tr><td>Expected Text</td><td>The string the tool compares the extracted text to. The parameter is case-sensitive and does not support wild cards or truncation.</td></tr>
<tr><td>Filters</td><td>The filters that are applied to measurement values before they are output. For more information, see Filters on page 167.</td></tr>
<tr><td>Decision</td><td>The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see Decisions on page 166.</td></tr>
</table>

Anchoring

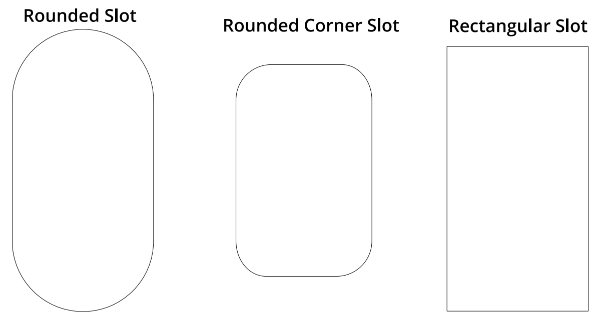
Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Opening

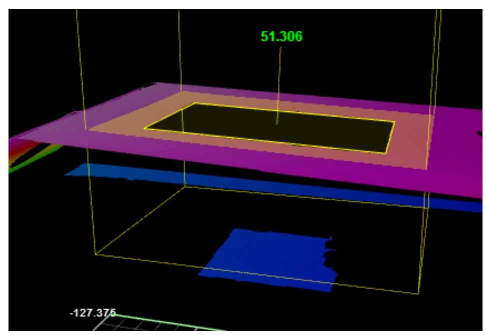
The Opening tool locates rounded, rectangular, and rounded corner openings. The opening can be on a surface at an angle to the sensor.



> The tool does not search for or detect the feature. The tool expects that the feature, conforming reasonably well to the defined parameters, is present and that it is on a sufficiently uniform background.

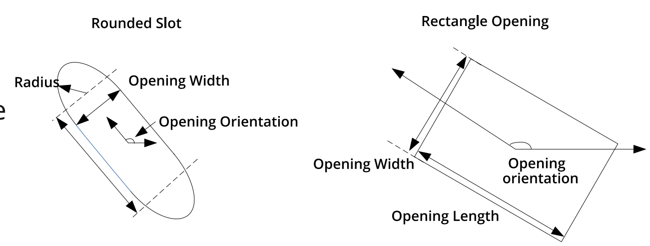
The tool uses a complex feature-locating algorithm to find a hold and then return measurements. See "Opening Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel.

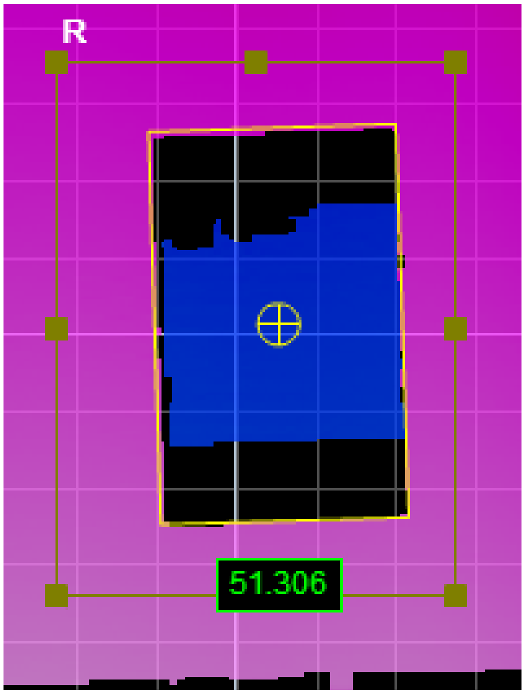
The algorithm can separate out background information that appears inside the opening. It can also detect a slot that only partially appears in the data.



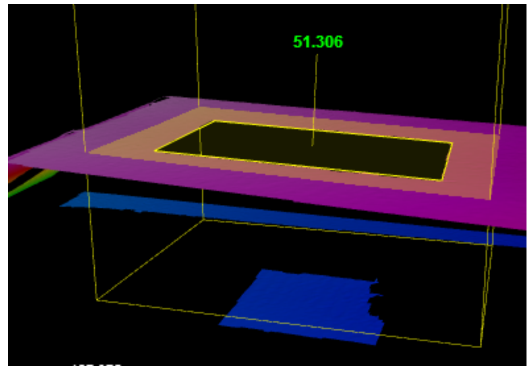
The shape of the opening is defined by its type and its nominal width, length, and radius.

The orientation defines the rotation around the normal of the alignment plane.

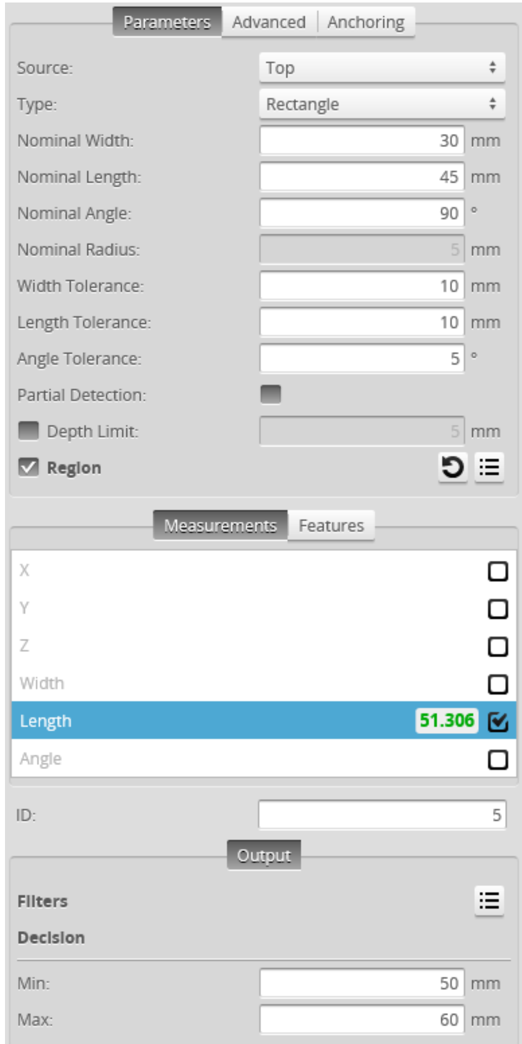


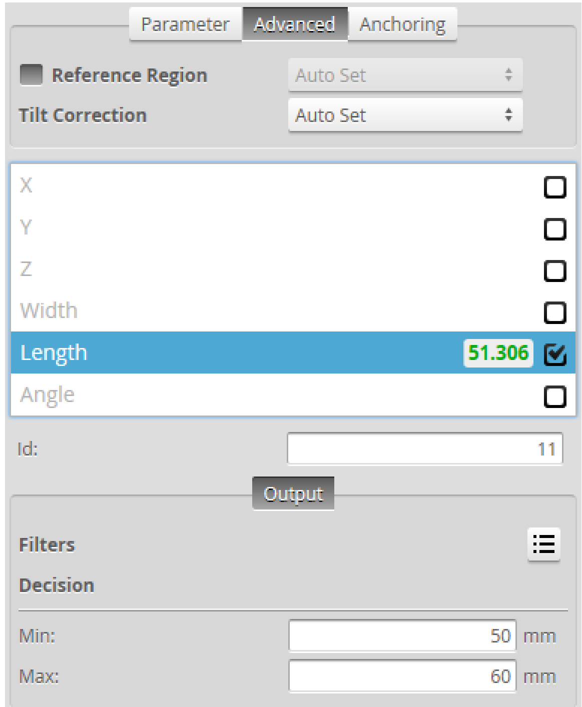


2D View



3D View


Parameters
Advanced
Anchoring
Source:
Top
Type:
Rectangle
Nominal Width:
30
mm
Nominal Length:
45
mm
Nominal Angle:
90
°
Nominal Radius:
5
mm
Width Tolerance:
10
mm
Length Tolerance:
10
mm
Angle Tolerance:
5
°
Partial Detection:
Depth Limit:
5
mm
Region
Measurements
Features
X
Y
Z
Width
Length
51.306
Angle
ID:
5
Output
Filters
Decision
Min:
50
mm
Max:
60
mm

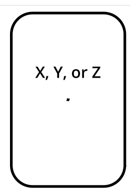
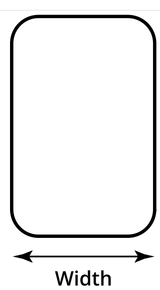


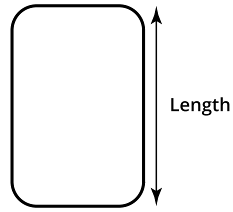
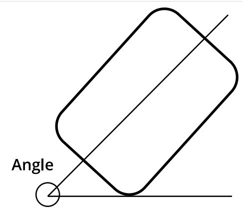
Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Determines the X position of the opening's center.	
Y Determines the Y position of the opening's center.	
Z Determines the Z position of the opening's center.	
Width Determines the width of the opening.	

Measurement	Illustration
Length Determines the length of the opening.	
Angle Determines the angle (rotation) around the normal of the alignment plane.	

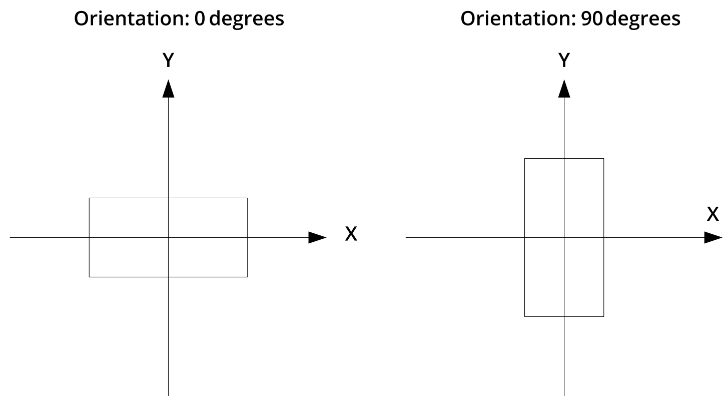
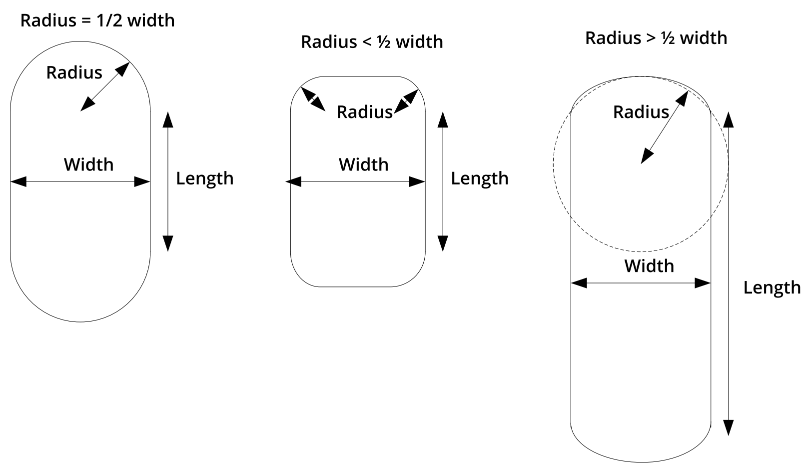
Features

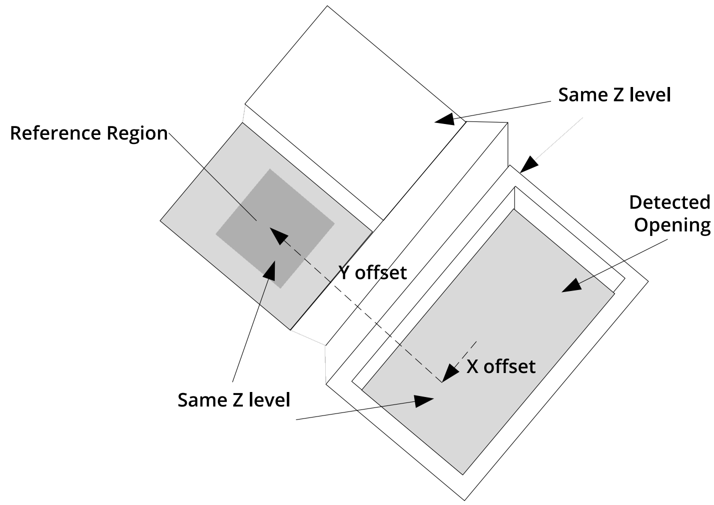
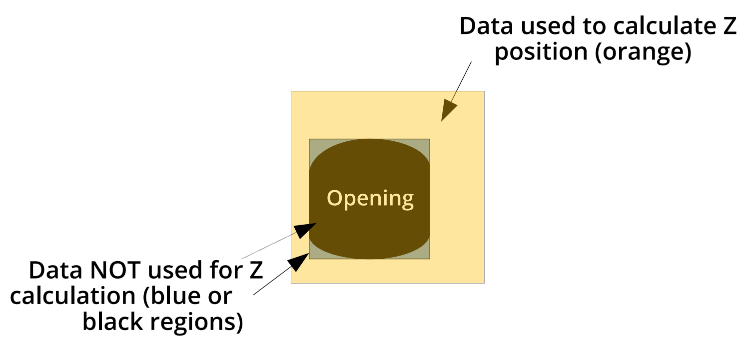
Type	Description
Center Point	The center point of the opening. The Z position of the center point is at the Z position of the surrounding surface.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Type	Rounded Slot, Rectangle.
Nominal Width	Nominal width of the opening.
Nominal Length	Nominal length of the opening.

Parameter	Description
Nominal Angle	Nominal angle of the opening. The default orientation is the length of the opening along the X axis.  The diagram above illustrates the case where the surface is not tilted. When the surface is tilted, the orientation is defined with respect to the normal of the surface, not with respect to the X-Y plane
Nominal Radius	Nominal radius of the opening ends. If the opening type is set to rectangular, the radius setting is disabled. The opening has an oval shape if the radius is equal to ½ of the width. The opening is a rounded rectangle when the radius is less than ½ of the width. 
Width Tolerance	The maximum variation from the nominal width (+/- from the nominal value).
Length Tolerance	The maximum variation from the nominal length (+/- from the nominal value).
Angle Tolerance	The maximum variation from the nominal orientation (+/- from the nominal value).
Partial Detection	Enable if only part of the opening is within the measurement region. If disabled, the opening must be completely in the region of interest for results to be valid.
Depth Limit	Data below this limit (relative to the surface) is excluded from the opening calculations.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.

Parameter	Description
Reference Regions	The tool uses the reference regions to calculate the Z position of the opening. Reference regions are relative to the center location of the feature. This option is typically used in cases where the surface around the opening is not flat.  When the Reference Regions setting is disabled, the tool measures the opening's Z position using the all data in the measurement region, except for a bounding rectangular region around the opening.  With one or more reference regions, the algorithm calculates the Z positions as the average values of the data within the regions. When you place the reference region manually, all of the data is used, whether the data is inside or outside the opening. You should place the reference region carefully.
Tilt Correction	Tilt of the target with respect to the alignment plane. **Autoset**: The tool automatically detects the tilt. The measurement region to cover more areas on the surface plane than other planes. **Custom**: You must enter the X and Y angles manually in the X Angle and Y Angle parameters (see below).

Parameter	Description
X Angle Y Angle	The X and Y angles you must specify when **Tilt Correction** is set to **Custom**. You can use the Surface Plane tool's X Angle and Y Angle measurements to get the angle of the surrounding surface, and then copy those measurement's values to the **X Angle** and **Y Angle** parameters of this tool. For more information, see Plane.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

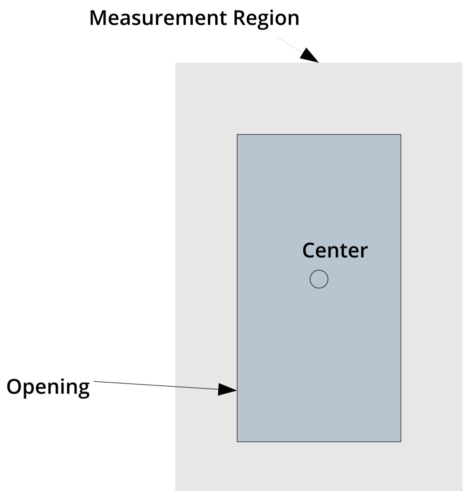
Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

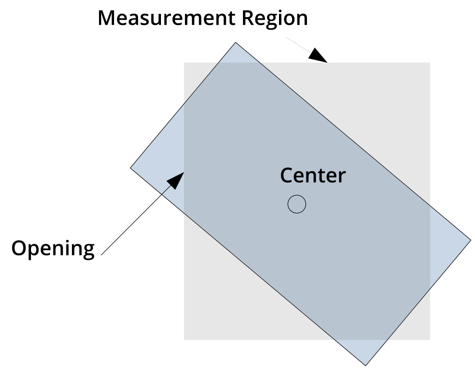
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Measurement Region

The center and the two sides and ends of the opening must be within the measurement region, even if **Partial Detection** is enabled.



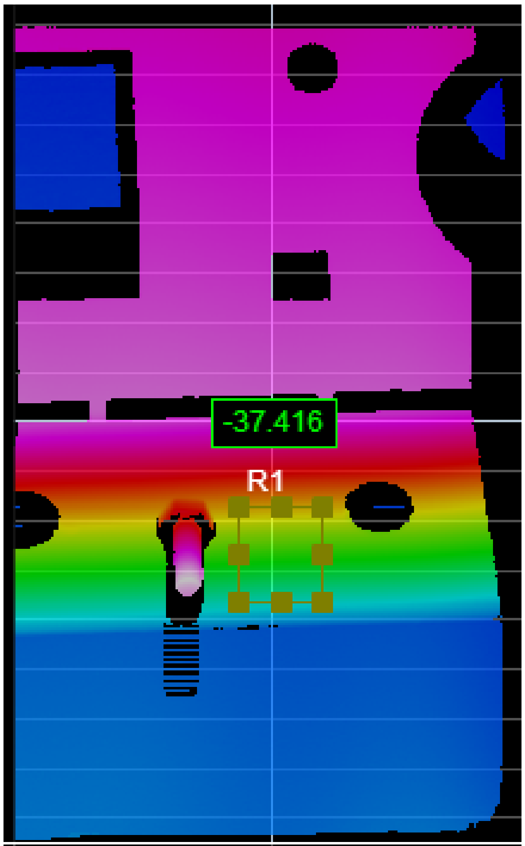


Plane

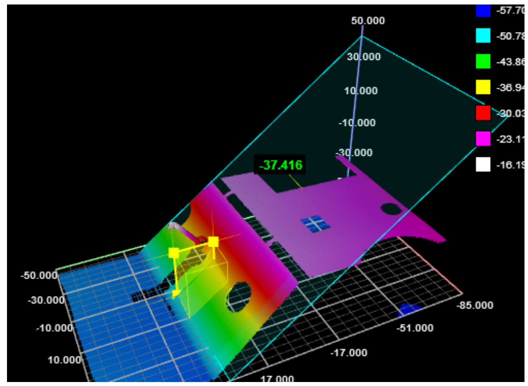
The Plane tool provides measurements that report a plane's position and orientation (X Angle, Y Angle, Z Offset, Normal, Distance), as well as the maximum and average deviations from the plane.

The Z offset reported is the Z position at zero position on the X axis and the Y axis.

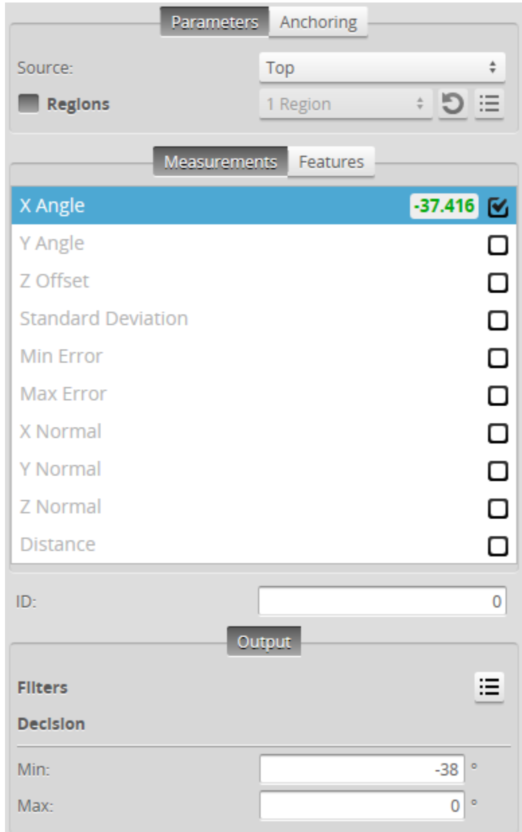
The results of the Angle X and Angle Y measurements can be used to manually customize the tilt angle in the Hole, Opening, and Stud tools.



2D View



3D View

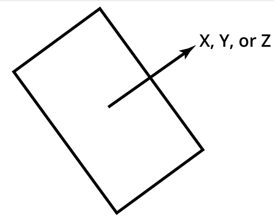
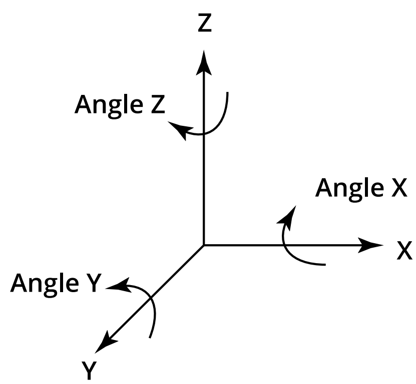


Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
Angle X Determines the X angle of the surface with respect to the alignment target.	
Angle Y Determines the Y angle of the surface with respect to the alignment target.	
Offset Z Determines the Z value of intersection of the plane and the Z axis.	
Standard Deviation Measures the standard deviation of the points of the surface from the detected plane within the specified region or regions.	
Min Error Measures the minimum error from the detected plane (the maximum distance below the plane, perpendicular to the plane) within the specified region or regions.	
Max Error Measures the maximum error from the detected plane (the maximum distance above the plane, perpendicular to the plane) within the specified region or regions.	
X Normal Returns the X component of the surface normal vector.	
Y Normal Returns the Y component of the surface normal vector.	
Z Normal Returns the Z component of the surface normal vector.	
Distance Distance from the origin to the plane.	

Features

Type	Description
Plane	The fitted plane.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Regions	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

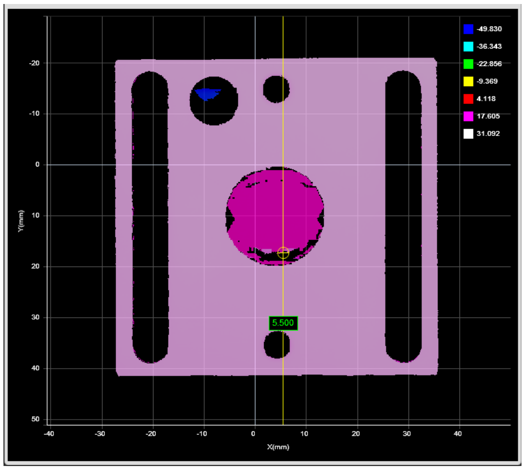
Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

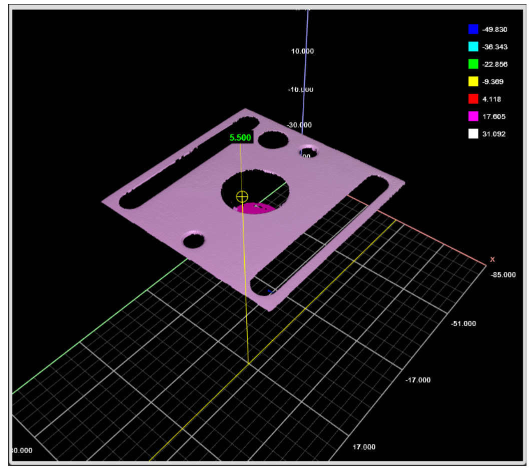
For more information on anchoring, see *Measurement Anchoring* on page 169.

Position

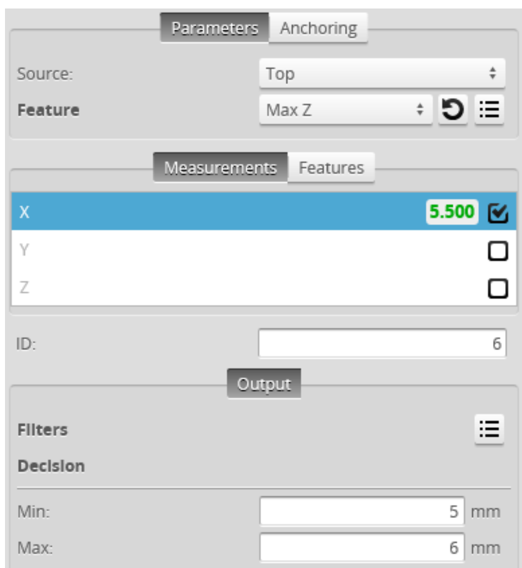
The Position tool reports the X, Y, or Z position of a part. The feature type must be specified and is one of the following: Average (the mean X, Y, and Z of the data points), Median (median X, Y, and Z of the data points), Centroid (the centroid of the data considered as a volume with respect to the z = 0 plane), Min X, Max X, Min Y, Max Y, Min Z, or Max Z.



2D View



3D View

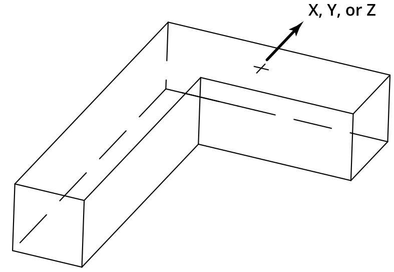


Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settingss

Measurements

Measurement	Illustration
X Determines the X position of the selected feature type.	
Y Determines the Y position of the selected feature type.	
Z Determines the Z position of the selected feature type.	

Features

Type	Description
Center Point	The returned position.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Feature	The feature the tool uses for its measurements. One of the following: • Average • Median • Centroid • Max X • Min X • Max Y • Min Y • Max Z • Min Z To set the region of a feature, adjust it graphically in the data viewer, or expand the feature using the expand button (☰) and enter the values in the fields. For more information on regions, see *Regions* on page 152.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

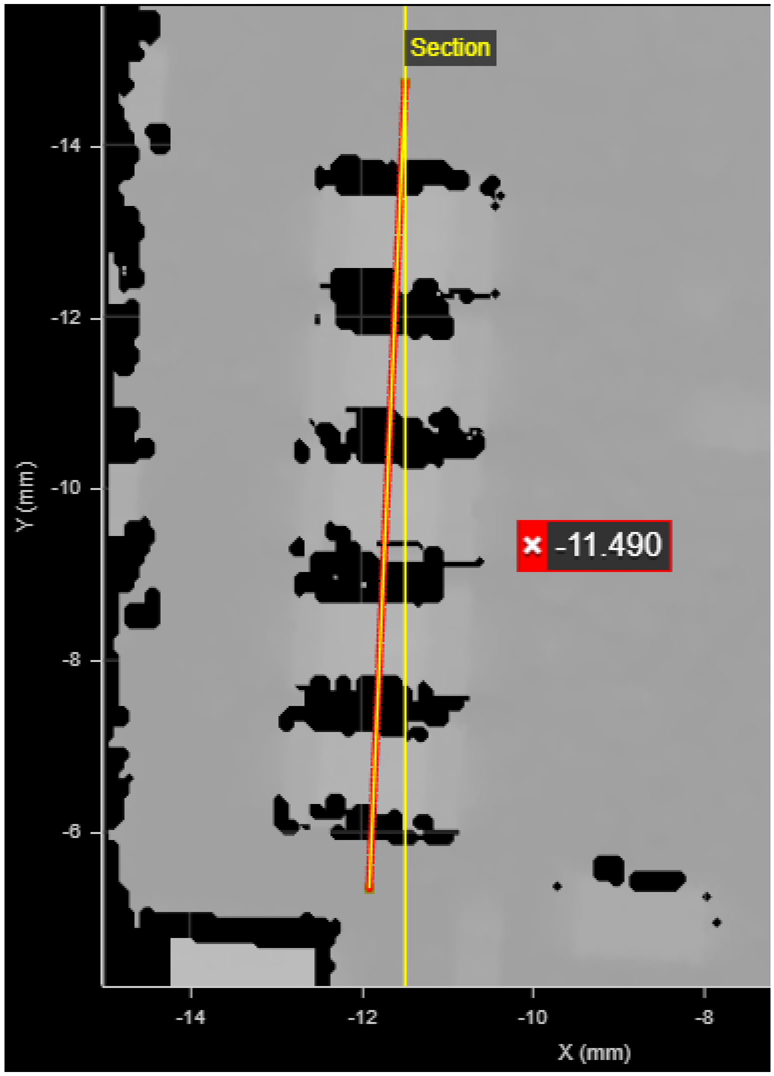
Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

> A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

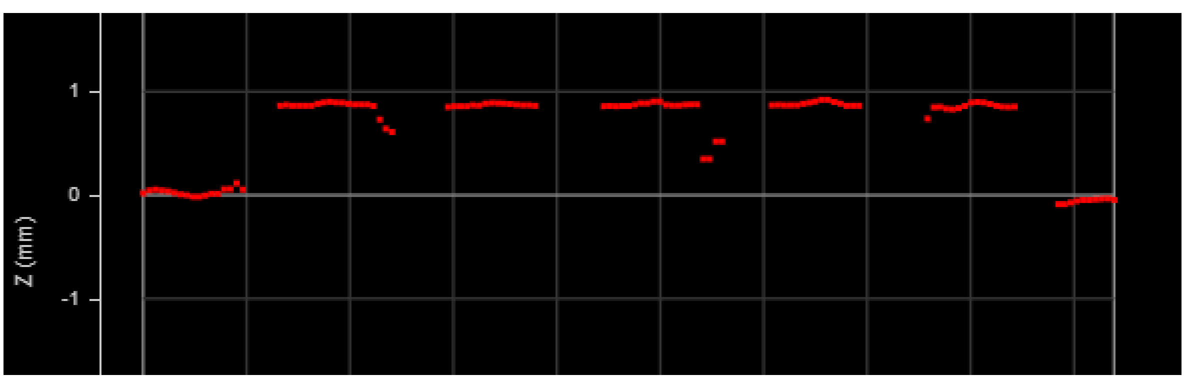
> For more information on anchoring, see *Measurement Anchoring* on page 169.

Section

The Surface Section tool lets you define a line on a surface (a "section") from which the tool extracts a profile. You can apply any Profile tool to the resulting profile (see *Profile Measurement* on page 206). Note that a section can have any XY orientation on the surface, but its profile is parallel to the Z axis.



A section over a row of components



The resulting profile

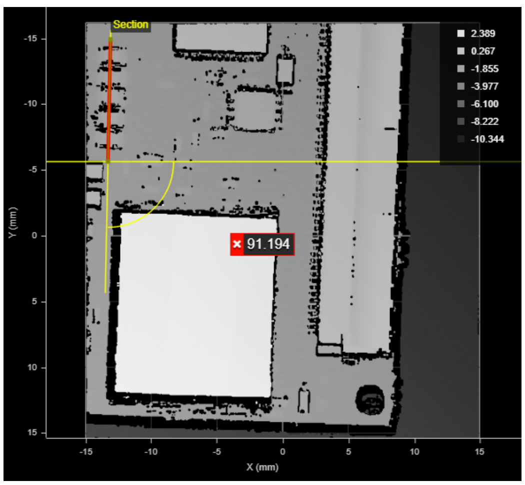
Note that profiles extracted from surfaces start at the point defined as the X/Y Start of the section. Profiles are always displayed horizontally, with X increasing to the right. The origin of extracted profiles is the beginning of the section, and not relative to the surface from which they are extracted.

The Surface Section tool provides functionality similar to sections you can define on the Models page (see *Models* on page 126). However, the Surface Section tool has a few advantages.

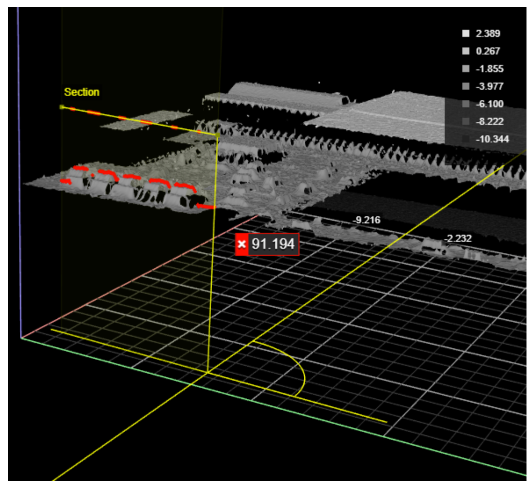
One advantage of the Surface Section tool is that you can anchor the tool to some other easily identifiable feature on the scan target, which "shifts" the section in relation to that feature: this increases repeatability.

Another advantage is that unlike sectioning generated from the Model page, the Surface Section can take any surface as input, such as a combined surface (using Surface Extend or Stitch), a transformed surface (using Surface Transformation), a filtered / corrected (Surface Filter and Surface Vibration Correction), and so on.

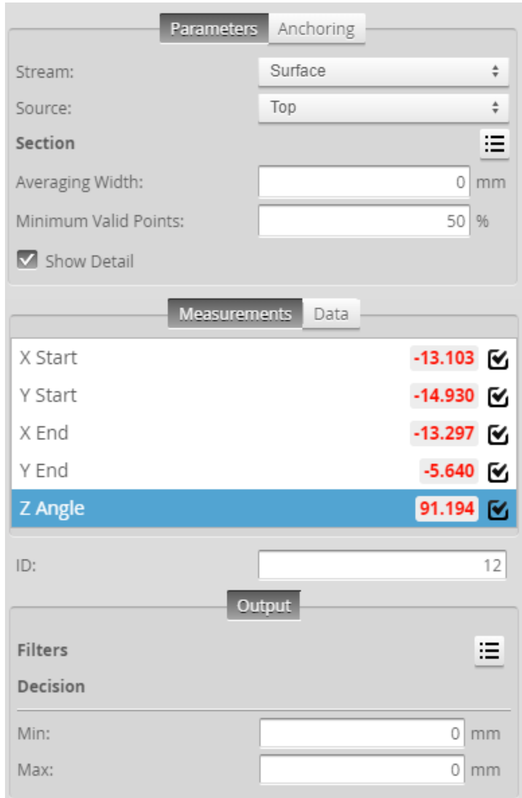
Finally, the Surface Section tool provides measurements useful for calculating the global X/Y coordinates of the resulting profile, using a Script tool Script (page 483). Even if you don't use anchors or the measurements, LMI recommends using the Surface Section tool over model-based sections.



2D View



3D View



Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Data, and Settings

Measurements

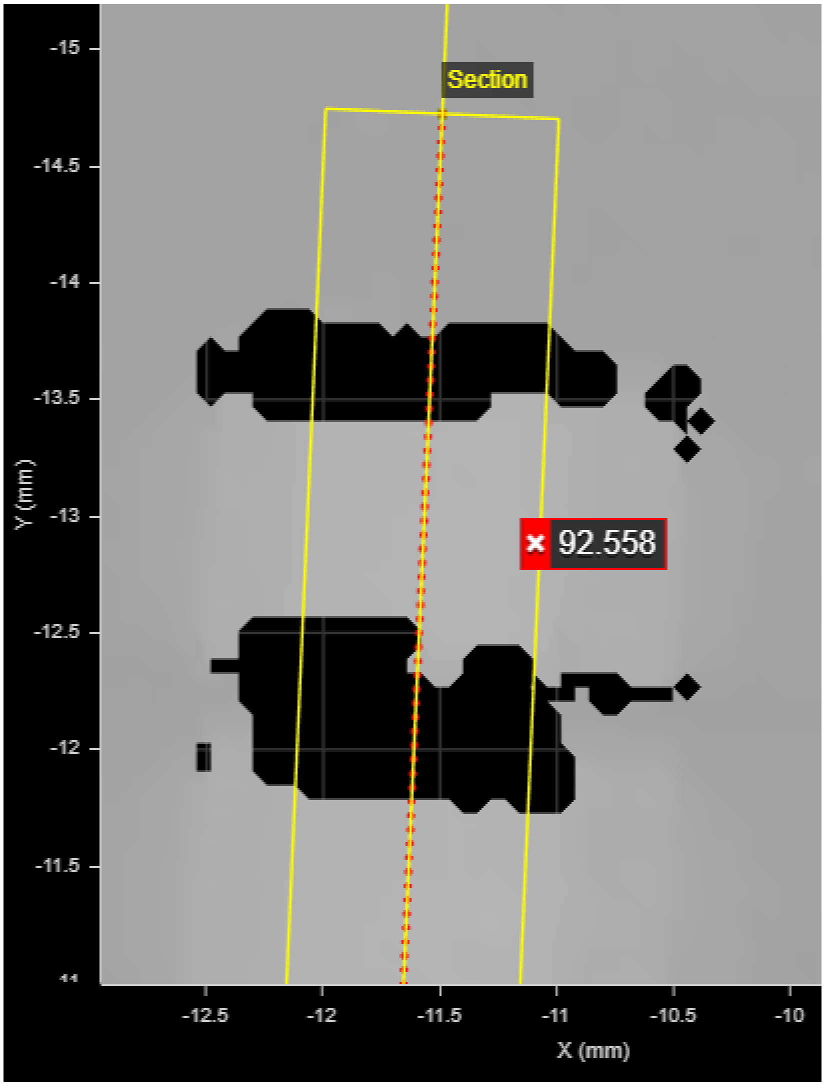
Measurement
X Start **Y Start** These measurements return the X and Y position of the start of the section, respectively.
X End **Y End** These measurements return the X and Y position of the end of the section, respectively.
Z Angle Returns the rotation of the section around the Z axis.

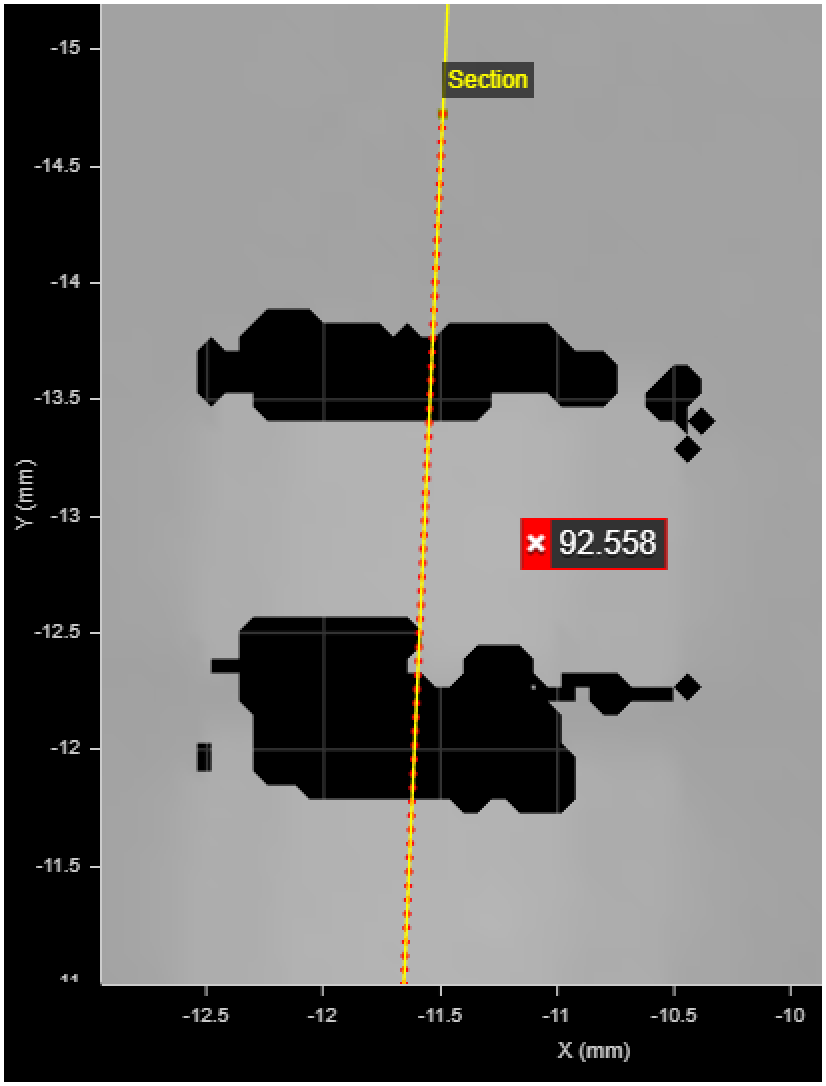
Data

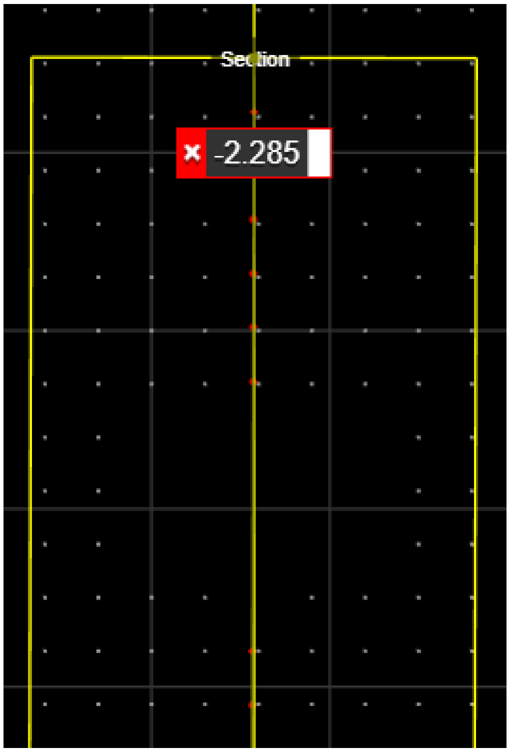
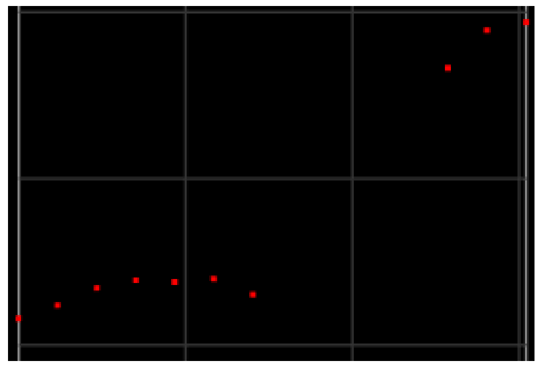
Type	**Description**
Profile	The profile the tool extracts from the surface. Available to profile tools for profile measurement.

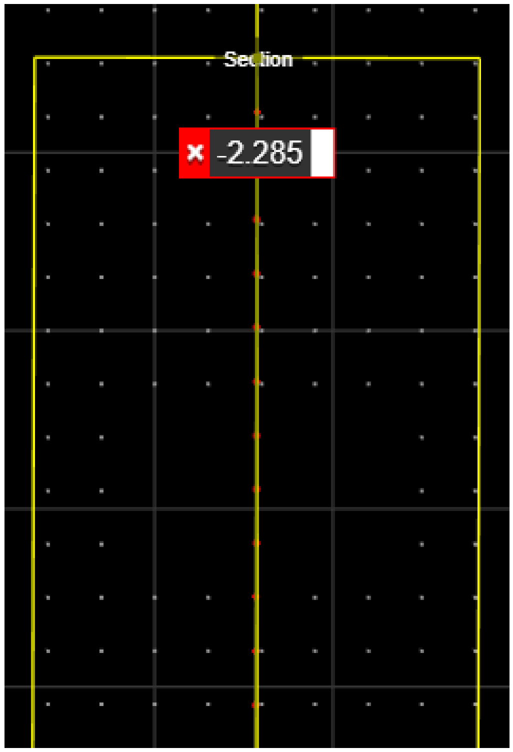
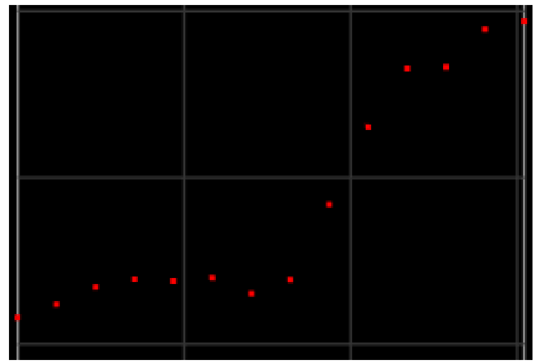
Parameters

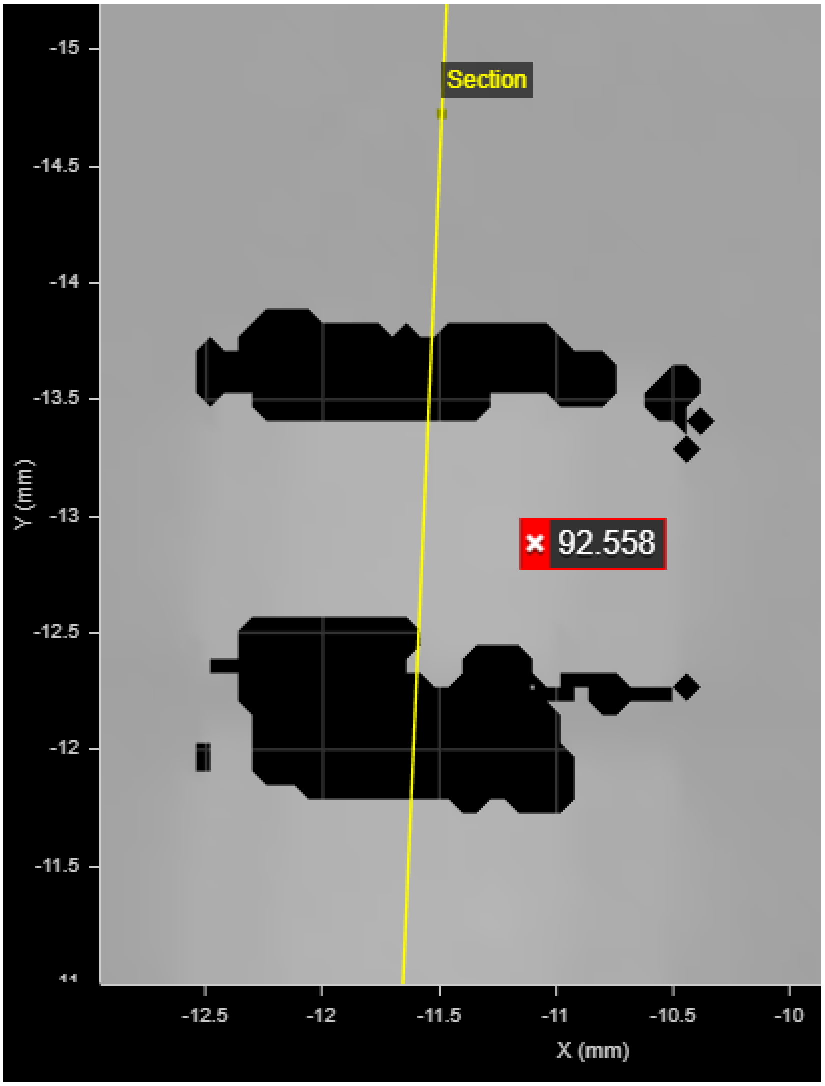
Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Section	Contains the coordinates of the two points that define the section.  **Point** The point to configure (1 or 2). **X, Y, Z** The coordinates of the point selected in **Point**.

Parameter	Description
Averaging Width	The width, in millimeters, of a window in which averaging of data points perpendicular to the section occurs. Use this to compensate for noise around the section. In the following, **Averaging Width** is set to 1. The red dots, representing the data points of the extracted profile, are the result of averaging the neighboring points along a line perpendicular to the section. When non-zero, this setting works in conjunction with the **Minimum Valid Points** setting (see below).  When set to 0, only data points directly under that section are used in the profile.

Parameter	Description
	

Parameter	Description
Minimum Valid Points	When **Averaging Width** is non-zero, the minimum percentage of neighboring points across the averaging width (perpendicular to the section) that need to be valid for a point to be output on the resulting profile. With the following Surface scan data (zoomed in and with the data viewer set to show individual data points), **Minimum Valid Points** has been set to 100%. As a result, no data points are output to the profile in the area that lacks valid data points (see profile to the right).   But with the following scan data, **Minimum Valid Points** has been set much lower, to 10%. As a result, the three or four data points to each side of the void are enough for an average to be calculated, and points are included in that area in the profile.

Parameter	Description
	 

Parameter	Description
Show Detail	Determines whether data points (in red) are displayed under the section in the data viewer. If this setting is disabled (as shown below), only the yellow line representing the defined section is displayed. 
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Segmentation

This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Segmentation tool separates surface data into "segments," based on the tool's parameters. Segments can be touching and overlapping to a certain degree. The Segmentation tool is especially useful in the food industry, for example to identify food items that are too small or too big, or items that are damaged.

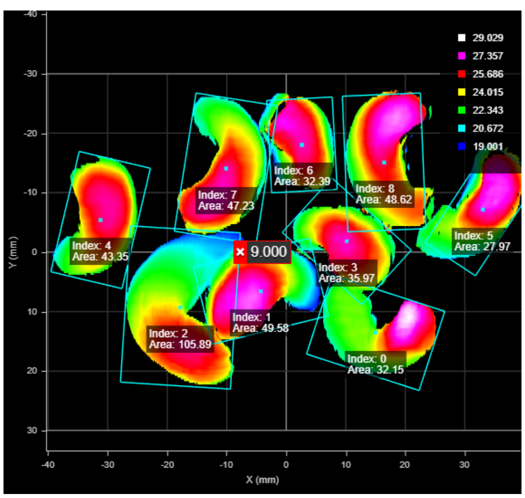
For each segment, the tool returns the X and Y position of its center, its length and width, and its area, as well as several more global measurements, such as maximum / minimum width or length, etc. For a complete list, see below.

The Segmentation tool can also be used as a second stage of processing after part detection. For example, part detection could be used to detect a tray (containing parts), and the Segmentation tool could then separate the parts within the tray. For information on part detection, see *Part Detection* on page 111. For a comparison of part detection, Surface Blob, and Surface Segmentation, see *Isolating Parts from Surface Data* on page 277.

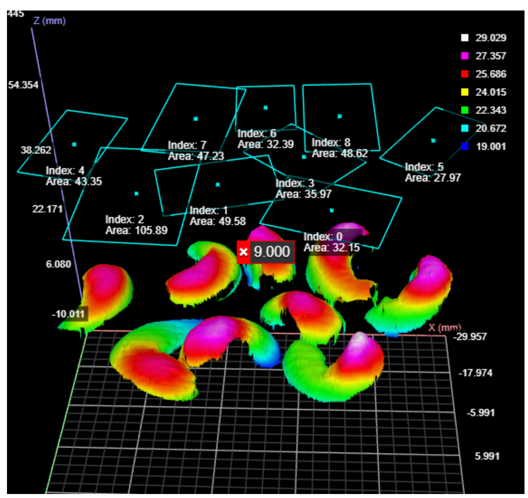
The Segmentation tool cannot handle large overlaps.

The Segmentation tool does not perform template matching.

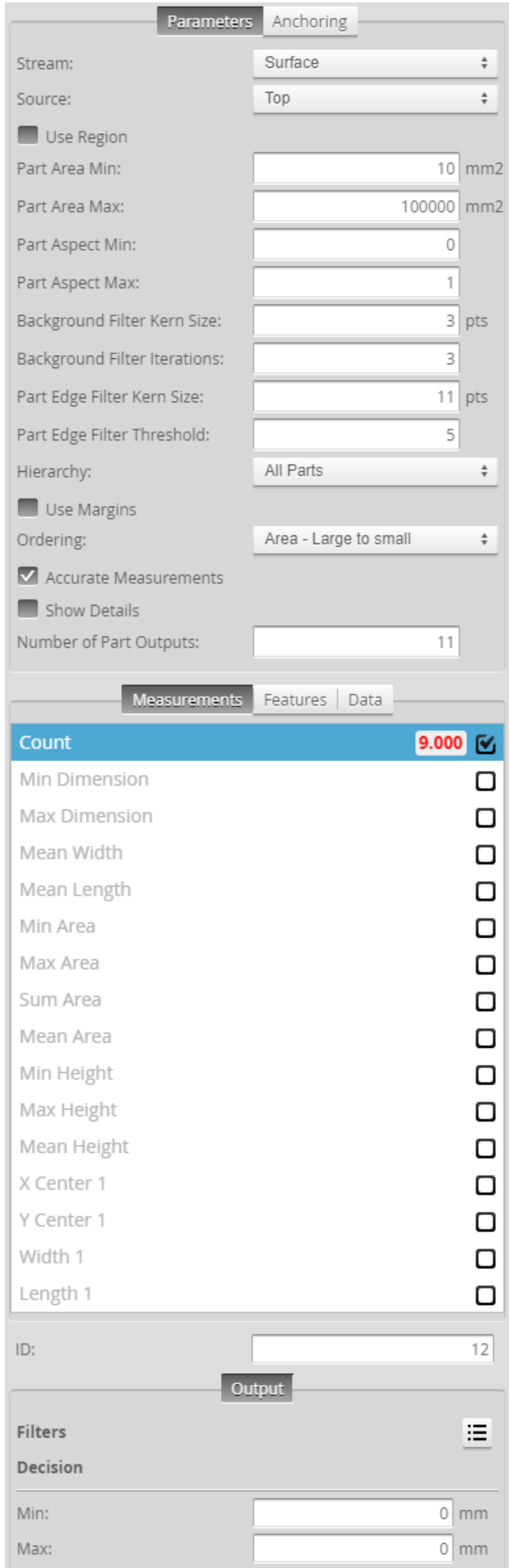
To reduce processing time, consider using the decimation filter. For more information on this filter, see *Filters* on page 108.



2D View



3D View



Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Data, and Settings

Measurements

Measurement
Count Returns the total number of segments identified, based on the tool's parameters.
Min Dimension **Max Dimension** The minimum and maximum dimensions among all of the identified segments.
Mean Width **Mean Length** The mean width and length of the segments, respectively.
Min Area **Max Area** The minimum and maximum area among all of the identified segments.
Sum Area The sum of the areas of the segments.
Mean Area The mean area of the segments.
Min Height **Max Height** The minimum and maximum heights among all of the identified segments.
Mean Height The mean height of the segments.
X Center {n} **Y Center {n}** The X and Y positions of the center of a part segmented from the surface. The **Number of Part Outputs** setting determines the number of measurements listed in the **Measurements** tab.
Length {n} **Width {n}** The length and width of a part segmented from the surface. These are always the major and minor axis of a part, respectively. The **Number of Part Outputs** setting determines the number of measurements listed in the **Measurements** tab.
Area {n} The area of a part segmented from the surface. The area is calculated using the contour of the part and resampling. For this reason, areas calculated using the Surface Volume tool will produce different measurements; for more information, see *Area* on page 481.

Features

Type	Description
Center Point {n}	The point representing the center of a segmented part. The **Number of Part Outputs** setting determines the number of point geometric features listed in the **Features** tab.

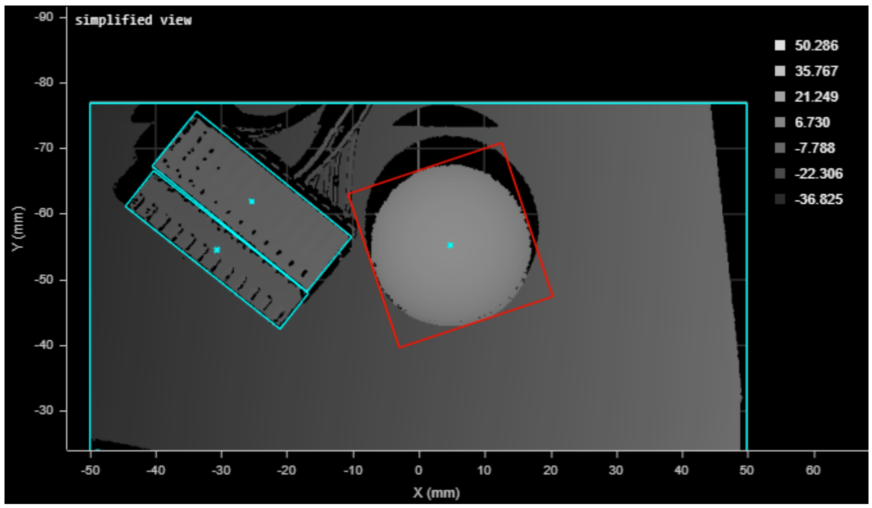
Data

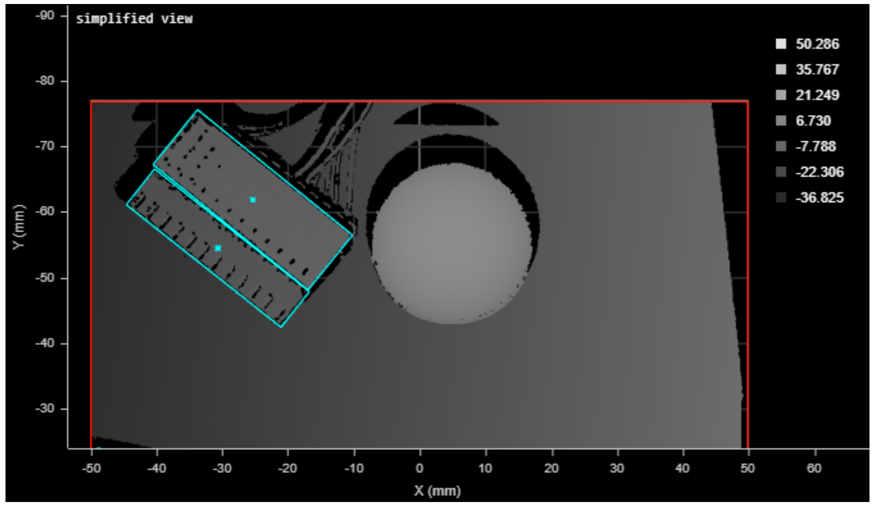
Type	Description
Segments Array	An array containing the segments. For an example of how to access this data from an SDK application or a GDK tool, see the appropriate sample in the SDK samples; for more information, see *Setup and Locations* on page 837.
Diagnostics Surface	Surface data you can use to evaluate the impact of the tool's kern size and iteration settings, which the tool uses to separate potential segments.
Surface {n}	Surface data corresponding to each segmented part.

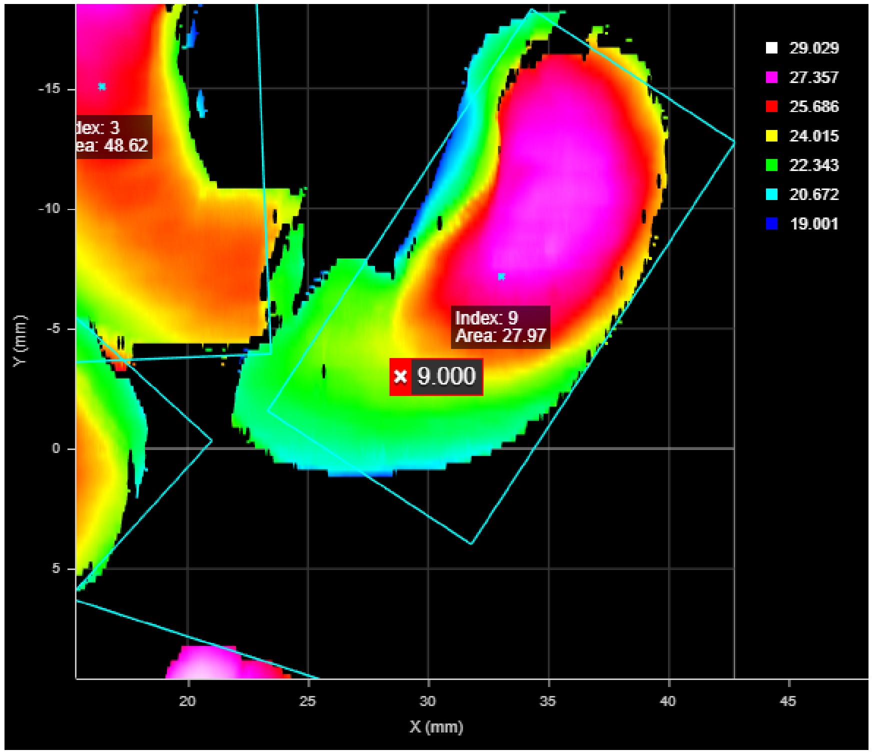
Parameters

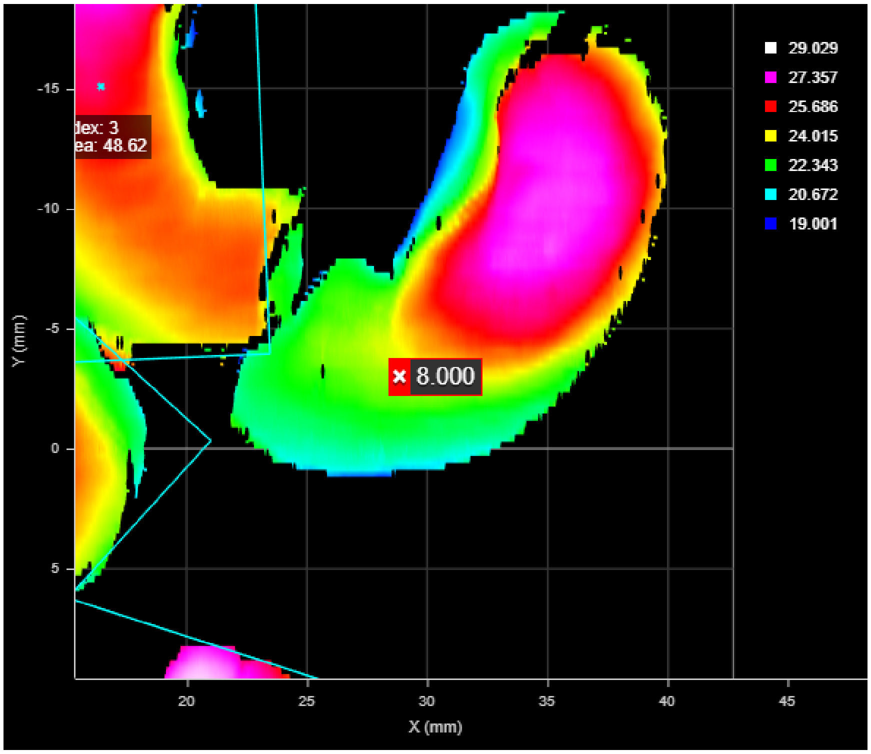
Parameter	Description
Source	The sensor that provides data for the tool's measurements.
Use Intensity	Causes the tool to use intensity. The option is only displayed if intensity data is available.
Use Region	Only displayed on older instances of this tool. Newer instances use "flexible regions" (see the parameters below in this table). Indicates whether the tool uses a user-defined region. If this option is not checked, the tool uses data from the entire active area.

Parameter	Description
Number of Regions Mask Type {n} / Region Type {n} Inner Circle Diameter Inner Ellipse Major Axis Inner Ellipse Minor Axis Sector Start Angle Sector Angle Range Mask Source Low Threshold High Threshold	Only displayed on newer instances of this tool. When you enable **Use Region**, the tool displays additional settings related to the measure region type. For details on flexible regions and their settings, see *Flexible Regions* on page 153. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 152.
Part Area Min Part Area Max	The minimum and maximum areas in square millimeters for a part of the scan data to be identified as a segment.
Part Aspect Min Part Aspect Max	The minimum and maximum aspect ratios (minimum axis length in mm) / (maximum axis length in mm) of the best fit ellipse to the segment contour points for a segment to qualify to be added to the list of found segments.
Background Filter Kern Size Background Filter Iterations	These settings perform background separation. The greater each of these values is, the more separation will be achieved. You must find a balance that removes noise adequately without degrading the segment find quality.
Part Edge Filter Kern Size	Use this value to adjust the “granularity” of the part edge detection.
Part Edge Filter Threshold	Controls the separation of the parts, increasing the gap between the parts so that they can be detected more easily.

Parameter	Description
Hierarchy	Use this setting to detect segments when they are surrounded by background data. Choose one of the following: **All Parts** or **External Parts**. **All Parts** This option lets you segment parts with surrounding background data. This is the default behavior in firmware 6.0 and later. Jobs created using firmware 5.3 SR1 or earlier default to **External Parts** (see below). For example, in the following image, with **All Parts** selected, the sphere is correctly segmented from the surrounding background.  Note that this option may result in "over-segmentation": the tool may segment a part into two segments. **External Parts** In the following image, with **External Parts**, the sphere is not identified as a segment because of the surrounding background. It is treated as part of a large segment that includes all of the background. (This "segment" is indicated by a red border that shows it's currently selected. Note that to exclude this kind of segment, you can set a maximum acceptable part area in the tool.)

Parameter	Description
	

Parameter	Description
Use Margins	When enabled, discards parts that are too close to the edge of the scanning area or the region, based on the left, right, top, and bottom values. The tool filters the parts using the center point. In the following, a part's center point is close to the edge of the XY scan area; the right margin is set to 0, so the part is not discarded. (Total part count is 9.)  In the following, the right margin has been set to 10 mm. Because the center point of the part is now within the margin, the tool discards the part. (Total part count is reduced to 8.)

Parameter	Description
	
Ordering	Orders the measurements, features, and surface data of the individual parts output by the tool. Choose one of the following: • Area - Large to small • Area - Small to large • Position - X increasing • Position - X decreasing • Position - Y increasing • Position - Y decreasing • Position - Z increasing • Position - Z decreasing
Show Details	Toggles whether the tool displays the index and area of each individual part.
Number of Part Outputs	Determines the number of parts the tool outputs as measurements, features (center points of parts), and surface data. Currently limited to 200 parts.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

Anchor	Description
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

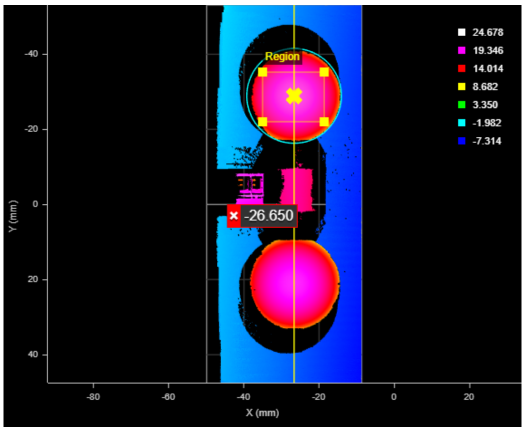
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

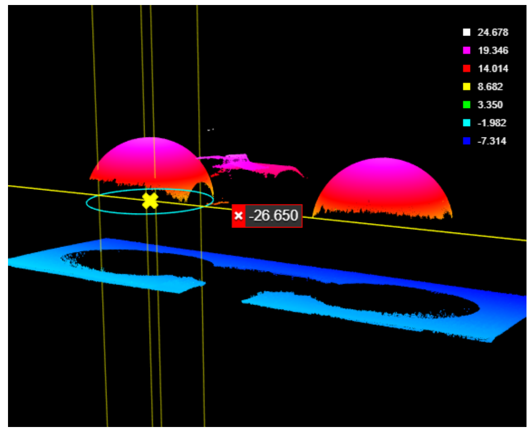
Sphere

The Sphere tool lets you compute characteristics of a scanned sphere by specifying a region to inspect. For example, you can use the tool to align a robot-mounted sensor to a ball-bar as shown in the images below.

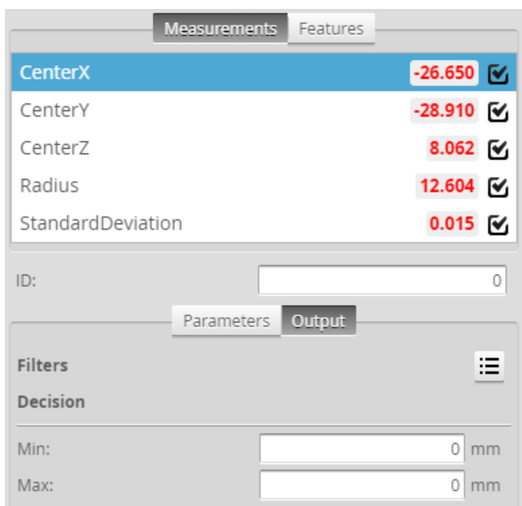
> For the tool to work properly, the tool's region typically must be enabled and set, and properly placed. For more information, see the table of parameters below.



2D View



3D View

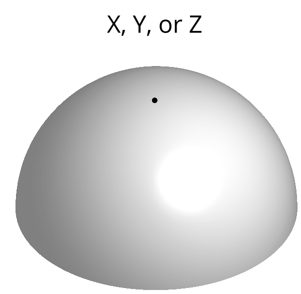
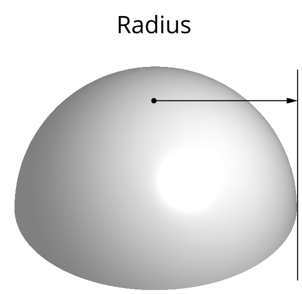


Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
Center X Determines the X position of the center of the sphere. **Center Y** Determines the Y position of the center of the sphere. **Center Z** Determines the Z position of the center of the sphere.	
Radius Determines the radius of the sphere.	
StandardDeviation Determines the error of the points compared to the computed sphere. It is defined as the square root of the variance of the distance of every point to the computed sphere.	

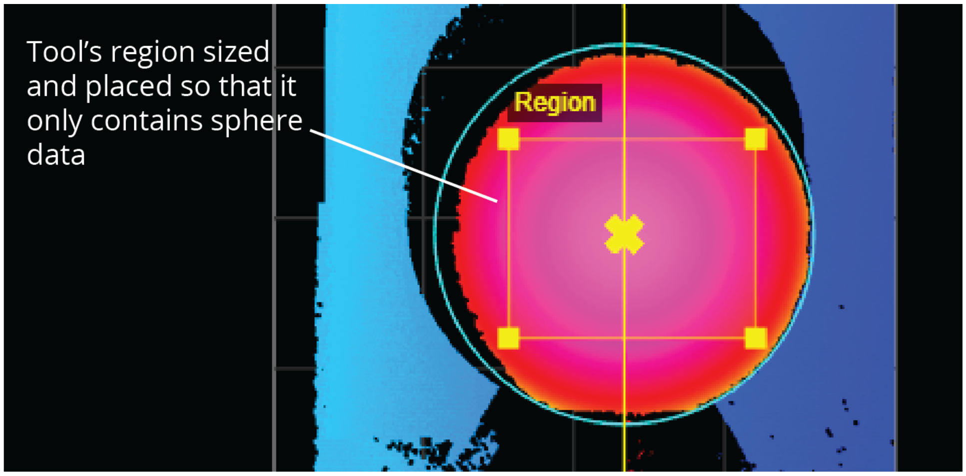
Features

Type	Description
Center	The center of the circle encompassing the widest part of the fitted sphere.
Circle	The circle encompassing the widest part of the fitted sphere.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.

Parameter	Description
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152. In order for the tool to correctly fit a sphere to the scan data, you must set the region so that it only contains data from the sphere on the target. 
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Stitch

> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Stitch tool lets you combine up to 24 frames of scans into a single Surface scan. This lets you get a much larger scan volume with fewer sensors (either in a single sensor system or a multi-sensor system). For each scan, you can specify not only X, Y, and Z offsets (translations), but also X, Y, and Z angles (rotations), defining its relationship with the others. This means that when the sensor system is mounted to a robot, or if you are using, for example, an X-Y table, you can get a complete scan with fewer sensors. The resulting combined scan can then be used as input by any other Surface or Feature tool from its **Stream** drop-down.

The tool performs rotation first, and then translation.

You cannot define sections on the combined scan; for more information on sections, see *Sections* on page 140.

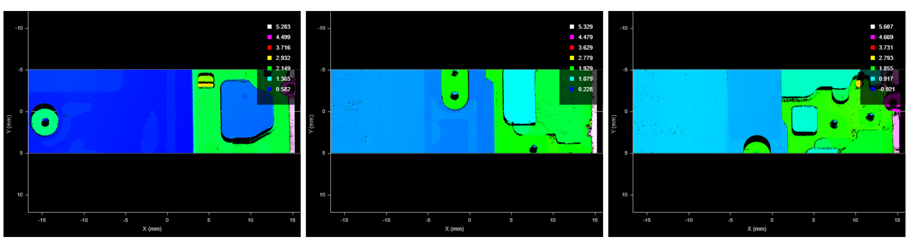
> The tool combines data simply by overwriting in sequence: it performs no averaging or blending. The tool also performs no fitting.

> Results are only as accurate as the motion system.

> Seams are often seen in combined data in stitching performed in anything other than along the Y axis.

The tool returns one measurement, which simply indicates the number of scans successfully added to the combined scan data.

The following shows three individual frames:



In the following, the tool has combined the frames into a single surface.

Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Data, and Settings

Measurements

Measurement
Captured Indicates the number of scans successfully added to the combined surface scan.

🗋	Only one of the following data types will contain data, depending on whether **Uniform Spacing** is enabled. For more information, see *Scan Modes* on page 88.

Data

Type	Description
Stitched Surface	The stitched surface scan, available for use as input in the **Stream** drop-down in other tools. Contains uniform data only and is empty if **Uniform Spacing** is disabled.
Stitched Raw Surface	The stitched surface scan, available for use as input in the **Stream** drop-down in other tools. Contains point cloud data only and is empty if **Uniform Spacing** is enabled.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Surface Count	The number of scans to combine into a single surface. For each, a "Surface Parameters" section is added. The tool accepts setting the number of scans to one: in this case it, behaves like a transform tool.
Enforce Frame Order	Restricts the stitching for specific frame indexes, starting at the frame indicated in **Start Frame Index**. If unchecked, an **Operation** drop-down is displayed (see below). This setting is disabled if you attempt to stitch data from individual scans acquired using the Snapshot button (that is, all frame indexes are at 1).
Operation	If **Enforce Frame Order** is disabled, the **Operation** drop-down is displayed. One of the following: • **Normal**: The tool automatically chooses this operation after you have chosen another operation. • **Reset buffers**: Resets the buffers used to stitch frames. • **Lock**: Lets you lock the current processing and outputs of the tool. Useful when you need to add another tool that will use this tool's output (for example, a Surface Section tool). If you do not lock the tool, as soon as you add the other tool, the output is cleared, which means you must re-execute the combined output again to configure the additional tool. Be sure to unlock the tool after you have configured any other tools.

Parameter	Description
Reset On Start	Clears buffers for the stitched surface when the sensor is started. Useful for situations where the sensor is started and stopped frequently (to capture a small number of frames), rather than starting the sensor and letting it run for a long period. Enable this parameter to prevent data from a previous capture session being stitched with data from the current capture session.
Bilinear Interpolation	Evaluates the height of each transformed point (through translation or rotation) based on its neighbors. More precise, but has an impact on performance.
Surface Parameters {n}	For each scan to be added to the combined surface scan, a **Surface Parameters** checkbox is added. To configure the parameters of the individual surfaces, check the box and configure the settings. Unchecking the checkbox does not disable the scan or its settings. The following settings are available: • Data Source • X, Y, and Z Offset • X, Y, and Z Angle
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

> A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

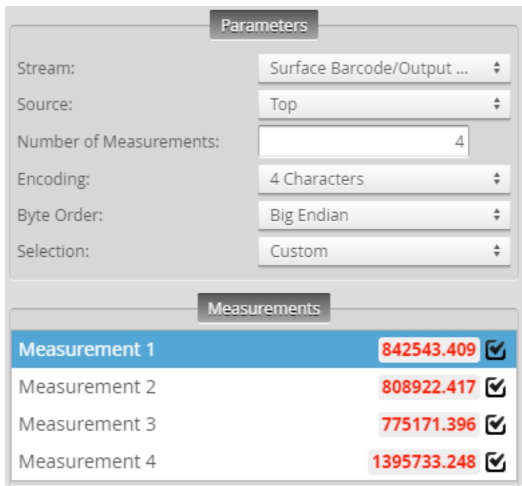
> For more information on anchoring, see *Measurement Anchoring* on page 169.

String Encoding

> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

> The String Encoding tool is only available from the drop-down in the **Tools** panel *after* a tool capable of providing compatible input, such as Surface Barcode or Surface OCR, has been added.

The String Encoding tool takes the string output from a Surface Barcode or Surface OCR tool and converts the characters to measurements that can be sent to PLCs. Measurements contain either a single value for each character, or a four-character string. You can set the endianness of the four-character string, letting you use the tool with any PLC.



For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements and Settings

Measurements

Measurement
Measurement {n} A decimal representation of either a single character or a four-character string, depending on the value of the **Encoding** parameter. In both cases, values are displayed with a decimal point, and three places after the decimal point. The number of measurements is set by the **Number of Measurements** parameter. The last character is always a null terminator (\0). If the string passed to the tool is longer than the number of measurements will accommodate, the last character is truncated and replaced with \0.

Parameters

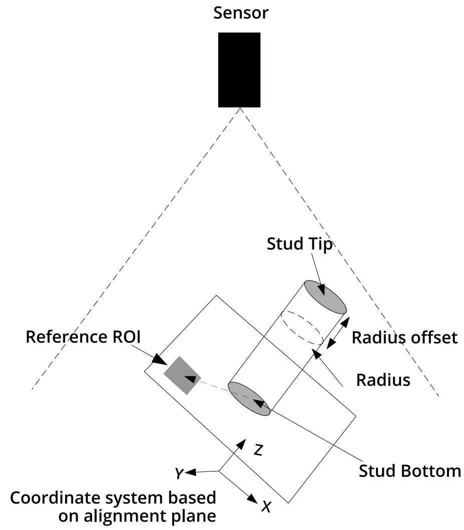
Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.

Parameter	Description
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool. If you switch from one type of data to another (for example, from section profile data to surface data), currently set input features will become invalid, and you will need to choose features of the correct data type.
Number of Measurements	The number of measurements the tool adds.
Encoding	One of the following: **4 characters**: Each measurement contains a four-character string, encoded using the byte order chosen in the **Byte Order** parameter. **1 character**: Each measurement contains a single character.
Byte Order	Selects the byte order the tool uses to encode strings in the measurements. One of the following: Big Endian or Little Endian.
Selection	Measurement selection functions. One of the following: **Enable All**: Enables all of the measurements. **Disable All**: Disables all of the measurements. (This parameter defaults to “Custom” before and after performing a selection.)
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Stud

The Stud tool measures the location and radius of a stud.

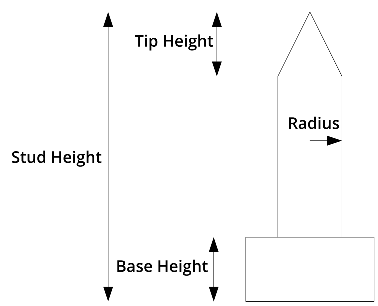
> The tool does not search for or detect the feature. The tool expects that the feature, conforming reasonably well to the defined parameters, is present and that it is on a sufficiently uniform background.

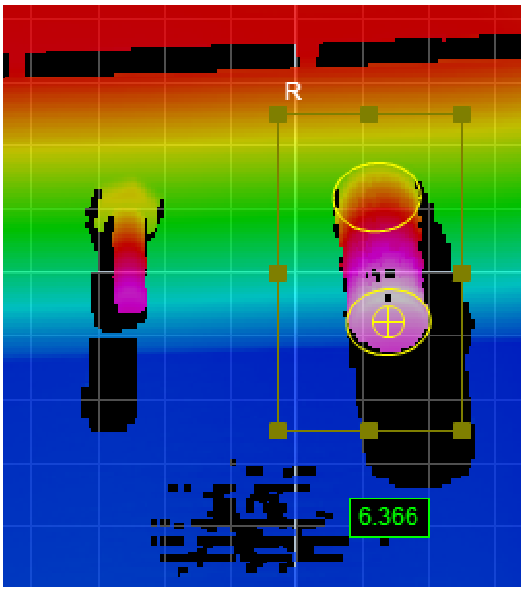


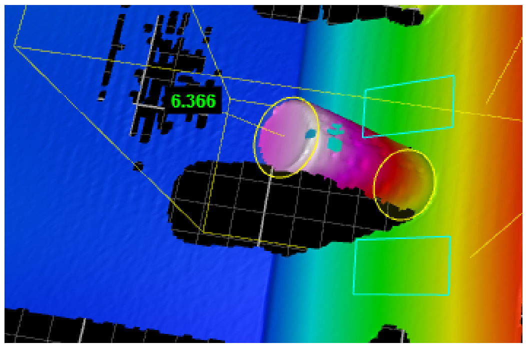
The tool uses a complex feature-locating algorithm to find a hold and then return measurements. See "Stud Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel.

The location of the stud is defined at either the stud tip or the stud base. The tip is the intersection of the stud axis and the top of the stud; the base is the intersection of the stud axis and the surrounding plane.

The stud shape is defined by the tip height and base height. The base and tip heights specify where the shaft with the nominal radius begins and ends.

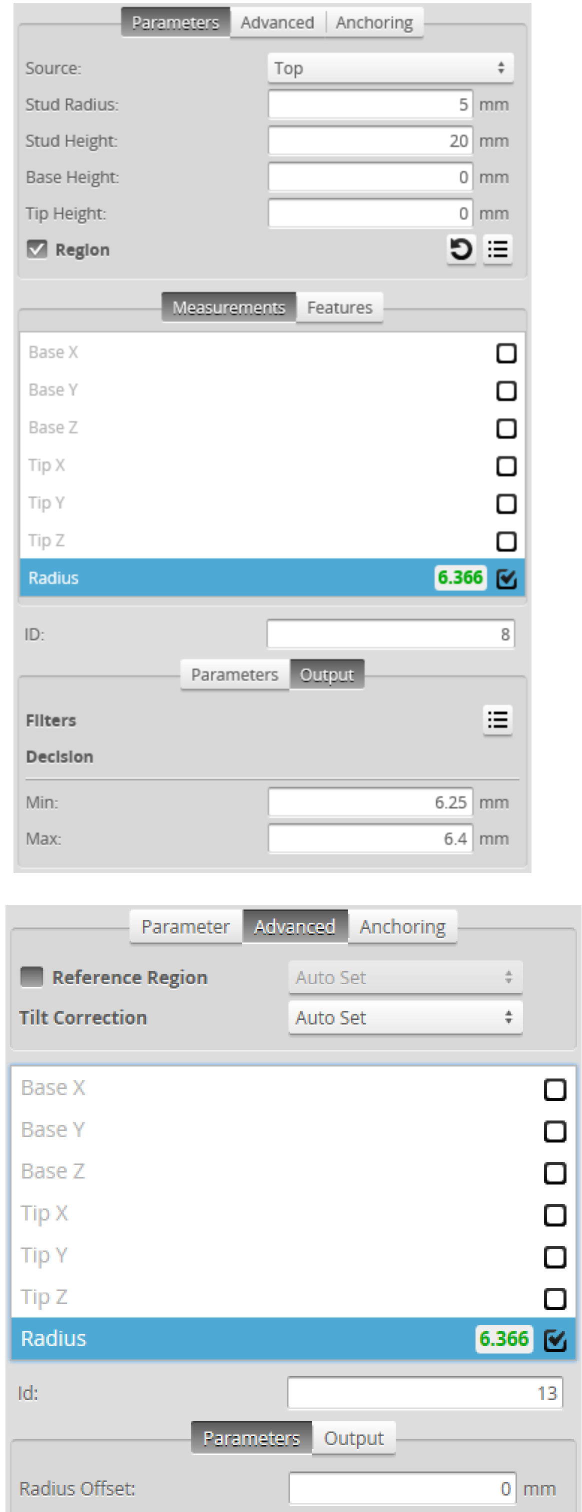






3D View

2D View

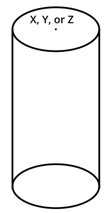
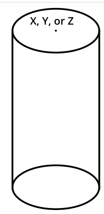
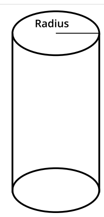


Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
Tip X Determines the X position of the stud tip.	
Tip Y Determines the Y position of the stud tip.	
Tip Z Determines the Z position of the stud tip.	
Base X Determines the X position of the stud base.	
Base Y Determines the Y position of the stud base.	
Base Z Determines the Z position of the stud base.	
Radius Determines the radius of the stud.	

Features

Type	Description
Tip Point	The center point of the tip of the stud.
Base Point	The center point of the base of the stud.

For more information on geometric features, see *Geometric Features* on page 164.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Stud Radius	Expected radius of the stud.
Stud Height	Expected height/length of the stud.

Parameter	Description
Base Height	The height above the base surface that will be ignored when the (truncated) cone is fit to the stud data.
Tip Height	The height from the top of the surface that will be ignored when the (truncated) cone is fit to the stud data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Reference Regions	The tool uses the reference regions to calculate the base plane of the stud. Reference regions are relative to the base of the stud.
Tilt Correction	Tilt of the target with respect to the alignment plane. **Autoset**: The tool automatically detects the tilt. The measurement region to cover more areas on the surface plane than other planes. **Custom**: You must enter the X and Y angles manually in the X Angle and Y Angle parameters (see below).
X Angle Y Angle	The X and Y angles you must specify when **Tilt Correction** is set to **Custom**. You can use the Surface Plane tool's X Angle and Y Angle measurements to get the angle of the surrounding surface, and then copy those measurement's values to the **X Angle** and **Y Angle** parameters of this tool. For more information, see Plane.
Radius Offset *(Radius measurement only)*	The distance from the tip of the stud from which the radius is measured.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

Measurement Region

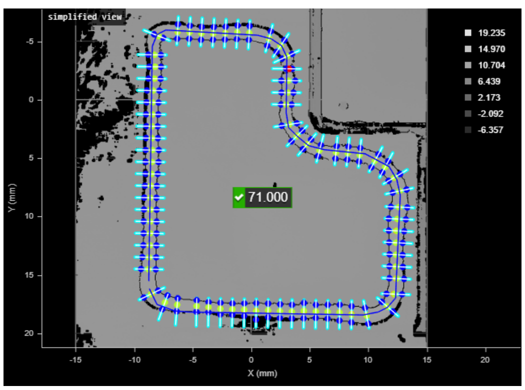
The tip and the side of the stud must be within the measurement region.

Track

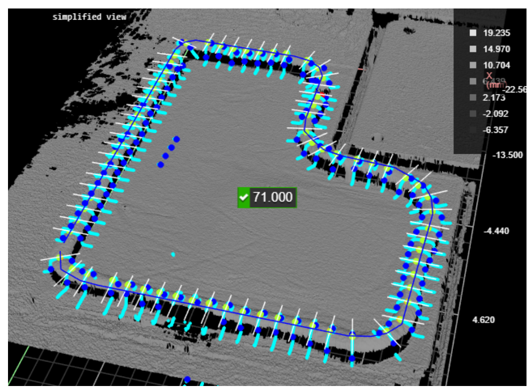
> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated by a PC. (The tool is not supported on GoMax.) For more information on PC-based acceleration, see *Gocator Acceleration* on page 529. The tool is supported in emulator scenarios.

The Track tool lets you perform quality control and inspection along a path you define on representative scan data. The Track tool is especially useful for inspecting materials such as glue / sealant beads. The tool returns width and height measurements of the material, as well as OK and NG ("no good") counts, which let you monitor material overflow and breaks. A major advantage of the tool is that it removes the need to configure individual tools for each location along the path. You can use point and line geometric features to anchor the tool (for more information on geometric features, see *Geometric Features* on page 164).

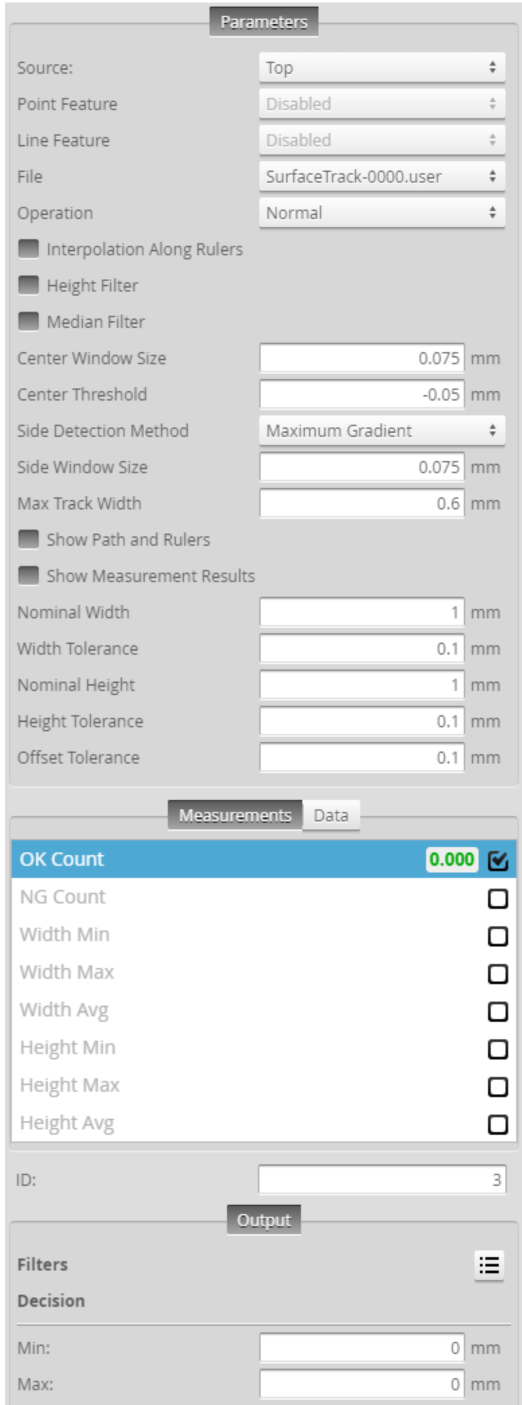
> Gocator sensors have a limited amount of space for storing path files. For this reason, when working with large datasets, we recommend that you run the Track tool on a PC through the Gocator accelerator. For more information on the accelerator, see *Gocator Acceleration* on page 529.



2D View

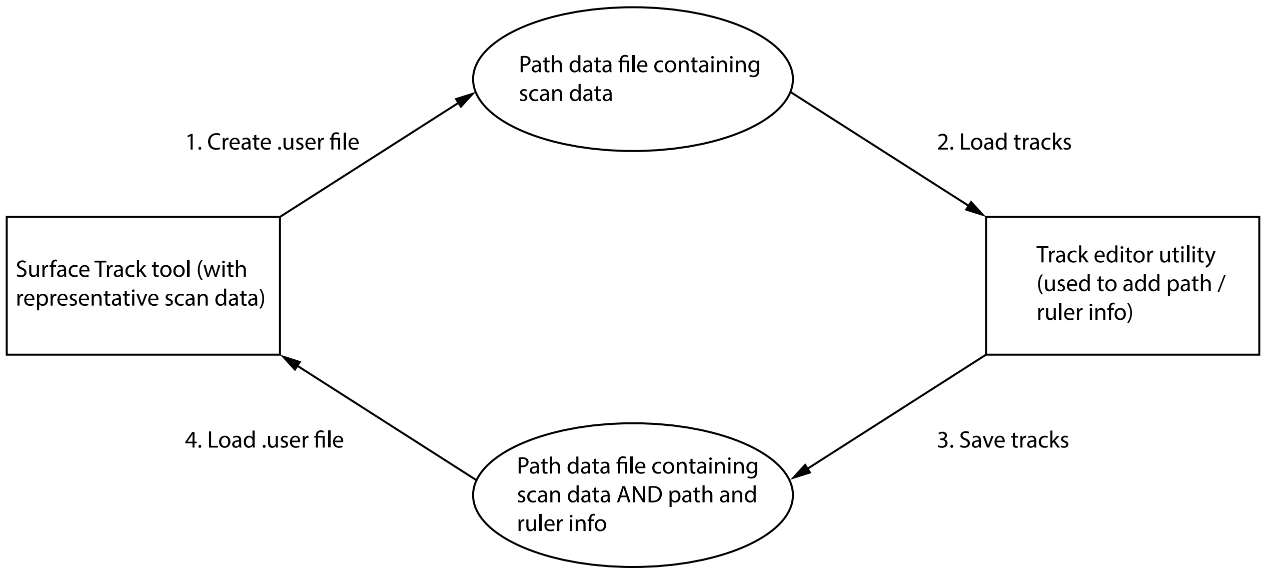


3D View



Measurement Panel

You define the path along which the tool performs its internal measurements using a separate, PC-based utility (the "track editor"). The following shows the relationship between the Track tool and the track editor.



For more information on the track editor, see *Using the Track Editor* on page 462

	All instances of the Track tool share the same path file set in **File** (ending in .user). For this reason, you must be careful when editing or removing path files shared by another instance of the tool.

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Key Concepts

The following are important concepts for using both the track editor (see *Using the Track Editor* on page 462) and the Track tool itself:

Track: The material being measured, for example glue or sealant. The material can sit on a flat area on the target, or sit in a groove where the material touches one or both sides.

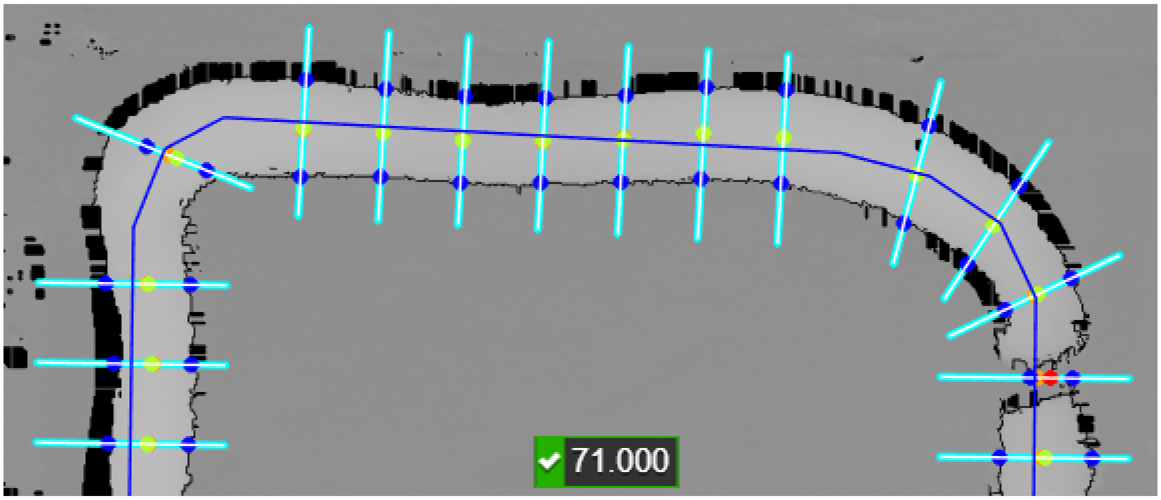
Path: The ideal centerline of the track. You define the path in the track editor. You can define more than one path for use on scanned targets, but the Track tool returns the combined results for all paths. For more information on the track editor, see *Using the Track Editor* on page 462.

Ruler: A ruler is one of the areas perpendicular to the path you define. You define the size and spacing of the rulers in the track editor. The Track tool extracts a profile from the surface data beneath a ruler and performs internal measurements based on the values you choose in the Track tool's parameters.

Ruler profiles: The profiles extracted from the surface data under a ruler. The tool's internal measurements, which are configured using the tool's settings, are applied to these profiles.

Segment: One portion of the path, between points created by clicking on an image of scan data in the track editor. You can choose to configure rulers in segments independently, or choose to configure them in a batch mode.

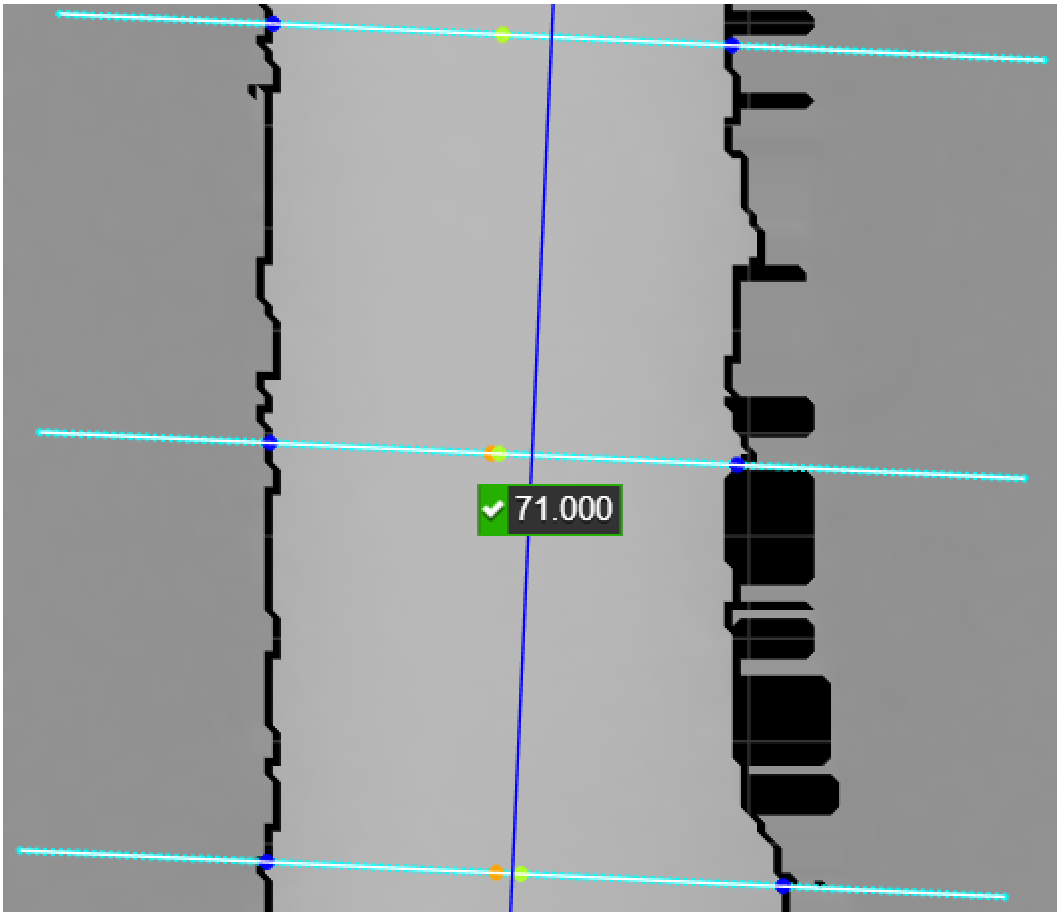
The following shows a track with rulers and measurement results:



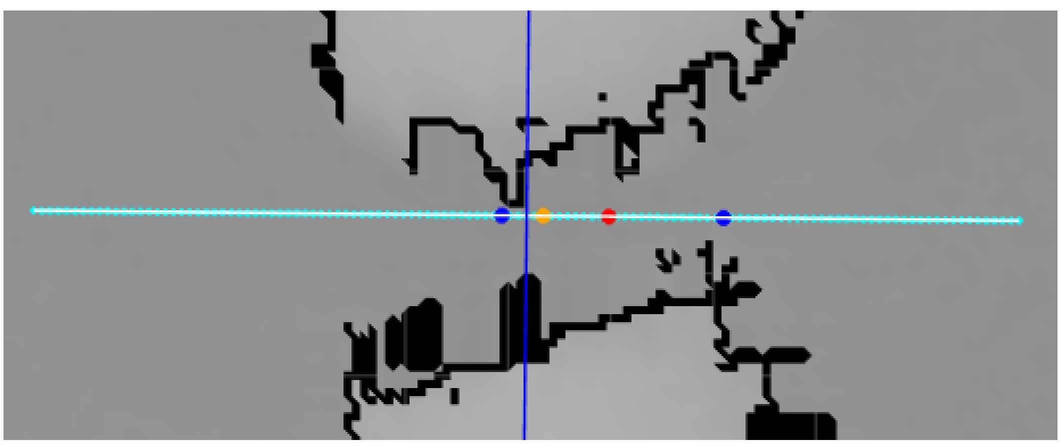
Track tool in data viewer, showing a track (lighter grey), path (dark blue line), rulers running perpendicular to the track (white lines centered on light blue dots). Dots of other colors provide additional information (see below).

When you enable **Show Measurement Results**, the Track tool displays dots on the rulers to provide the following information (see also the images below):

- Light blue dots: The data points in the ruler profile. When you enable **Show Path and Rulers**, the tool displays a white line centered on these dots to indicate the location of the ruler.
- Dark blue dotes: The detected sides of the track. These represent the width of the track under that ruler.
- Green dots: Center points on rulers that pass the criteria set in the tool. These count toward the "OK Count" measurement.
- Red dots: Center points on rulers that fail at least one of the criteria set in the tool. These count toward the "NG Count" measurement.
- Orange dots: The peak (highest) point on the ruler. If the center point (green or red) is the same as the peak point, the tool only shows the center point.



Three "OK" rulers, indicated by green center points. In the bottom two, the peak point (orange) is slightly to the left of the center point (green).



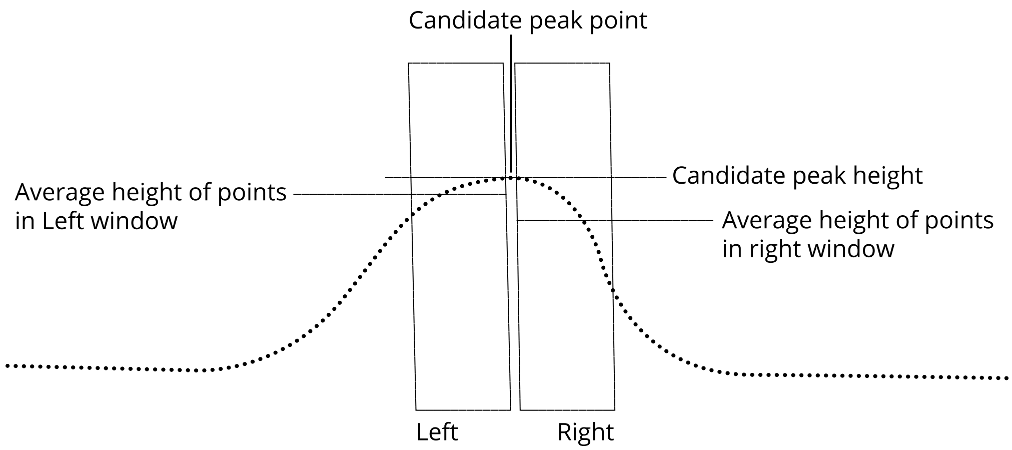
A "NG" ruler, indicated by the red center point.

Track Location

The tool attempts to locate the track using the profile data it extracts under each ruler, and does this by first locating the "peak" (the highest point on the ruler profile, based on certain criteria) and then locating the side points representing the "sides" of the track.

Peak Detection

The tool determines the peak point on a ruler profile by moving two windows—one to each side of the point being examined—and comparing the average height in those windows with the height of the point being examined. (The size of these windows is specified in **Center Window Size**.) If the height of the point being examined is greater than both the left and right average height by the value specified in **Center Threshold**, that point is considered a candidate peak point. The tool uses the candidate point with the *highest* average height over both windows as the peak point.



Side Detection

After the tool has located the peak point, it locates the sides of the track starting from the peak point. You can choose between two methods for side detection: Maximum Gradient and Height Threshold.

Maximum Gradient:

Use this side detection method when the slope of the two sides show a clear drop-off. The following settings define the area in which the tool searches for a maximum gradient, which will determine the edge of the track.

Maximum Gradient Side Detection Parameters

Side Window Size	The size of the two adjacent windows the tool uses to determine the maximum slope on the left and right side of the track. Set this to roughly 3 to 5 times the smaller of the X and Y resolution of the sensor.
Max Track Width	The maximum width of the track over the ruler profile the tool searches for edge points. The tool uses this value to limit where the edge of the track might be detected. Set this to slightly larger than **Side Window Size**.

Height Threshold:

Use this side detection method when the slope of two sides is very gradual. The tool finds the left and right edges by averaging the height of small fixed-size windows moving away from the peak point. Edge points are the left-most and right-most window locations where the average height is *below* a minimum height threshold.

Height Threshold Side Detection Parameters

Side Height Threshold	The minimum height that the average calculated in the fixed-width height threshold windows must be below.

Center Point Detection

The Track tool calculates the center point as the mid point between the left and right side points. This means that the center point may be different from the peak point.

Configuring the Track Tool

To configure the tool, you must first acquire scan data of a representative target; preferably, the material on the target will fall within the expected tolerances. Next, you save the scan data from within the Track tool, and then load the scan data into the track editor. Then, after adding a path or paths, and configuring rulers to the data, you load the track data back into the Track tool. Finally, you configure the tool. For more information on key concepts you need to understand to configure the Track tool, see *Key Concepts* on page 451.

To configure the Track tool:

1. Scan a representative target, or load previously scanned data.

   For more information on loading previously scanned data, see *Recording, Playback, and Measurement Simulation* on page 66.

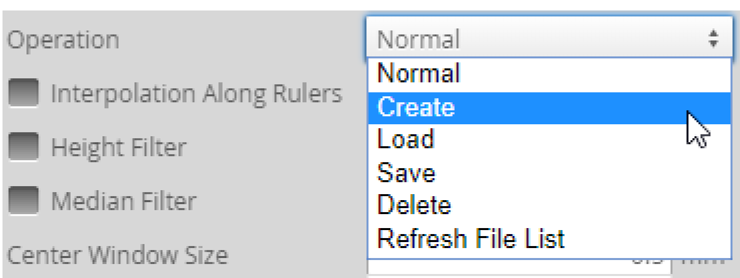
2. Add a Surface Track tool.

   Gocator adds a Surface Track tool and creates a "C:\GoTools\SurfaceTrack" folder if it doesn't exist. Note that previous versions of this tool created and placed files in a "C:\LMI" folder. Files are still read from both locations but only written to C:\GoTools. Rename the existing C:\LMI folder to C:\GoTools for seamless transition.

   For more information on adding a tool, see *Adding and Configuring a Measurement Tool* on page 149.

3. In the Surface Track tool, choose **Create** from the **Operation** drop-down.

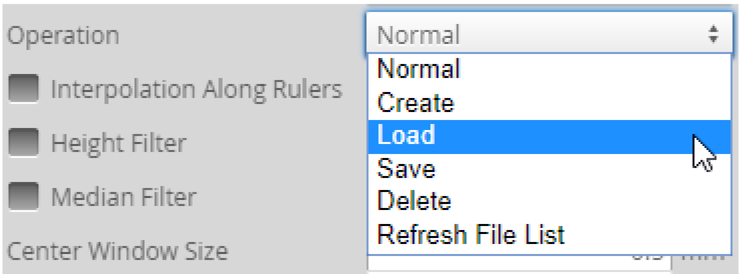
   The tool creates a file (for example, SurfaceTrack-0000.user) containing scan data in "C:/LMI/Surface Track". You will use the track editor to add path data to this file.

4. Launch the track editor and configure the path or paths.

   For information on using the track editor, see *Using the Track Editor* on page 462.

5. After you have finished editing the track data in the track editor, in the Surface Track tool, choose **Load** in the **Operation** drop-down to load the path data you just created.

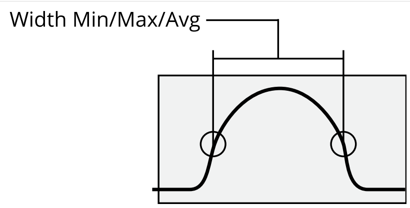


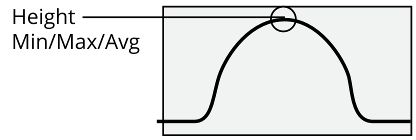
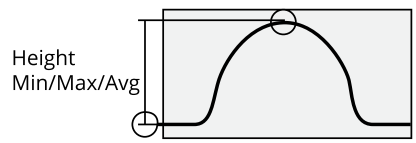
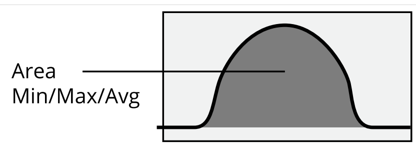
6. Configure the Track tool as required.

   For information on the tool's measurements and settings, see the below.

Measurements, Data, and Settings

Measurements

Measurement	Illustration
OK Count Returns the number of rulers along the path that pass all of the criteria set in the tool's parameters.	
NG Count Returns the number of rulers along the track path that fail the criteria set in the tool's parameters. (They are "no good.")	
Width Min **Width Max** **Width Avg** These measurements return the minimum, maximum, and average width of the track.	 *Width measurements on a ruler profile. The Track tool's settings determine the locations of the "sides" of the track.*

Measurement	Illustration
Height Min **Height Max** **Height Avg** These measurements return the minimum, maximum, and average height of the track at the center point. When **Height Mode** is set to **Absolute Height**, the height returned is the global height. When it is set to **Step Height**, the height is relative to the surface next to the track.	 *Height measurements on a ruler profile with Height Mode set to Absolute Height.*  *Height measurements on a ruler profile with Height Mode set to Step Height.*
Area Min **Area Max** **Area Avg** These measurements return the minimum, maximum, and average area under the rulers.	 *Area measurements under a ruler profile.*

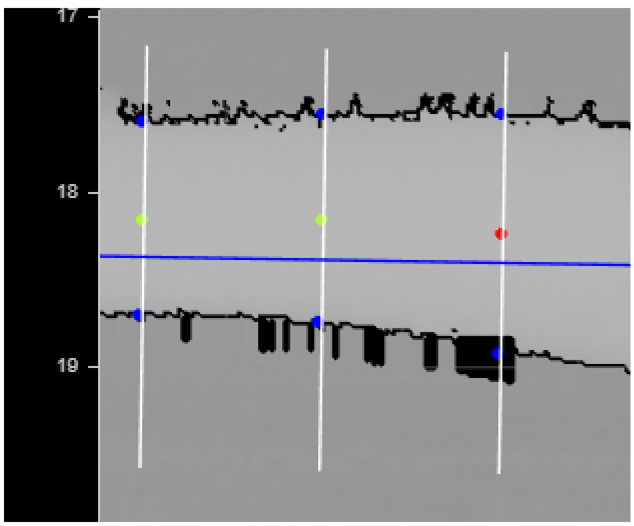
Data

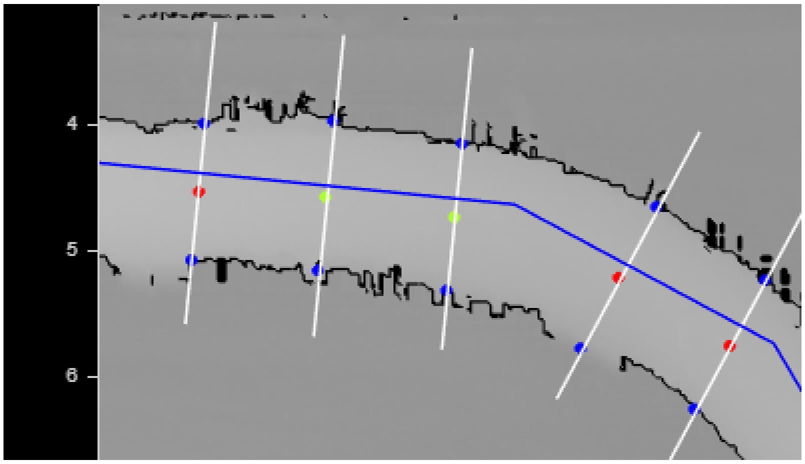
Type	Description
Output Measurement	Data containing the results from each ruler, namely: • track ID • segment ID • track width • track height • track offset • X position of the center point • Y position of the center point A sample included in the SDK package shows how you can use this output data in an application.
Profiles List	A list of the profiles extracted from the tracks.
Profiles List Diagnostics Surface	Surface data created by combining the extracted profiles. Use for diagnostics.

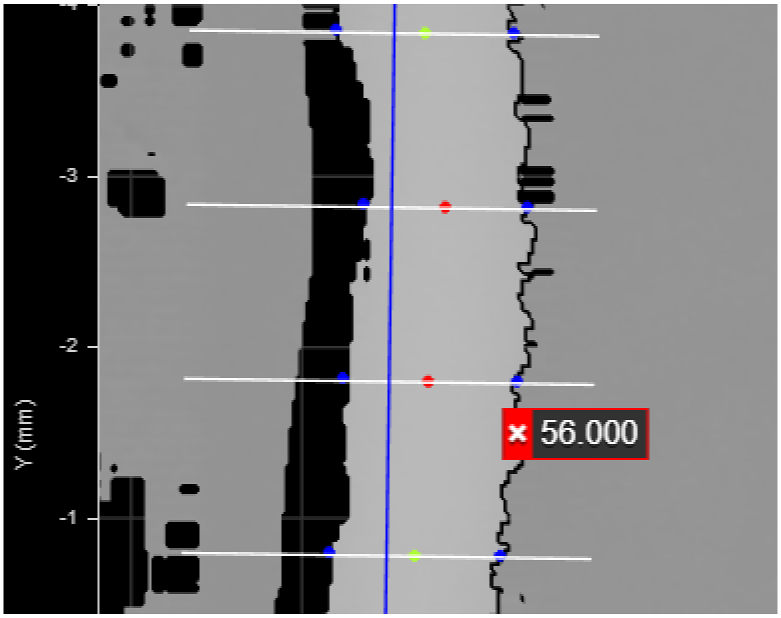
Main Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.

Parameter	Description
Point Feature Line Feature	Point and line geometric features (produced by another tool) that you can select as anchors for translation and rotation transformations, respectively. Currently, you must select both in order for anchoring to work. For more information on geometric features, see *Geometric Features* on page 164.
File	The CSV file that contains scan and path data. You add path data to the file using the track editor. For more information on the track editor, see *Using the Track Editor* on page 462.
Operation	Provides operations related to the CSV scan / path data file. One of the following: • **Normal**: Selected by the tool after you perform another file operation. • **Create**: Creates a new CSV file for use with the track editor. • **Load**: Loads the path file selected in **File**. • **Save**: Saves changes made in the scan data, as well as the geometric features used as anchors in the **Point Feature** and **Line Feature** settings, to the file selected in **File**. • **Delete**: Deletes the path file selected in **File**. • **Refresh File List**: Refreshes the list of files.
Interpolation	Enables linear interpolation on the profile extracted from the rulers to achieve sub-pixel accuracy in the width and height measurements.
Height Filter Threshold High Threshold Low	When **Height Filter** is enabled, use the **Threshold Low** and **Threshold High** settings to set a range to filter out noise or exclude other undesired data along the ruler profiles.
Median Filter Window Size	When **Median Filter** is enabled, specify the window the tool will use to smooth the height values of the points in the ruler profiles in the **Window Size** setting.
Center Window Size	The size of the left and right windows the tool moves along the ruler profile to detect whether the point centered between the two is the highest point along a ruler (the center point). Set this to roughly 50% of the typical width of the track as a starting point.
Center Threshold	The center point is determined by moving two side-by-side windows (left and right, **Center Window Size** setting) over each ruler profile. At each point, the height value between the two windows is compared to the average height of the left and right windows. If the center point height is greater, by the amount set in **Center Threshold**, than the average height in both the left and right windows, that point is considered a candidate center point. The candidate center point with the highest average height over both windows is used as the center point. It may be necessary to use a negative number in some cases. It may be necessary to use a negative value under some circumstances. For example, when the top point slightly dips below its surroundings.

Parameter	Description
Side Detection Method	The method the tool uses to detect the two sides of the track. One of the following: **Maximum Gradient** or **Height Threshold**. For more information on side detection method settings, see *Track* on page 449.
Height Mode	Determines how height values are interpreted in the tool's **Nominal Height** setting and what the returned height measurements represent. One of the following: **Absolute Height** - Height values are interpreted globally (the entire scan data). **Step Height** - Height values are relative to the surrounding area of the track.
Show Path and Rulers	Displays the path and rulers (as defined in the track editor) on the scan data.
Show Measurement Result	Shows dots on each ruler representing the results of the internal measurements on the profile extracted from the surface data under the ruler. For more information, see *Key Concepts* on page 451.
Nominal Width	The expected width of the track.
Width Tolerance	The tolerance applied to the nominal width. In the following, the distance between the blue dots indicating the width of the track under the ruler to the right (white vertical line) is greater than the width tolerance; this is indicated by the red center point dot, and counts as a NG measurement. The widths of the track under the two rulers to the left are within tolerance; this is indicated by green center points, and count as OK measurements. The track is lighter grey than the surrounding surface. 
Nominal Height	The expected height of the track. The expected height is the absolute height in the scan data, not relative to the surrounding area. This setting applies to the peak point, not the center point.

Parameter	Description
Height Tolerance	The tolerance applied to the nominal height. This setting applies to the peak point, not the center point. In the following, the red center points indicate that the height at that point is outside of the height tolerance. Green points indicate heights within tolerance. The track is lighter grey than the surrounding surface. 
Nominal Area	The expected cross-sectional area under the rulers on the track.
Area Tolerance	The tolerance applied to the nominal area.

Parameter	Description
Offset Tolerance	The maximum allowed distance between the center (highest) point on a ruler and the path. This setting applies to the center point. In the following, the top and bottom center points (green) are at an acceptable distance from the blue path. The red center points fail because they are too far from the path. The track is lighter grey than the surrounding surface. 
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchoring

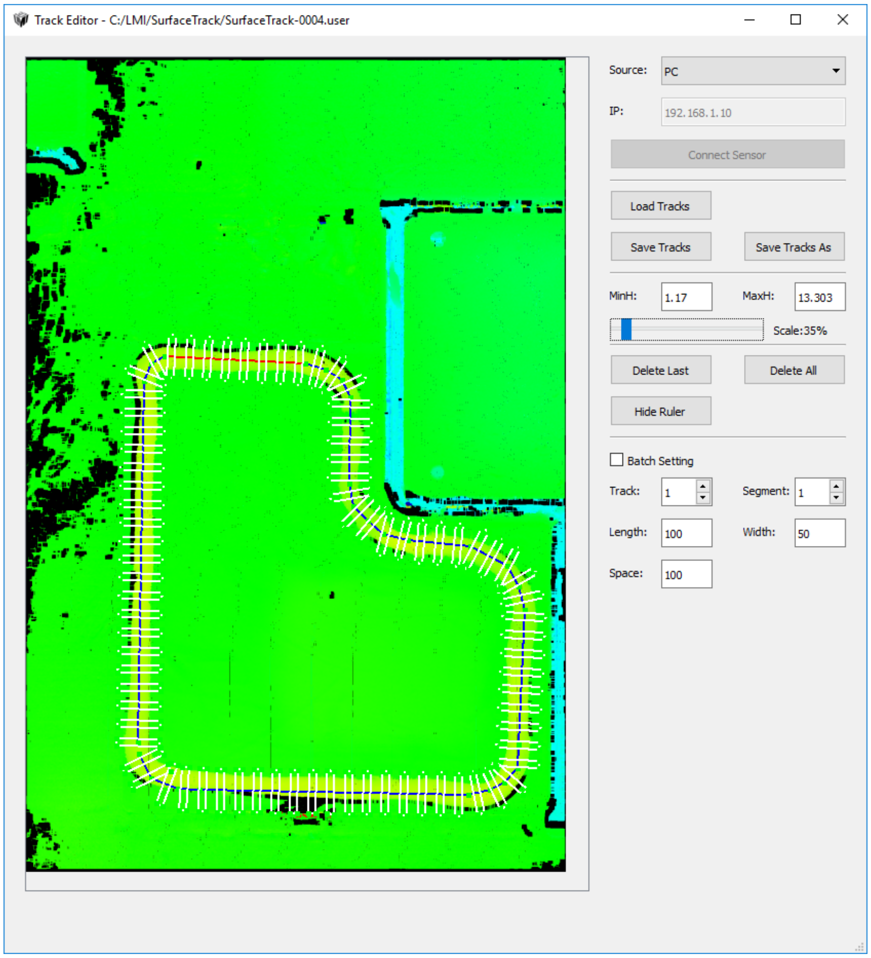
Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

For more information on anchoring, see *Measurement Anchoring* on page 169.

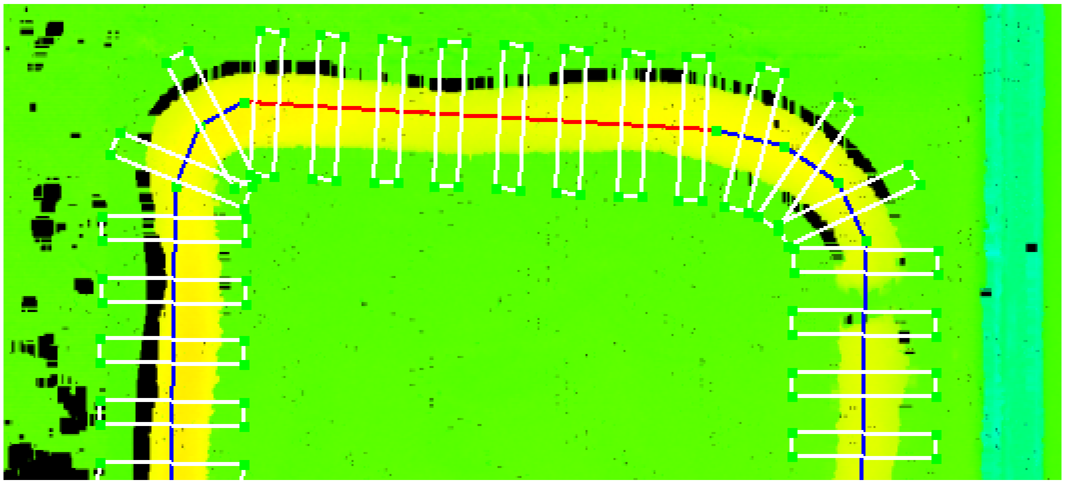
Using the Track Editor

You use the track editor to configure "path" and "ruler" information on a frame of scan data from a sensor. The Track tool uses this information to inspect targets along the defined path.



The track editor

In the track editor, you can define one or more paths, and configure rulers along these paths.



Closeup of the track editor window, showing a track of material on a surface (yellow on green), a path (blue segments; red segment for the currently selected segment), path points (green dots), and rulers (white rectangles).

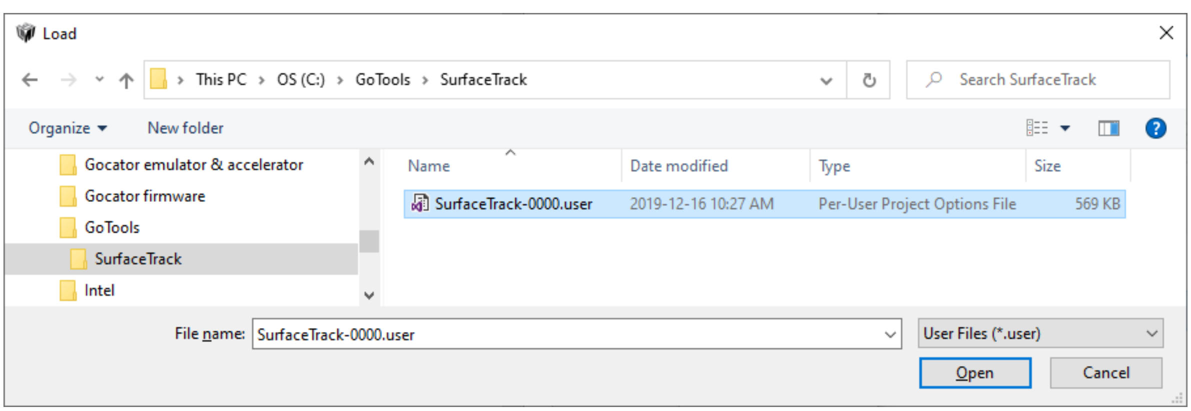
	The following assumes that you have already scanned a representative target and created a CSV file from within the Track tool. For more information, see the first steps of *To configure the Track tool:* on page 455.

Loading and working with scan/track data:

1. In the track editor, in the Source drop-down, choose one of the following:

- **PC**: Choose this option if you are using the Track tool through the accelerator. The track editor will retrieve the path data file from local (PC) storage and save changes there. (Choose the same if you are using the emulator).
- **Sensor**: Choose this option if you are not using the accelerator. The track editor will retrieve the path data file from the sensor at the IP address specified in the **IP** field. Because sensors have a limited amount of space to store path data, only use this option for simple paths.

2. Click **Load Tracks**, navigate to "C:\GoTools\SurfaceTrack" (if you have chosen PC as the source), and choose the .user file you created using the Surface Track tool.



The track editor loads the data. If paths have been previously defined, they are also loaded.

Note that previous versions of this tool created and placed files in a "C:\LMI" folder. Files are still read from both locations but only written to C:\GoTools. Rename the existing C:\LMI folder to C:\GoTools for seamless transition.

3. Do one or more of the following:

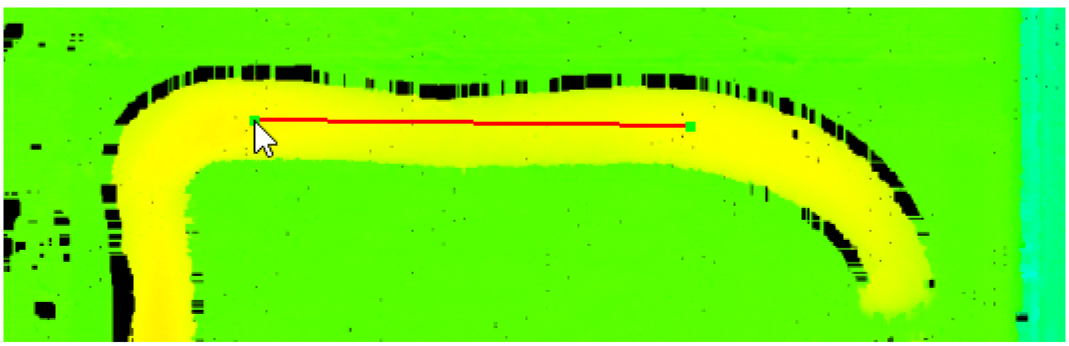
- Move the slider to the left or right to zoom in or out in the editor's viewer.
- Move the data in the track editor's window using the scrollbars or the mouse wheel.
- Set MinH and MaxH and then reload the track data to assign a narrower height range to the height map colors. This may help make the track clearer in the editor.

After you have loaded the data, you must add a path and configure its rulers.

To add a path:

1. In the track editor, click on the middle of the track somewhere in the scan data, move the mouse pointer to another location and click again.

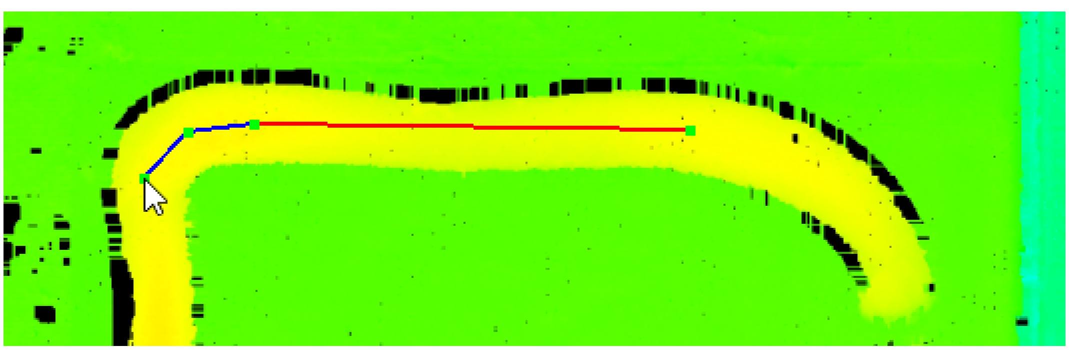
   A red segment between the first two green path points appears in the editor window.

   You can move path points using the mouse at any time to adjust the path. You can also delete the last point by clicking **Delete Last**. To delete all path points, click **Delete All**.

2. Continue clicking along the track to add more path points, building up the path.

   When adding points on corners, add more points to follow the track more precisely.

3. Continue clicking until you complete the path along the track.

   You cannot close the path: simply click close to the starting path point when you have finished.

4. Click **Save Tracks** to save the path information to the data.

5. (Optional) You can add other paths if necessary by clicking somewhere in the scan data *after* you have saved the track data.

After you have finished adding a path, you must configure the rulers on the path (the dimensions and the spacing of the rulers). You can choose to apply dimensions/spacing to *all* rulers in all segments at the same time by checking **Batch Setting**. The settings also apply to all paths if you have defined more than one path.

Otherwise, you must move through the individual path segments by clicking the spinner control in the **Segment** field and set the ruler dimensions for each segment. If you have defined multiple paths, you will have to click through the paths too, using the **Track** spinner.

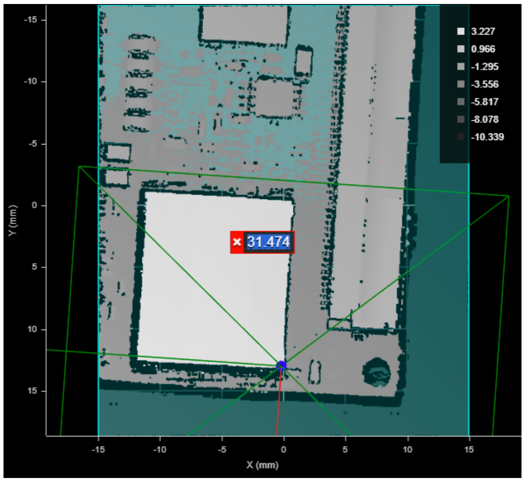
The following table lists the ruler settings available in the track editor:

Track editor: ruler settings

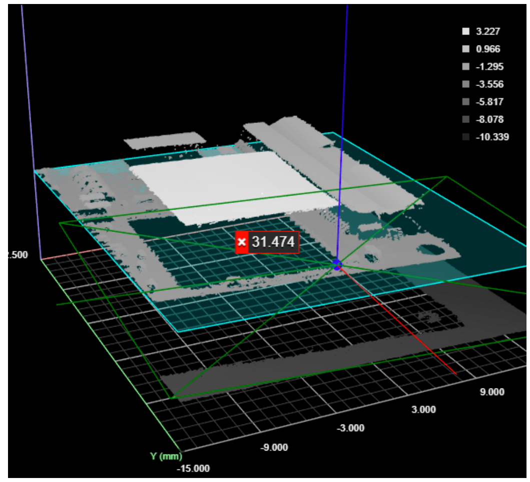
Setting	Description
Length	The dimension of the ruler perpendicular to the path. Be sure to use a value large enough to cover the track from one side to another and to include enough surface on each side of the track (the surface to which the material is applied) for the Track tool to properly detect the track.
Width	The dimension of the ruler along the path.
Space	The space between rulers on the path. Because you will typically place path points closer together around corners, you may need to use smaller spacing around corners.

Transform

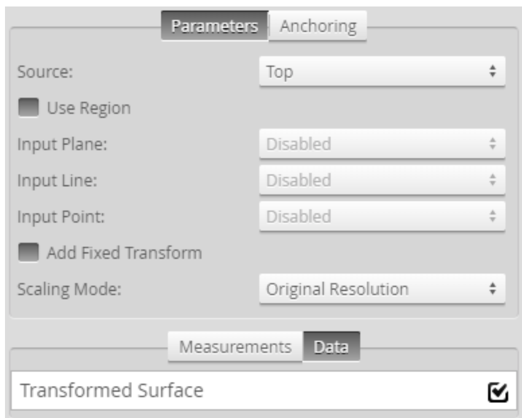
The Surface Transform tool generates a new surface based on the coordinate system of geometric features the tool uses as input. The tool can take a zero-plane, line, and origin point to define this new coordinate system. You can then apply the built-in measurement tools or GDK tools to this new surface data. This could let you, for example, get the height of a feature relative to a slightly tilted or warped adjacent or surrounding reference surface, rather than the absolute height in the original scan volume relative to the sensor. The result is increased repeatability of your measurements.



2D View

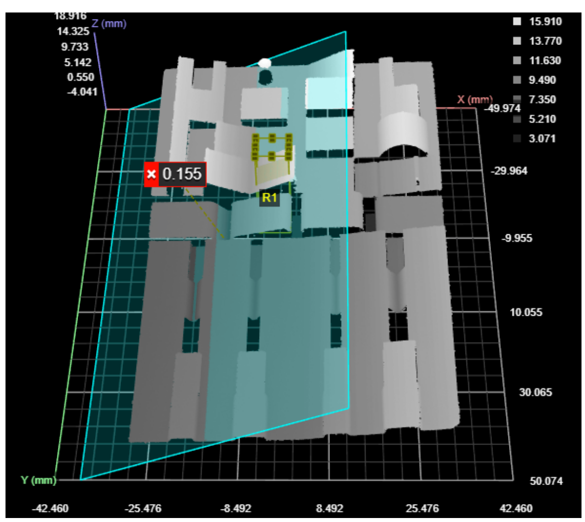


3D View

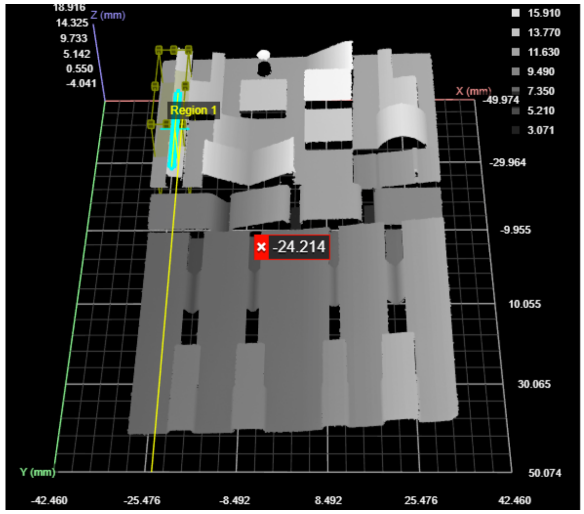


Measurement Panel

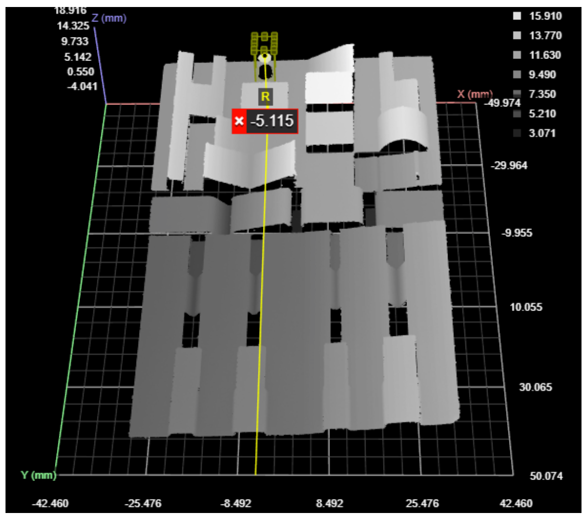
In *Combinations of geometric feature inputs and results* on page 468, the following geometric features are used by a Surface Transform tool in various combinations (a plane, a line, and a point).



A Surface Plane tool, with the region set to a small left-facing angled surface

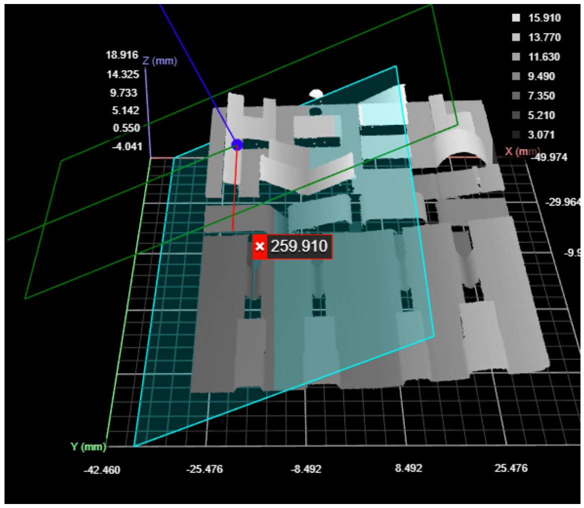


A Surface Edge tool, with the region set to the left edge of a raised surface (upper left of data viewer).



A Surface Position tool (maximum Z), with the region set to the raised point near the top of the data viewer.

Furthermore, in the sections below, two types of data are shown: the original (input) scan data and the transformed data. When the tool displays the original data, it overlays indicators of the new, transformed coordinate system on the data.



A Surface Transform tool using all three types of geometric feature inputs.
The data viewer is set to display the input surface data with an overlay of the transformed coordinate system.

In the data viewer, the following is displayed:

X, Y, and Z axes

The transformed axes are represented above by the red, green, and blue lines intersecting on the surface data above. Note how these are rotated with respect to the original coordinate system (the background grid, axes, and values along the axes).

Origin

The new origin is represented by the dark blue dot at the intersection of the transformed axes.

Plane

The new plane is represented by the cyan rectangle.

Bounding box containing the transformed surface

The bounding box that indicates where the transformed data is in relation to the original coordinate system.

🗋	To switch between the original and transformed data, choose Surface or Tool in the first drop-down above the data viewer, respectively.

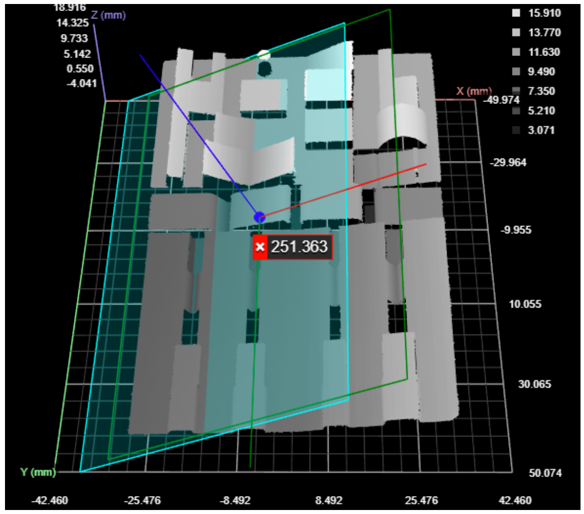
Combinations of geometric feature inputs and results

The Surface Transform tool accepts all combinations of input geometric features (plane, line, and point). For details and examples of each, see the following sections.

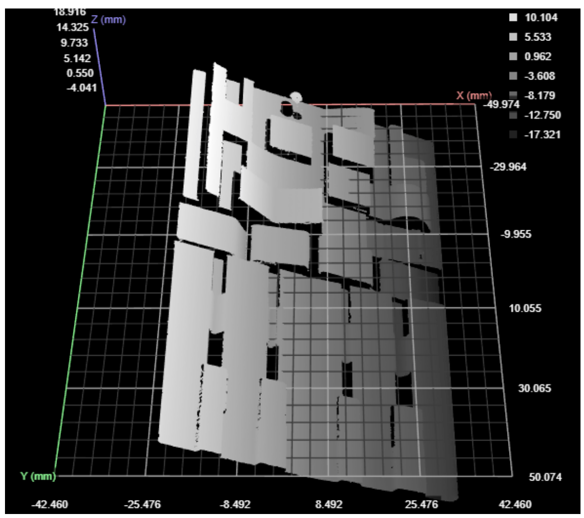
Plane

New Z=0 XY Plane	New X Axis	New Origin
Matches the input plane.	Parallel to the old X axis.	Old origin projected to plane.

Original data with overlay



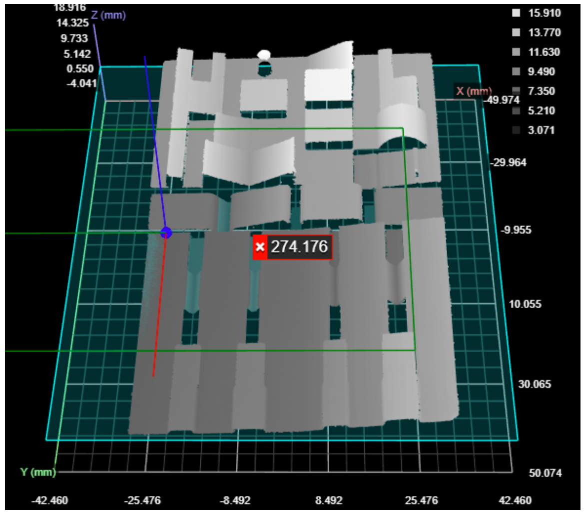
Transformed data



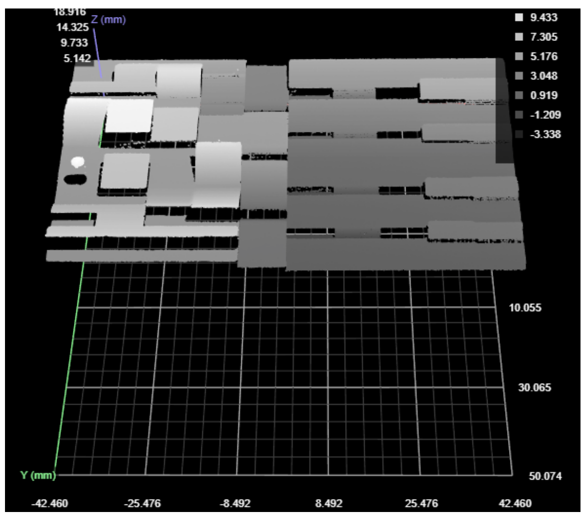
Line

New Z=0 XY Plane	New X Axis	New Origin
The new plane contains the line. The intersection of the new plane and the old plane is perpendicular to the input line.	Matches the line.	Old origin projected onto the line.

Original data with overlay



Transformed data

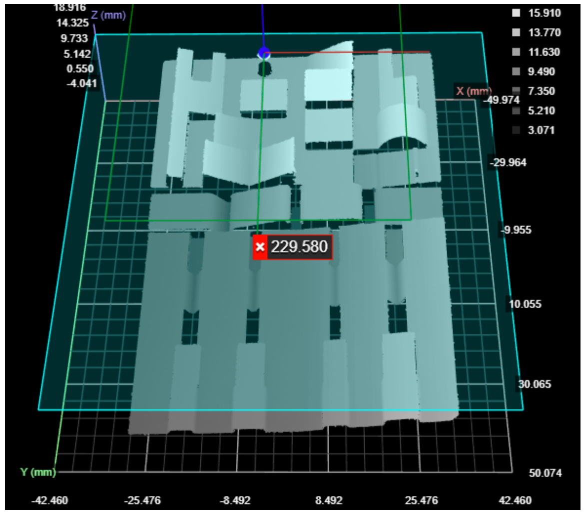


The direction of the X axis depends on the tool generating the line that Surface Transform takes as input. You may need to adjust the direction using the **Add Fixed Transform** settings.

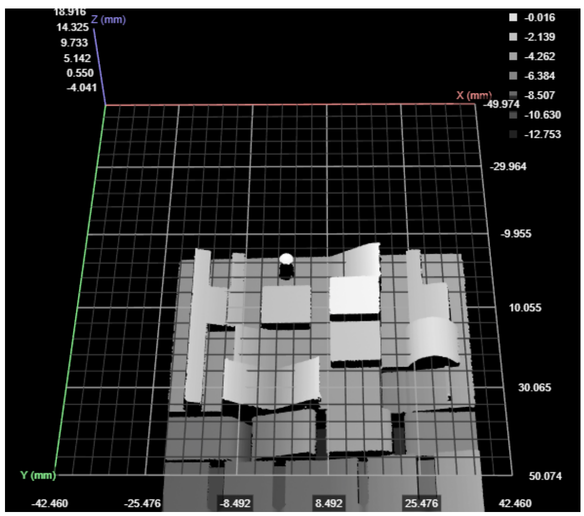
Point

New Z=0 XY Plane	New X Axis	New Origin
Through the input point, parallel to old Z=0 plane.	Parallel to the old axis.	The input point.

Original data with overlay



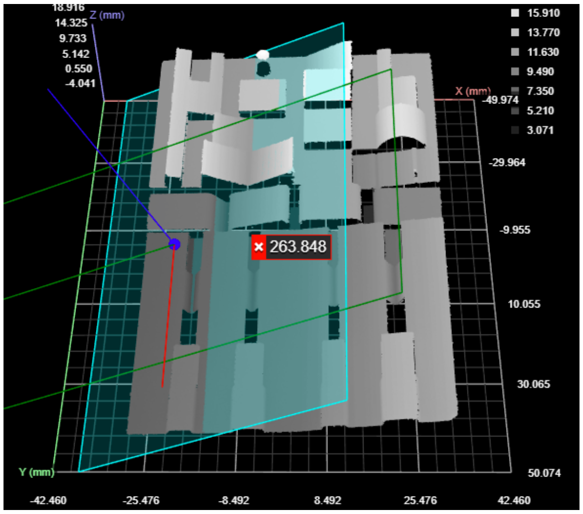
Transformed data



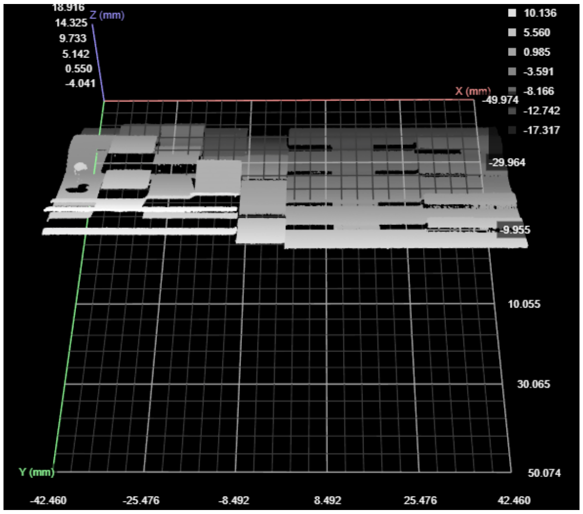
Plane + Line

New Z=0 XY Plane	New X Axis	New Origin
Matches the input plane.	Line projected onto the plane.	Old origin projected onto the projected line.

Original data with overlay



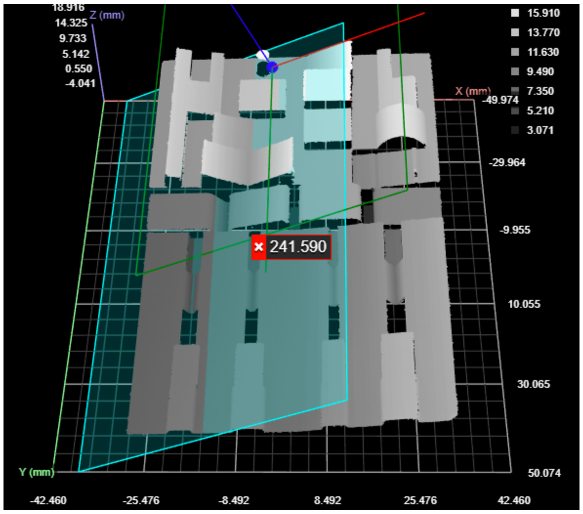
Transformed data



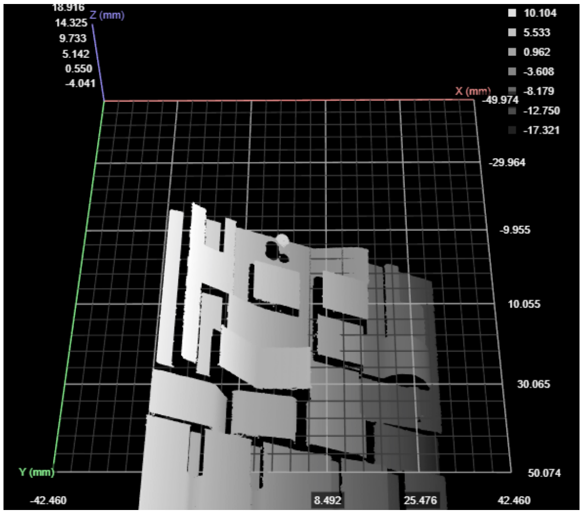
Plane + Point

New Z=0 XY Plane	New X Axis	New Origin
Matches the input plane.	Parallel to the old X axis.	At the input point, projected onto the plane.

Original data with overlay



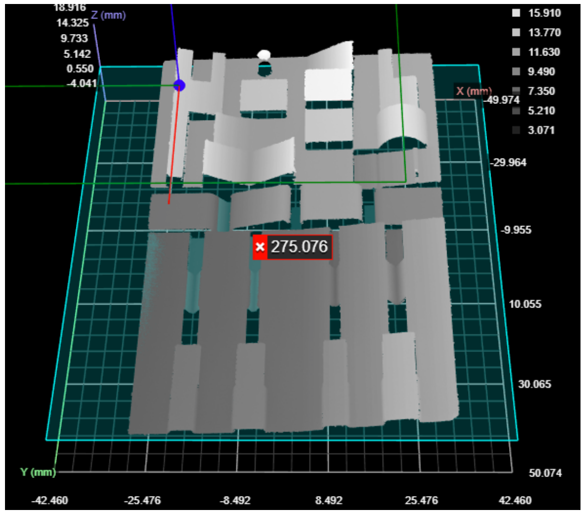
Transformed data



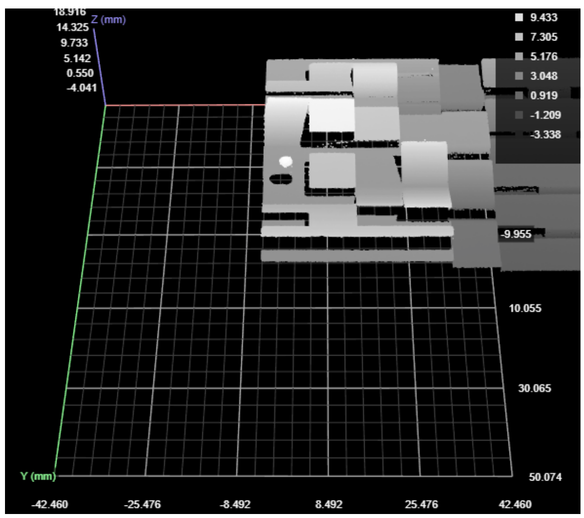
Line + Point

New Z=0 XY Plane	New X Axis	New Origin
The new plane contains the line. The intersection of the new plane and the old plane is perpendicular to the input line.	Matches the line.	The input point projected onto the line.

Original data with overlay



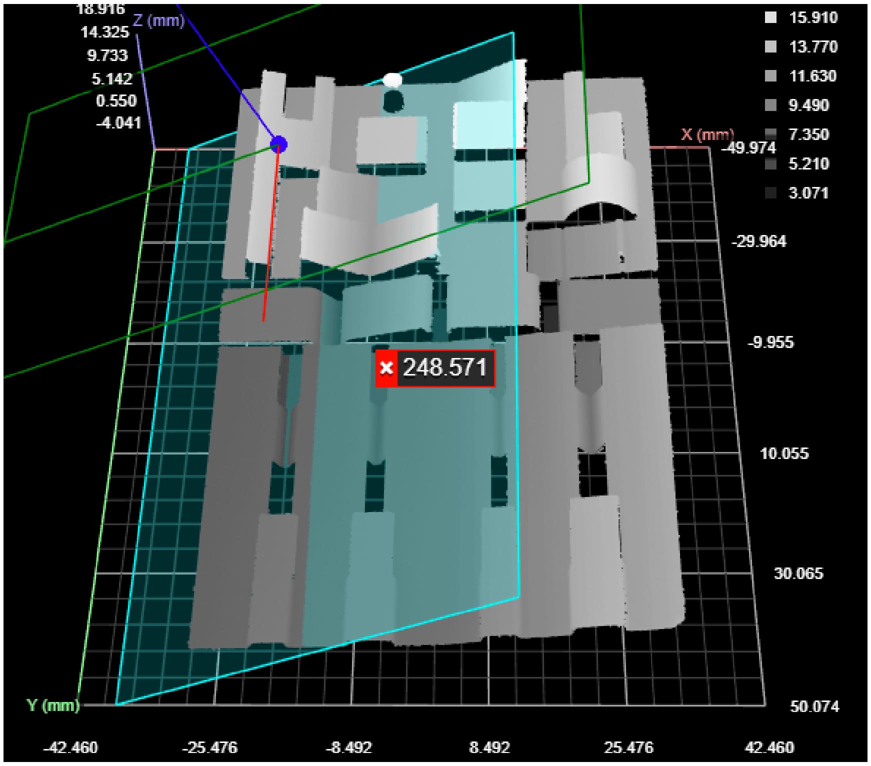
Transformed data



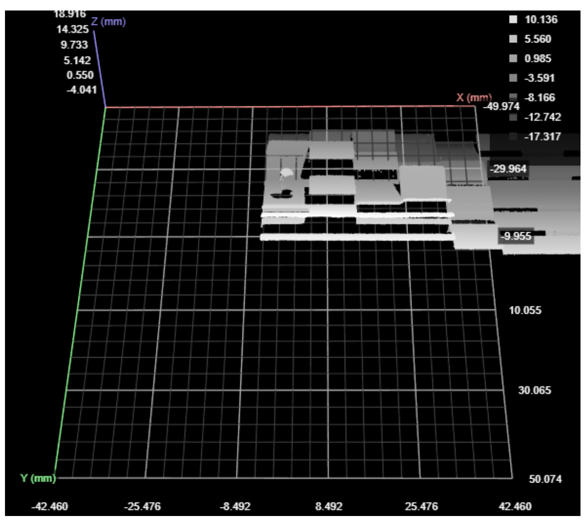
Plane + Line + Point

New Z=0 XY Plane	New X Axis	New Origin
Matches the input plane.	The input line projected onto the plane.	The input point projected onto the input line.

Original data with overlay



Transformed data



For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements, Data, and Settings

Measurements

Measurement
Running Time (ms) The amount of time required for tool execution. Used for diagnostic purposes.

Data

Type	Description
Transformed Surface	The transformed surface. Available via the Stream drop-down in other tools.

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Input Plane	The plane the tool uses to transform the surface scan data.
Input Line	The line the tool uses to transform the surface scan data.
Input Point	The point the tool uses to transform the surface scan data.
Scaling Mode	On G3 sensors, leave this set to the default **Original Resolution**.
Add Fixed Transform	When enabled, displays X, Y, and Z offset and angle fields you can use to set additional transformations, which are applied after any transformations supplied by the input geometric features. Setting a fixed transformation can be useful if the geometric features the tool uses results in data rotated to an unusual orientation; you could, for example, rotate the data 90 or 180 degrees so that it is in the "expected" orientation, or shift it so that it's easier to work with.
Use Region	When this setting is enabled, the tool only outputs the surface contained in the defined region.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.

> A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

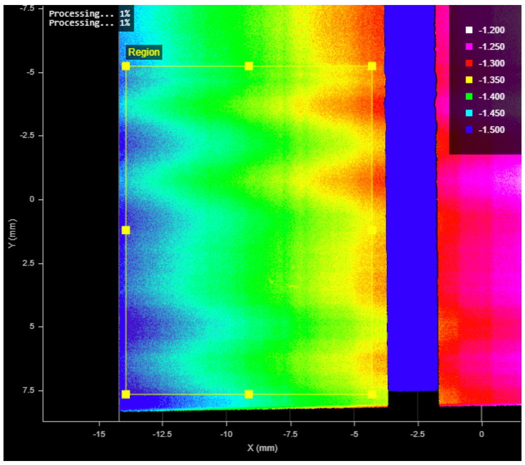
For more information on anchoring, see *Measurement Anchoring* on page 169.

Vibration Correction

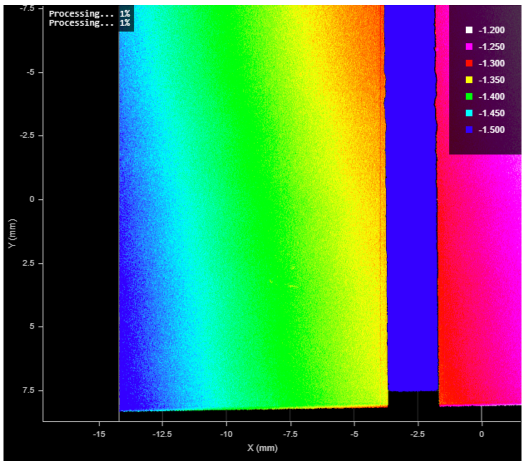
> This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Vibration Correction tool analyzes variation in surface data to remove high frequency noise in the data. The tool is useful for improving repeatability and accuracy of measurements when subtle vibrations in your transport system introduce height variations. The tool's intended use is to send corrected surface data to other tools.

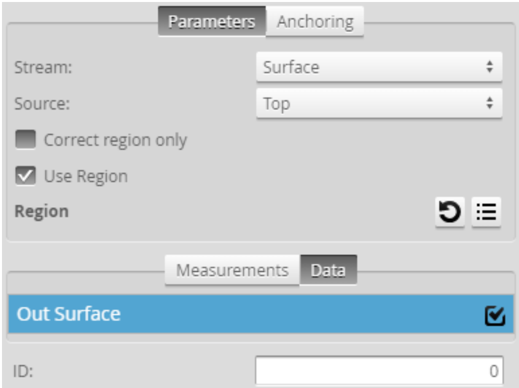
> The Vibration Correction tool requires at least 64 lines of data in the surface data it receives as input to be able to output corrected surface data.



Uncorrected surface data



Corrected surface data: a better representation of the actual target



Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Data and Settings

Data

Type	Description
Corrected Surface	Surface data corrected for vibration, available for use as input in the **Stream** drop-down in other tools.
Difference Surface	Diagnostic Surface data showing the difference between the corrected surface and the original. Available for use as input in the **Stream** drop-down in other tools

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Correct region only	If enabled, only the area under the region is corrected for vibration in the output surface data. This setting can be useful if vibration regularly occurs in a specific area of the scan data. This option is only displayed if **Use Region** is enabled.
Use Region	When enabled, lets you set a region and optionally choose to apply vibration correction only to that region (using **Correct region only**).
Region	The region whose data the tool will use to calculate the vibration correction.

Anchoring

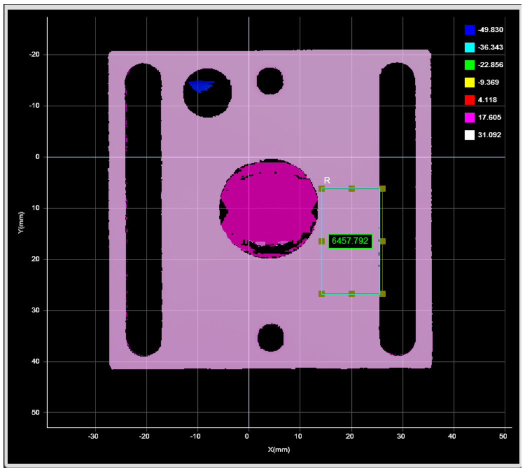
Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

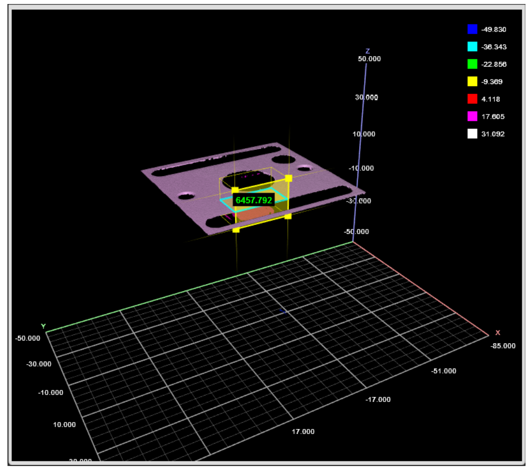
For more information on anchoring, see *Measurement Anchoring* on page 169.

Volume

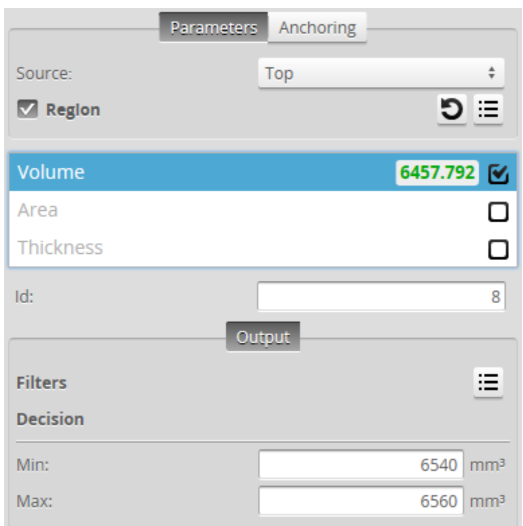
The Volume tool determines the volume, area, and thickness of a part.



2D View



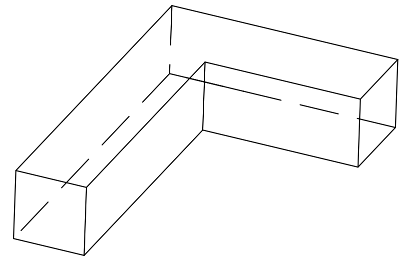
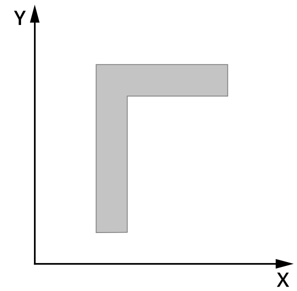
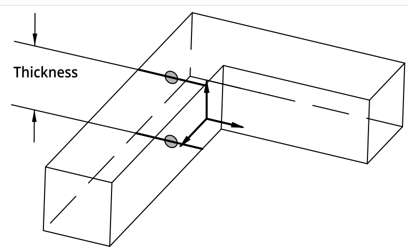
3D View



Measurement Panel

For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements

Measurement	Illustration
Volume Measures volume in XYZ space.	
Area Measures area in the XY plane. The area is the number of valid points multiplied by the X and Y resolution. Note that this is different compared to the area calculations produced by Surface Segmentation and Surface Blob; for more information, see the descriptions of the *Area {n}* measurements in *Segmentation* on page 426 and *Blob* on page 289.	
Thickness Measures thickness (height) of a part.	

Parameters

Parameter	Description
Source	The sensor that provides data for the tool's measurements. For more information, see *Source* on page 151.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 152.
Location *(Thickness measurement only)*	One of the following: • Max • Min • Average • Median • 2D Centroid (height of the centroid in the XY plane) • 3D Centroid (height of the centroid in the XYZ space).
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.

Parameter	Description
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Anchoring

Anchor	Description
X, Y, or Z	Lets you choose the X, Y, or Z measurement of another tool to use as a positional anchor for this tool.
Z angle	Lets you choose the Z Angle measurement of another tool to use as an angle anchor for this tool.

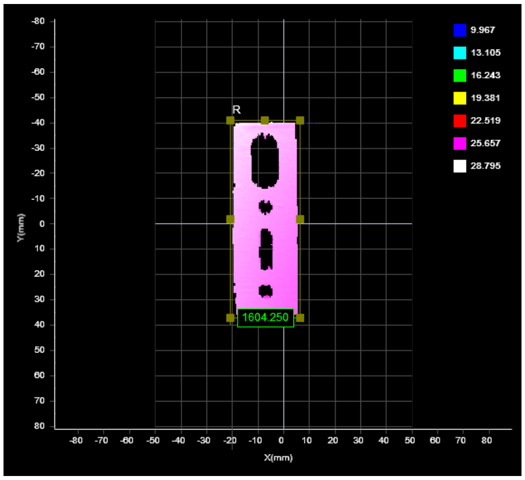
A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

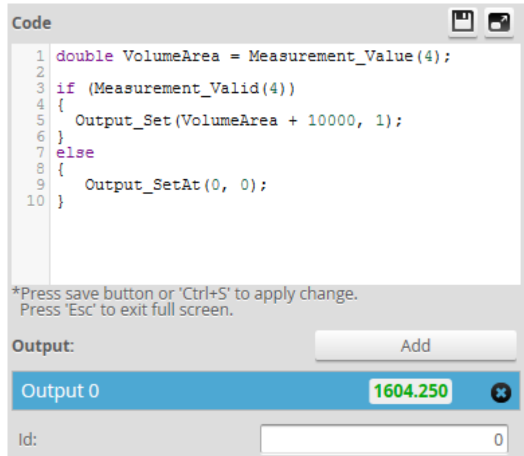
For more information on anchoring, see *Measurement Anchoring* on page 169.

Script

A Script measurement can be used to program a custom measurement using a simplified C-based syntax. A script measurement can produce multiple measurement values and decisions for the output.

For more information on script tool syntax, see *Scripts* on page 505.





To create or edit a Script measurement:

1. Add a new Script tool or select an existing Script measurement.

2. Edit the script code.

3. Add script outputs using the **Add** button.
   For each script output that is added, an index will be added to the **Output** drop-down and a unique ID will be generated.
   To remove a script output, click on the button next to it.

4. Click the **Save** button to save the script code.
   If there is a mistake in the script syntax, the result will be shown as a "Invalid" with a red border in the data viewer when you run the sensor.
   Outputs from multiple measurement tools can be used as inputs to the script. A typical script would take results from other measurement tools using the value and decision function, and output the result using the output function. Stamp information, such as time and encoder stamps, are available in the script, whereas the actual 3D point cloud data is not. (The script engine is not powerful enough to process the data itself.) Only one script can be created.

Feature Measurement

The following sections describe Gocator's Feature tools.

Feature tools produce measurements based on more complex geometry, letting you implement applications more quickly by reducing dependence on writing scripts to accomplish these kinds of measurements. Feature tools take geometric features generated by other tools as input and perform measurements on those features.

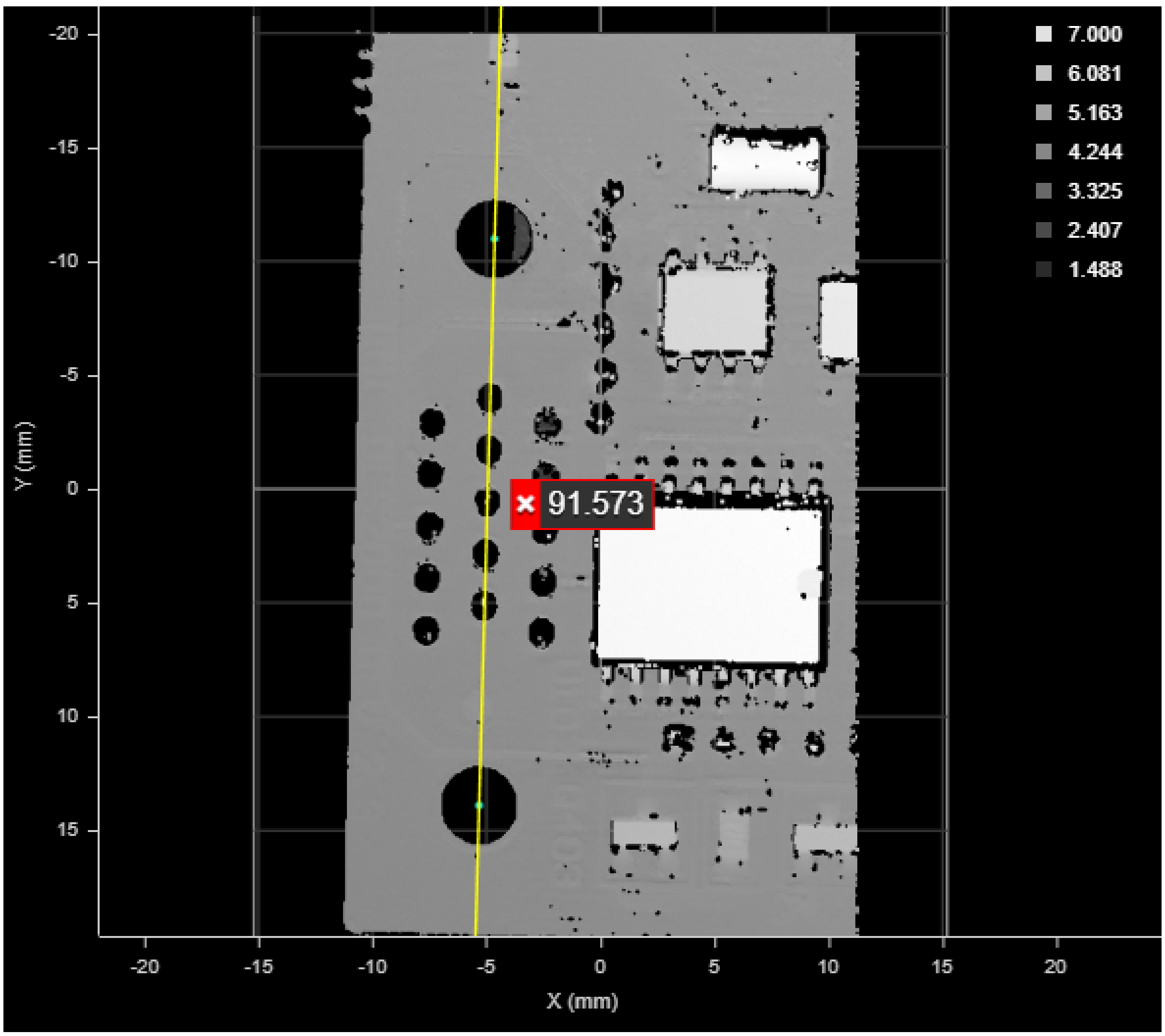
Feature tools are available in either Profile or Surface mode.

> The Circle geometric feature currently cannot be used by any of the built-in Feature tools.

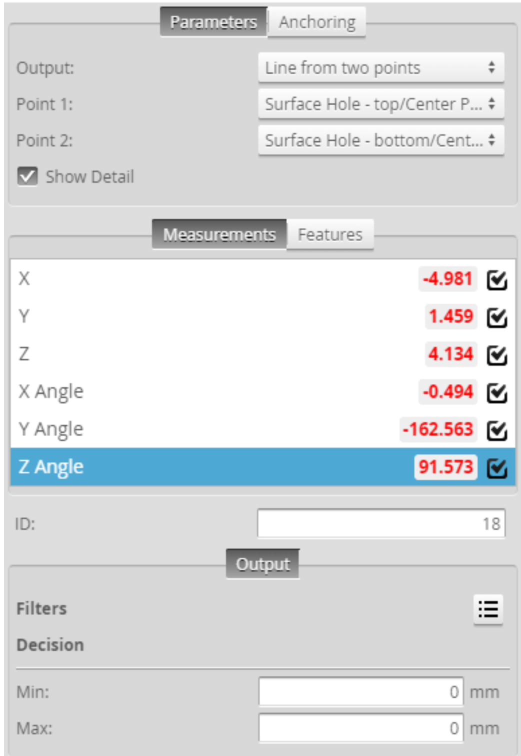
Create

The Feature Create tool lets you generate geometric features from other geometric features (ones generated by other tools). For example, you can create a line from two points, or create a plane from a point and a line. The tool can generate points, lines, circles, or planes. You can also extract measurement values from the geometric features generated by other tools; you can use these values as decisions or use them as anchors in other tools. The advantage of the Feature Create tool is that it means you need to rely less on Script tools or SDK/GDK applications to perform complex geometric operations.

For example, in the following, a Feature Create tool takes the hole geometric features output by two Surface Hole tools to generate a line geometric feature (near-vertical yellow line between the cyan hole center points).



You could perform measurements on the resulting line (X, Y, and Z positional measurements on the line's center point, and, more importantly, angle measurements on the line). You could also use the line's Z angle as an anchor in other tool's in order to increase repeatability.



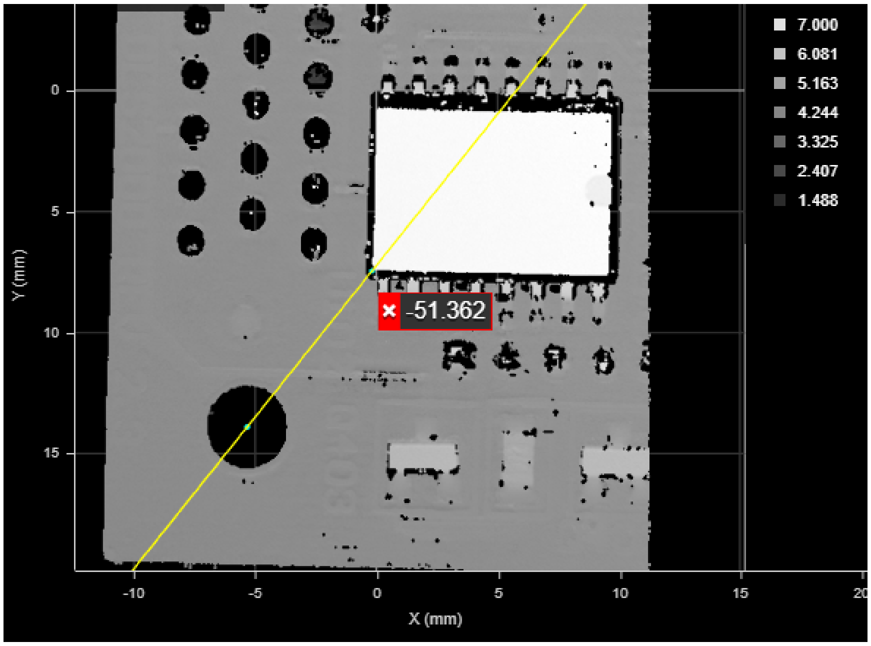
Measurement Panel

The following sections describe the output types available in the **Output** drop-down, the inputs required by each output, and the resulting output.

Line from Two Points

The **Line from two points** type of output takes two point geometric features as input.

The resulting output is a line geometric feature connecting the two points.



A line between the center point of a hole and the corner of the chip.
(The corner is the intersect point resulting from the Feature Intersect tool, taking the left vertical and lower horizontal line edges of the chip as input.)

The X, Y, and Z measurements return the midpoint of the line. The X, Y, and Z Angle measurements return the angle of the line.

Perpendicular or Parallel Line from Point and Line

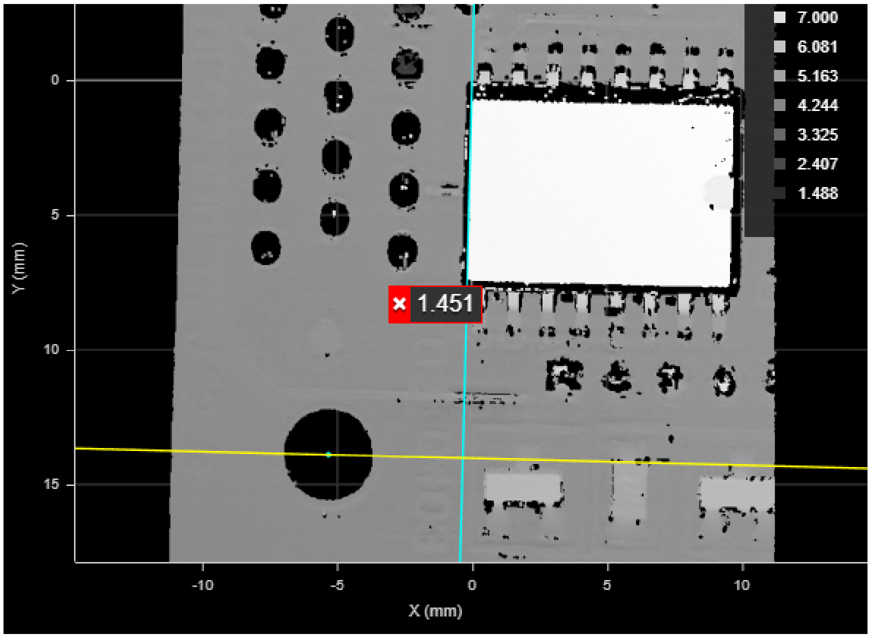
These types of output take a point and a line geometric feature as input to create another line.

For both of these types of line output, the X, Y, and Z measurements return the position of the point.

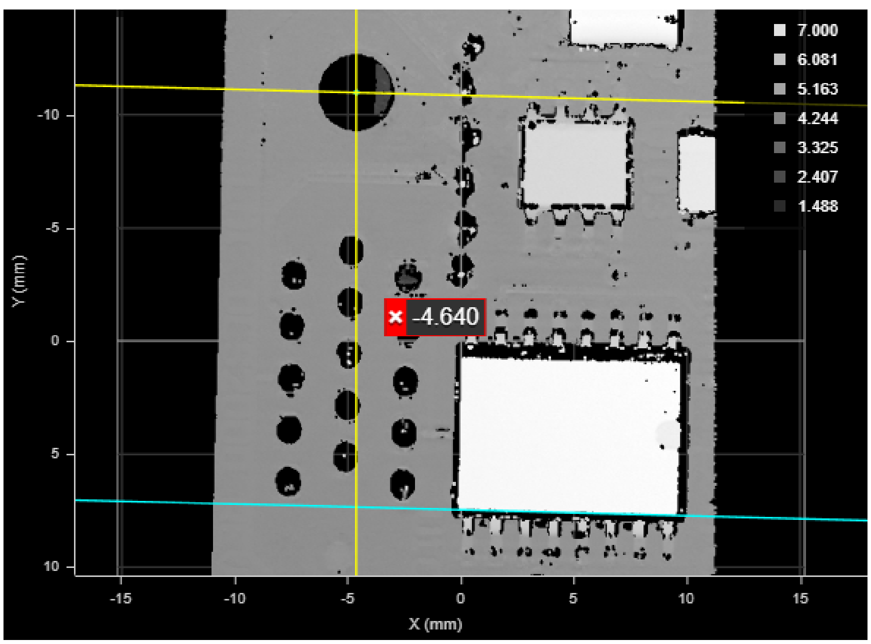
For perpendicular line output, the X, Y, and Z angle measurements return the angles of the line.

For parallel line output, the Z angle measurement returns the angle of the line; the X and Y angle measurements both return 180.000.

In the following, the tool generates a roughly vertical line (yellow) perpendicular to the input line (cyan line along the left edge of the large integrated circuit), passing through the input point (cyan dot at the center of the hole).



In the following, the tool generates a roughly horizontal line (yellow) parallel to the input line (cyan line along the bottom edge of the large integrated circuit), passing through the input point (cyan dot at the center of the hole).



Perpendicular Line from Point to Plane

Creates a perpendicular line from a point up to a plane.

Projected Point on Plane

Creates a point projected onto a plane.

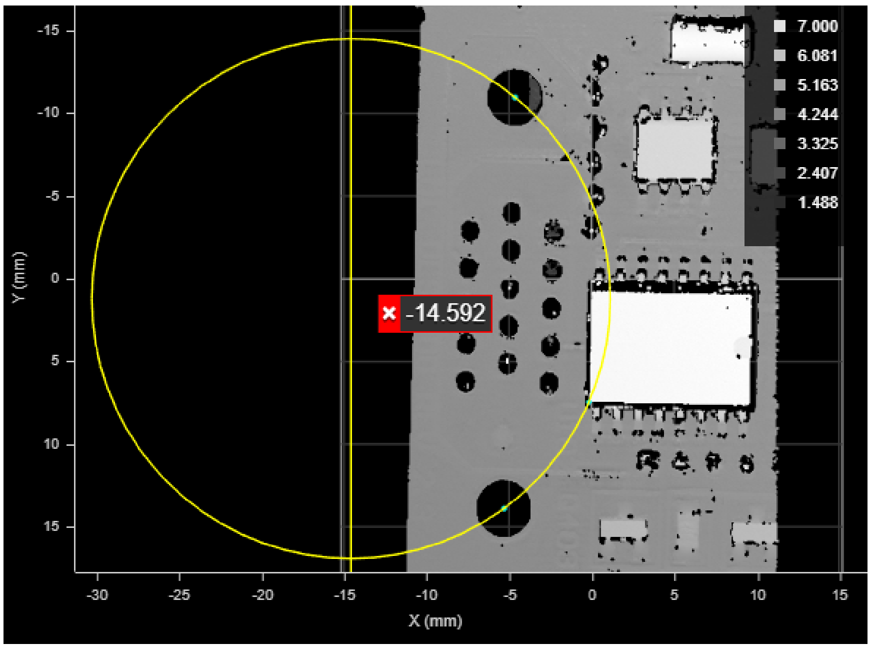
Projected Line on Plane

Creates a line projected onto a plane.

Circle from Points

The **Circle from points** output type takes three point geometric features and fits a circle to those points. The circle is always on the XY plane.

The X, Y, and Z measurements return the center of the circle. The X, Y, and Z Angle measurements return a 0.000 value.



Circle generated from the center points of the two holes and the corner of the chip (cyan points).
(The corner is the intersect point resulting from the Feature Intersect tool,
taking the left vertical and lower horizontal line edges of the chip as input.)

Plane from Point and Normal

Creates a plane from a point and a normal.

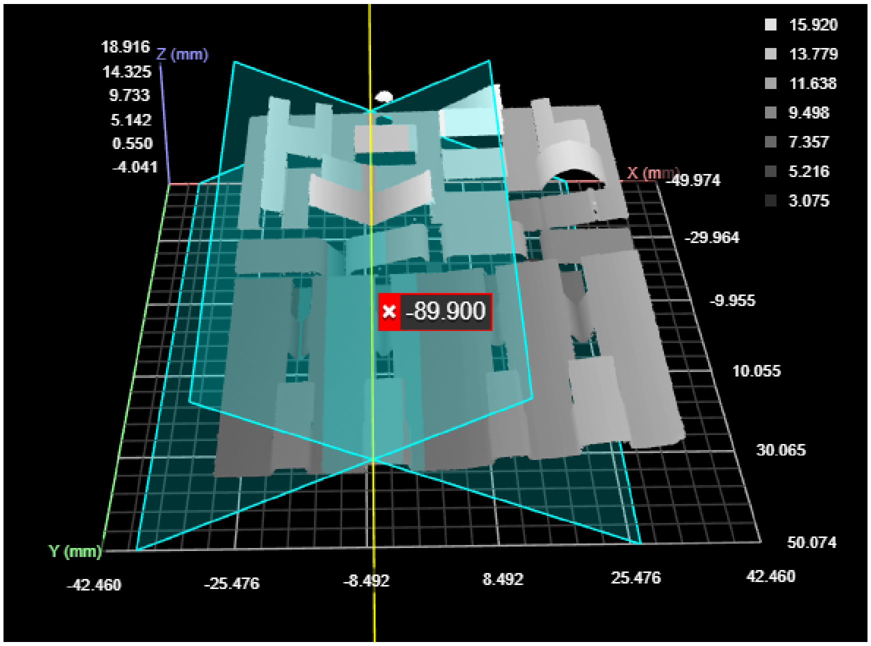
Plane from Three Points

Creates a plane from three points.

Line from Two Planes

The **Line from two planes** output type takes two plane geometric features as input and creates a line at their intersection.

The X, Y, and Z measurements return the midpoint. The X, Y, and Z Angle measurements return the angle of the line.

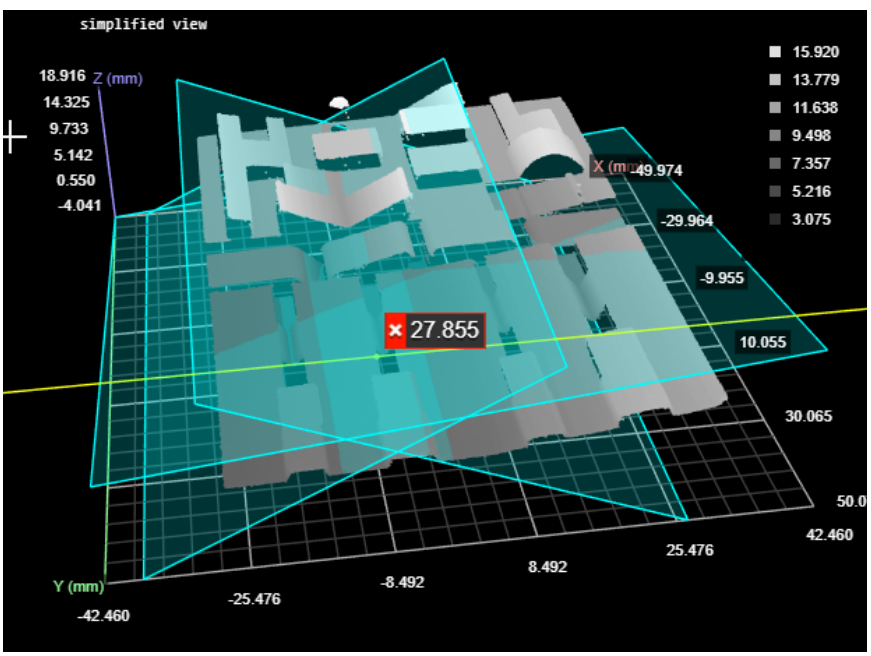


A line generated at the intersection of two planes. The Z angle is indicated.

Point from Three Planes

The **Point from three planes** output type takes three plane geometric features as input and creates a point at their intersection.

The X, Y, and Z measurements return the position of the intersect point. The X, Y, and Z Angle measurements return 0.000 values.



A point generated at the intersection of two planes. The Y position is indicated here.

Point from Line and Circle

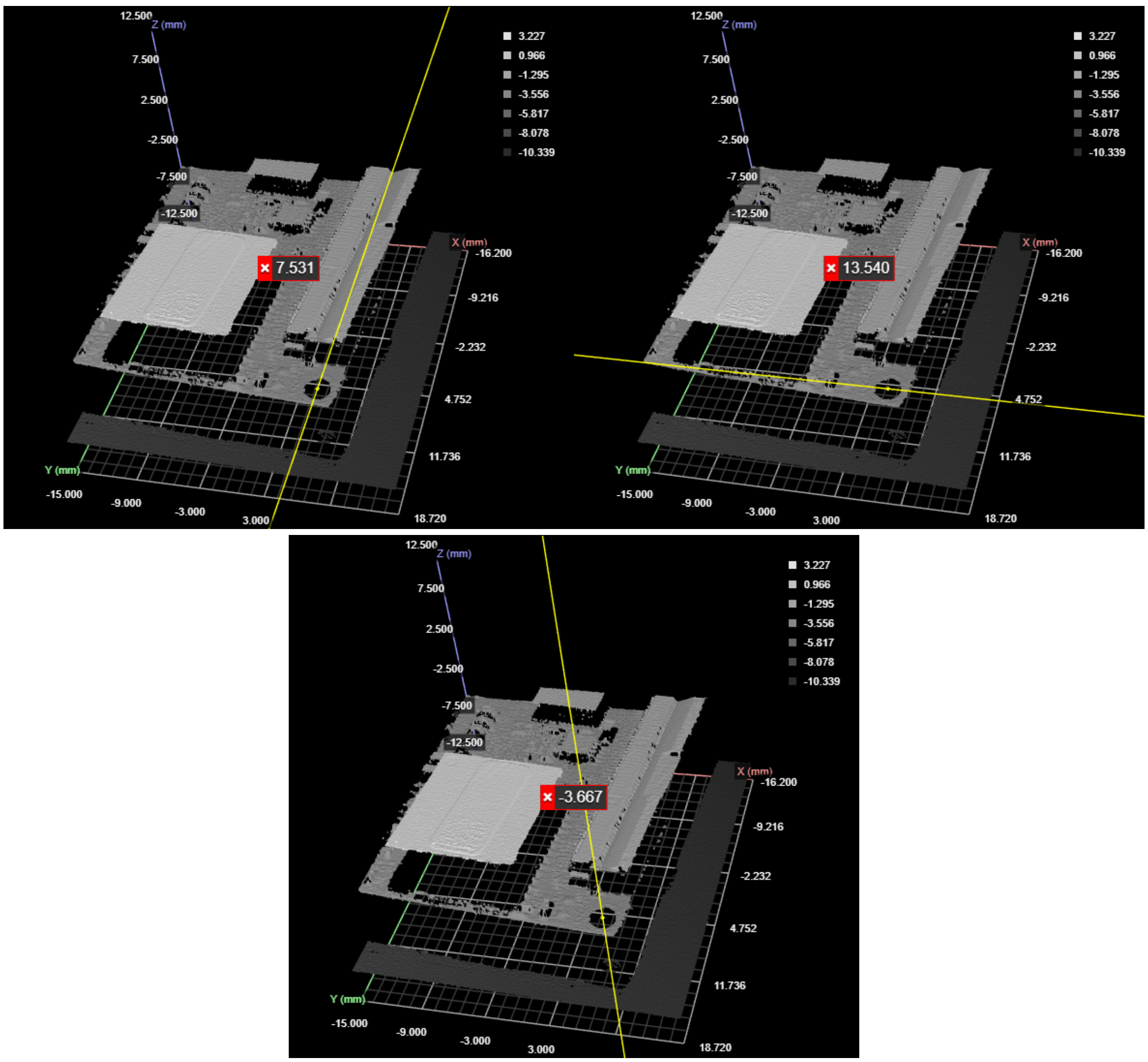
Creates a point from a line and a circle (their intersection).

Point or Line

The **Point** and **Line** types of output take a point or a line geometric feature as input, respectively.

These outputs are useful if the tool takes features generated by another Feature Create tool as input, on which you want to perform measurements in the second Feature Create tool. Also, this can be useful if you have developed GDK tools that only generate geometric features (no measurements): you can use this tool to extract those measurements.

For point output, the X, Y, and Z measurements return the X, Y, and Z position of the point; the angle measurements all produce 0.000 as values.



Positional measurements of a point

For line output, the X, Y, and Z measurements return the midpoint of the line. The Z Angle measurement returns the angle of the line around the Z axis. The X angle is always 0.000, and the Y angle is always 180.000.

See *Adding and Configuring a Measurement Tool* on page 149 for instructions on how to add measurement tools.

Measurements

Measurement	Illustration
X, **Y**, **Z** The X, Y, and Z positions of some aspect of the geometric feature. For more information, see the sections above.	
X Angle, **Y Angle**, **Z Angle** The X, Y, and Z angles of some aspect of the geometric feature. For more information, see the sections above.	

Note that even when enabled on the **Features** tab, not all features are generated. (For example, with Line selected as the output type, only a line geometric feature can be generated: point, circle, and plane features are not generated.)

Features

Type	Description
Point	The generated point geometric feature.
Line	The generated line geometric feature.
Circle	The generated circle geometric feature.
Plane	The generated plane geometric feature.

Parameters

Parameter	Description
Output	The type of output the tool generates. Switching between the options changes the input types displayed in the tool.
Show Detail	Toggles the display of the input geometric features in the data viewer.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Dimension

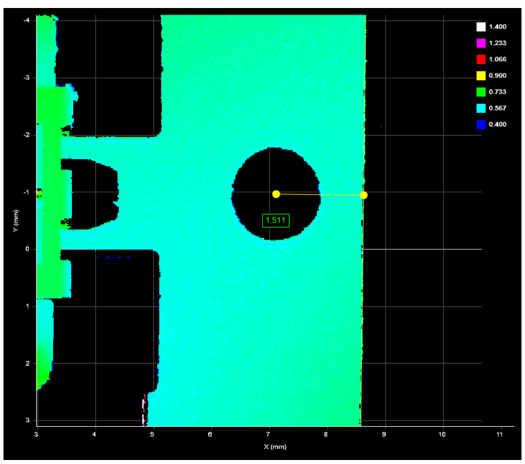
The Feature Dimension tool provides dimensional measurements from a point geometric feature to a reference point, line, or plane geometric feature.

Some examples:

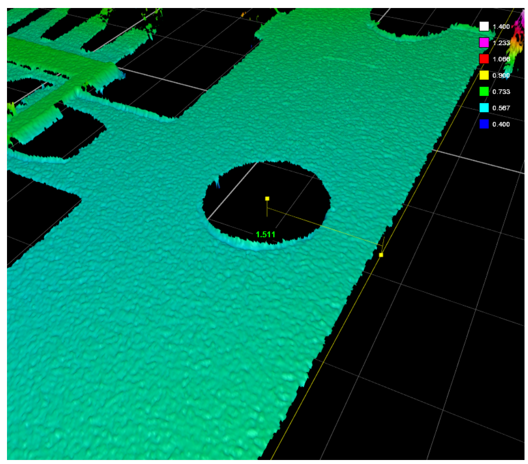
- Measuring the distance between the center of a hole and an edge.
- Measuring the distance between the centers of two holes.
- Measuring the distance between a point and a plane.
- Measuring the distance between a point and the closest point on a circle.
- Obtaining the length of a stud by measuring the distance between its tip and base.

The sensor compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 166.

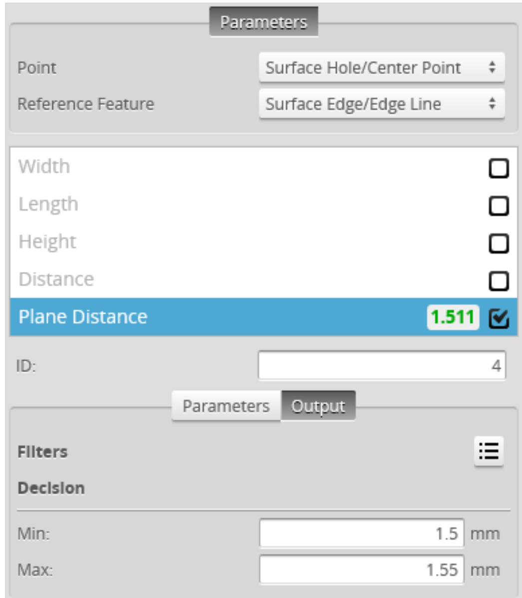
See *Adding and Configuring a Measurement Tool* on page 149 for instructions on how to add measurement tools.



2D View



3D View

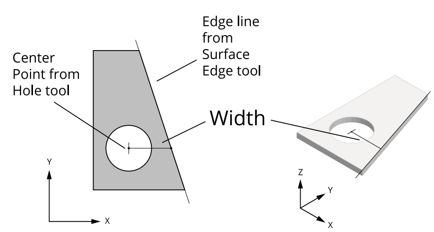


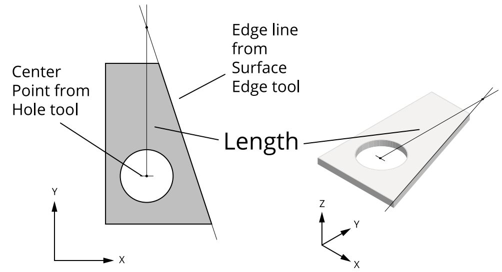
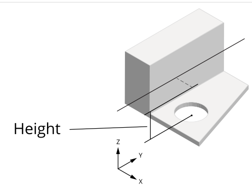
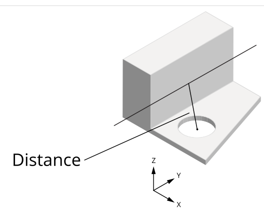
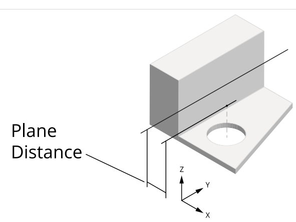
Measurement Panel

In the following measurement descriptions, the first geometric feature is set in the **Point** drop-down. The second geometric feature is set in the **Reference Feature** drop-down.

When **Reference Feature** is set to a feature other than a point, such as a circle or a line, measurements are between the point in **Point** and the *nearest point* on the reference feature (for example, the nearest point on a circle).

Measurements

Measurement	Illustration
Width *Point-point*: The difference on the X axis between the points. *Point-line*: The difference on the X axis between the point and a point on the line. For profiles, the point on the line is at the same Z position as the first point. For surface data, the point on the line is at the same Y position. *Point-plane*: The difference on the X axis between a point and a point on the plane with the same Y and Z coordinates as the first point (or the intersection of the plane and a line from the first point, parallel to the X axis).	

Measurement	Illustration
Length *Point-point*: The difference on the Y axis between the points. *Point-line*: The difference on the Y axis between the point and, for profiles, the nearest point on the line; currently, always zero. For surface data, the point on the line is at the same X position as the first point. *Point-plane*: The difference on the Y axis between the point and a point on the plane with the same X and Z coordinates as the first point (or the intersection of the plane and a line from the first point, parallel to the Y axis).	
Height *Point-point*: The difference on the Z axis between the points. *Point-line*: The difference on the Z axis between the point and, for profiles, a point on the line at the same X position as the first point. For surface data, the point on the line is the one nearest to the first point. *Point-plane*: The difference on the Z axis between the point and a point on the plane with the same X and Y coordinates as the first point (or the intersection of the plane and a line from the first point, parallel to the Z axis).	
Distance *Point-point*: The direct, Euclidean distance between two point geometric features. *Point-line*: The direct, Euclidean distance between a point and the nearest point on the line. *Point-plane*: The direct, Euclidean distance between a point and the nearest point on the plane.	
Plane Distance *Point-point*: The distance between two point geometric features. For profile data, the points are projected onto the XZ plane (always the same as the Distance measurement). For surface data, the points are projected onto the XY plane. *Point-line*: The distance between a point and a line. For profile data, projected onto the XZ plane (always the same as the Distance measurement). For surface data, the distance is projected onto the XY plane. *Point-plane*: The distance between a point and a plane. For profiles, the distance is projected onto the XZ plane (always the same as the Distance measurement). For surface data, the distance is projected onto the XY plane.	

Parameters

Parameter	Description
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool. If you switch from one type of data to another (for example, from section profile data to surface data), currently set input features will become invalid, and you will need to choose features of the correct data type.
Point	A point geometric feature generated by another tool.
Reference Feature	A feature generated by another tool. Dimensional measurements are calculated *from* the reference feature *to* the point in the **Point** setting.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Intersect

The Feature Intersect tool returns the intersection of a line or plane geometric features and a second line or plane geometric feature. For line-line intersections, the lines are projected onto the Z = reference Z line plane for features extracted from a surface, and the intersection of the lines projected onto the Y = 0 plane for features extracted from a profile. The angle measurement between the two lines is also returned. The lines the tool takes as input are generated by other tools, such as Surface Edge or Surface Ellipse.

The Feature Intersect tool saves you from having to write complicated calculations in script tools to find intersect point between lines. Previously, calculating the intercept point of two lines was difficult and prone to bugs, involving finding lines in indirect ways.

The Feature Intersect tool's positional measurements are particularly useful as anchor sources. For example, you can easily find a corner point on a part from two edges (produced by two Surface Edge tools) and using the X and Y positions as anchor sources.

When you use these positional anchors in combination with a Z Angle anchor from tools such as Surface Edge, you can achieve extremely robust, repeatable measurements.

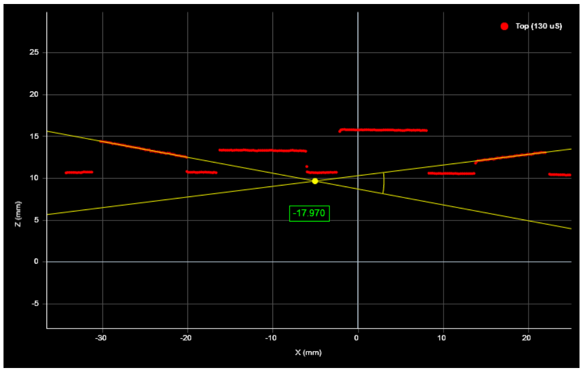
🗋	This tool's Angle measurement cannot be used as an angle anchor source. Only Z Angle measurements can be used as angle anchor sources.

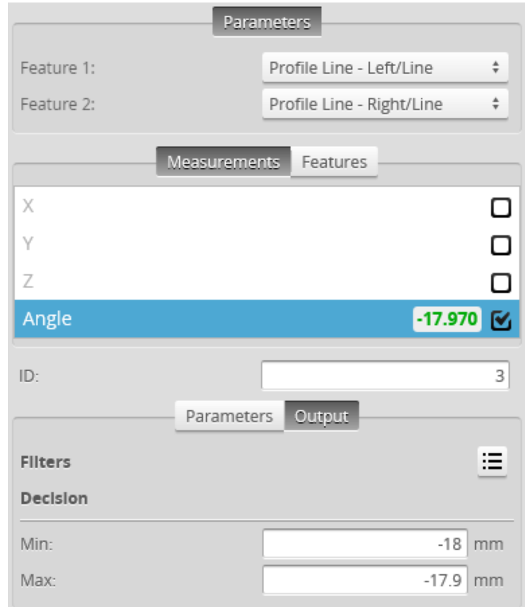
For more information on anchoring, see *Measurement Anchoring* on page 169.

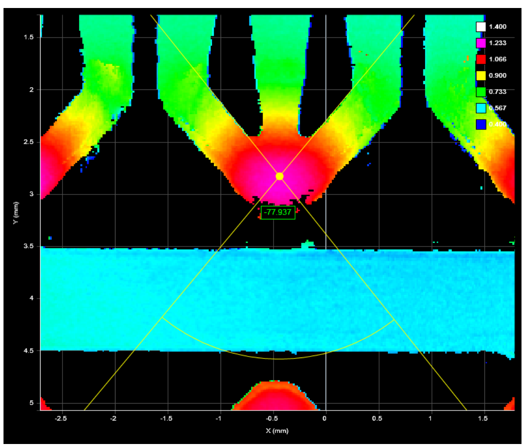
The Feature Intersect tool can also generate a point geometric feature representing the point of intersection of the lines that the Feature Dimension tool can use in measurements.

The sensor compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 166.

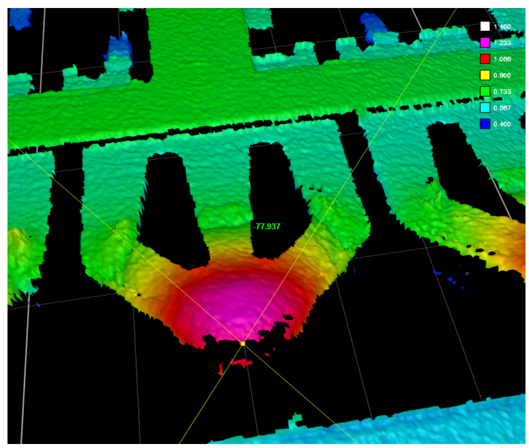
See *Adding and Configuring a Measurement Tool* on page 149 for instructions on how to add measurement tools.



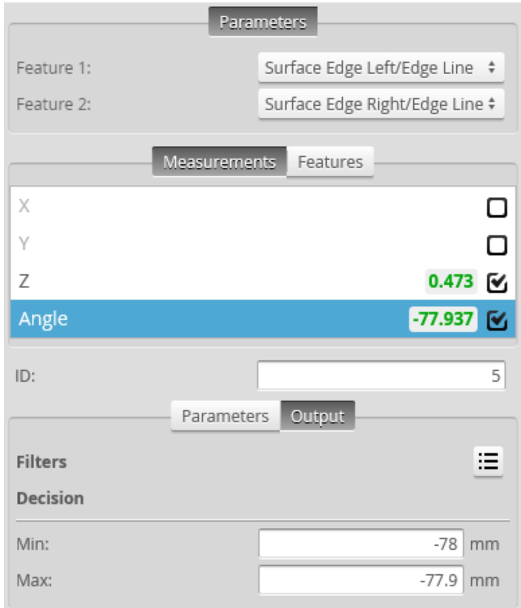




2D View

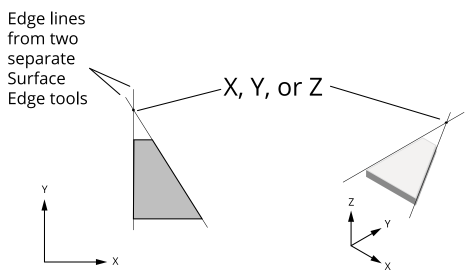


3D View



Measurement Panel

Measurements

Measurement	**Illustration**
X *Line-Line*: The X position of the intersect point between the lines. *Line-Plane*: The X position of the intersect point between the line and the plane. *Plane-Plane*: The X position of the center of the line intersecting the planes.	
Y *Line-Line*: The Y position of the intersect point between the lines. *Line-Plane*: The Y position of the intersect point between the line and the plane. *Plane-Plane*: The Y position of the center of the line intersecting the planes.	
Z *Line-Line*: The Z position of the intersect point between the lines. *Line-Plane*: The Z position of the intersect point between the line and the plane. *Plane-Plane*: The Z position of the center of the line intersecting the planes.	

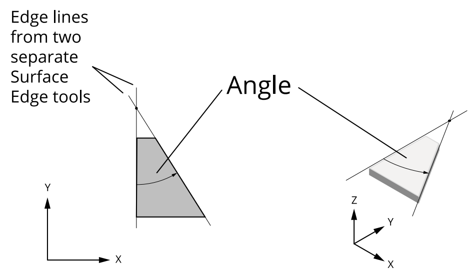
Measurement	Illustration
Angle	

Line-Line: The angle between the lines, as measured from the line selected in **Reference Feature** to the line selected in **Line**.

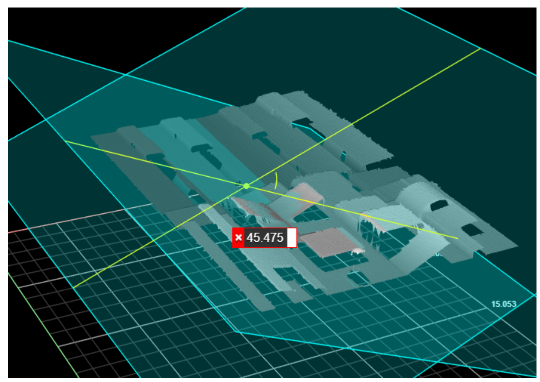
Line-Plane:The angle between the line and the perpendicular projection of the line onto the plane, as measured from the plane geometric feature selected in **Reference Feature** to the line selected in **Line**.

Plane-Plane:The angle between the two planes, as measured from the plane geometric features selected in **Feature 1** and **Feature 2**.

For line-line and line-plane angle measurements, the **Angle Range** setting determines how angles are expressed. (The setting does nothing with plane-plane angle measurements.)



In the following image, the angle is measured between two planes (the small angled surfaces facing each other in the center of the image).



Features

Type	Description
Intersect Point	The intersect point of the two features.

Parameters

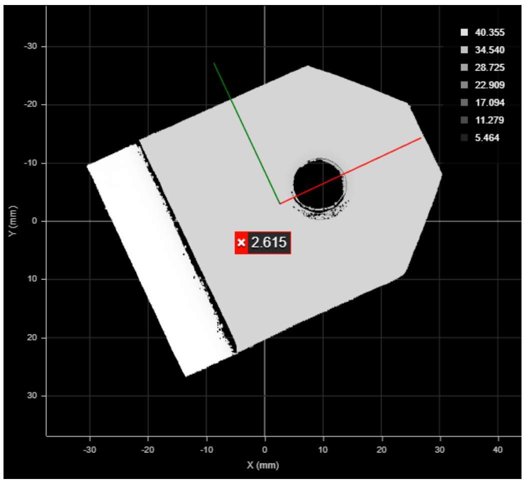
Parameter	Description
Stream	The data that the tool will apply measurements to. This setting is only displayed when data from another tool is available as input for this tool. If you switch from one type of data to another (for example, from section profile data to surface data), currently set input features will become invalid, and you will need to choose features of the correct data type.
Feature 1	A line or plane geometric feature generated by another tool.
Feature 2	A line or plane geometric feature generated by another tool. For the Angle measurement, the angle is measured *from* this feature.

Parameter	Description
Angle Range *(Angle measurement only; does nothing with plane-plane measurements)*	Determines the angle range.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

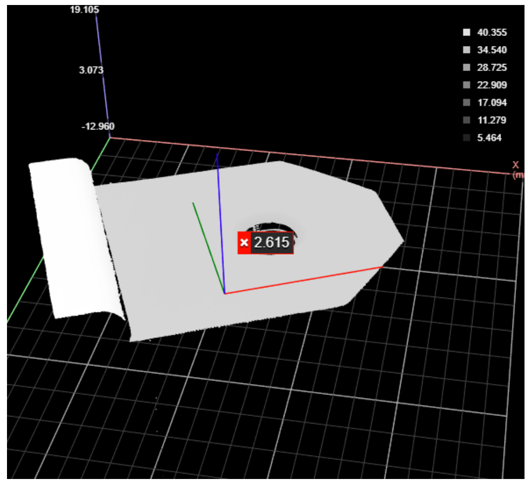
Robot Pose

The Feature Robot Pose tool takes geometric features as input and outputs positional and rotational values. You can use these values in a robot system to control the robot. If you are using a Universal Robots robot in your system and have calibrated the robot with LMI's Gocator URCap plugin, you can use the plugin's Receive node to retrieve these values to quickly and easily get the pose of a part. The URCap can then use the pose to move the robot to the position of the part, to pick up the part, for example. For more information on the Gocator URCap, see *Universal Robots Integration* on page 814.

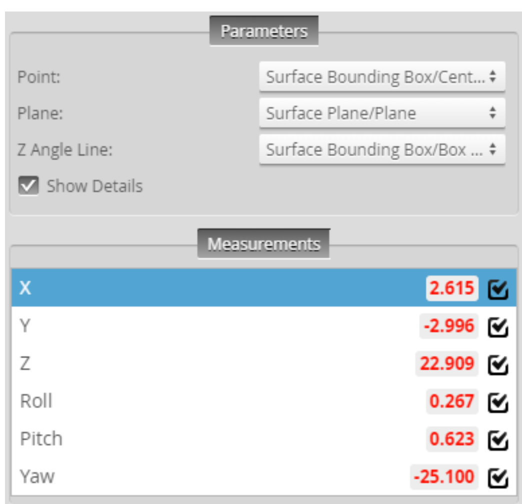
In the following images, the Robot Pose tool has returned positional (X, Y, and Z) and rotational (roll, pitch, and yaw) information on a part.



2D View



3D View

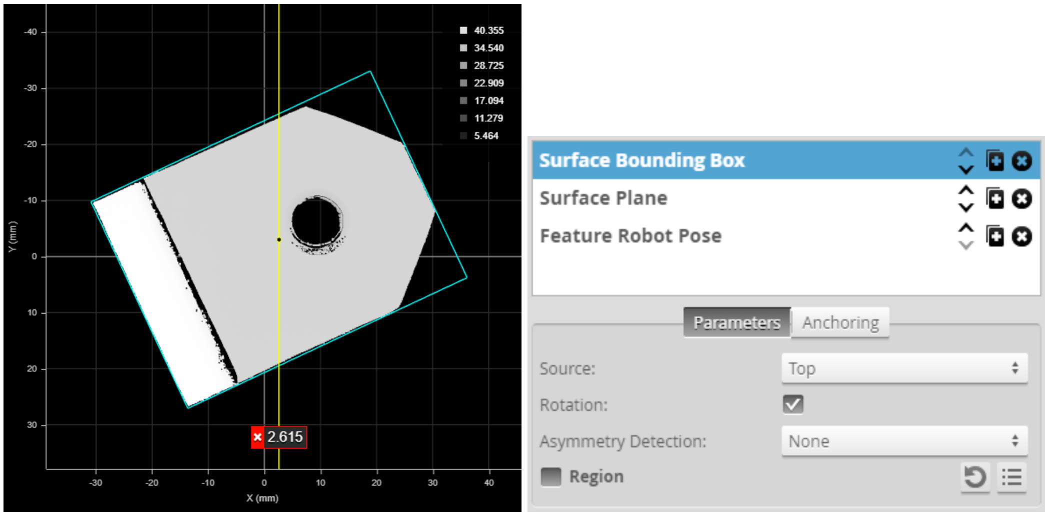


Measurement Panel

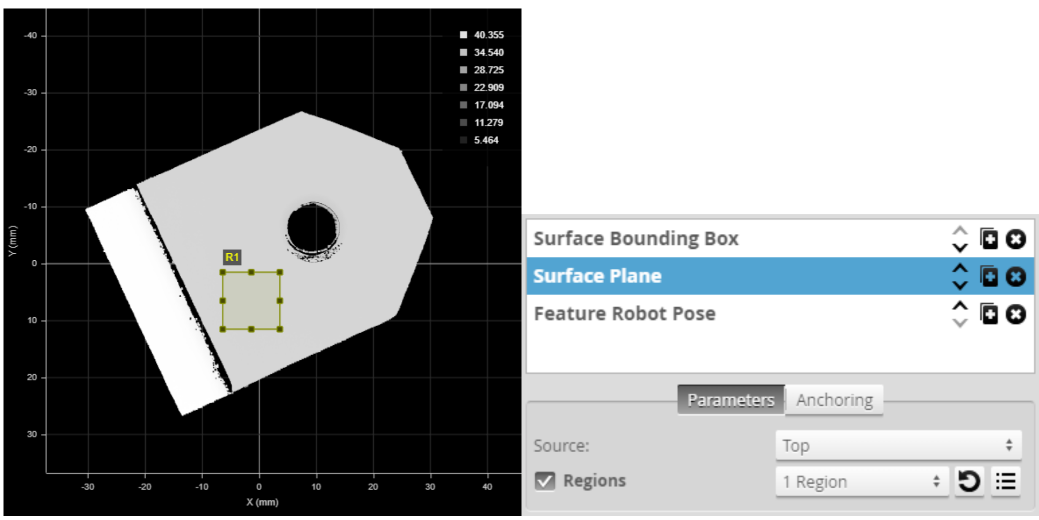
At a minimum, the Robot Pose tool needs the following input:

- A Point geometric feature to determine XYZ information
- A Plane geometric feature to determine roll and pitch (rotation around the X and Y axes)

Including a Line geometric feature lets the tool also return yaw (Z rotational information). For example, to get pose information for the part shown below, you could first configure a Surface Bounding Box tool and a Surface Plane tool.

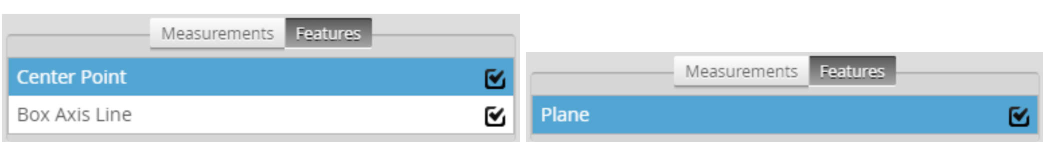


Bounding Box tool. The tool is configured to rotate to accommodate the orientation of the part.



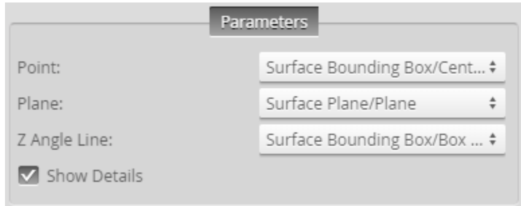
Surface Plane tool on flat area of part.

With both tools, you must enable the required feature outputs on the Feature tabs:



Enabled geometric features in Features tabs of Bounding Box and Plane tools, respectively.

Then select the features as input (the first three parameters) in the Robot Pose tool:



For information on adding, managing, and removing tools and measurements, as well as detailed descriptions of settings common to most tools, see *Tools Panel* on page 149.

Measurements and Settings

Measurements

Measurement	Illustration
X, **Y**, **Z** The X, Y, and Z positions of the Point geometric feature.	
Roll, **Pitch**, **Yaw** The rotational angles of the Plane and Line geometric feature.	

Data

Type	Description
Matrix	Data containing a matrix representing the same pose as the tool's measurements. It can be deserialized into a GoRobotMatrix structure using the GoRobot library.

Parameters

Parameter	Description
Point	The Point geometric feature the tool extracts the X, Y, and Z measurements from. This input is required.
Plane	The Plane geometric feature the tool extracts the Roll and Pitch measurements from. This input is required.
Z Angle Line	The Line geometric feature the tool extracts the Yaw measurement from. This input is optional. If you omit it, the X and Y axes will be parallel to the sensor's X and Y axes.
Show Details	Toggles the display of additional visualizations in the data viewer.
Filters	The filters that are applied to measurement values before they are output. For more information, see *Filters* on page 167.
Decision	The **Max** and **Min** settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see *Decisions* on page 166.

Scripts

Script tools use outputs from other measurement tools to produce custom measurements.

Similar to other measurement tools, a script tool can output multiple measurement values and decisions. Scripts are added, configured, and removed much like other measurement tools; for more information on this, see *Script* under *Profile Measurement* on page 206 or *Surface Measurement* on page 277.

> Scripts must be less than 27,000 characters long.

Script tools use a simplified C-based syntax. The following elements of the C language are supported:

Supported Elements

Elements	Supported
Control operators	if, while, do, for, switch and return.
Data types	char, int, unsigned int, float, double, long long (64-bit integer).
Arithmetic and logical Operator	Standard C arithmetic operators, except ternary operator (i.e., "condition?trueValue:falseValue"). Explicit casting (e.g., int a = (int) a_float) is not supported.
Function declarations	Standard C function declarations with argument passed by values. Pointers are not supported.
Array declarations	Standard C array declarations. For example: float measurements[5].
Standard arithmetic functions	+ , −, *, /, %, ++, --

Built-in Script Functions

The script engine provides the following types of functions:

- Measurement
- Output
- Memory
- Runtime variable
- Stamp
- Math

Measurement Functions

Function	Description
int Measurement_Exists(int id)	Determines if a measurement exists by ID. Parameters: id – Measurement ID Returns: 0 – measurement does not exist 1 – measurement exists
int Measurement_Valid(int id)	Determines if a measurement value is valid by its ID.

Function	Description
	Parameters: id - Measurement ID Returns 0 - Measurement is invalid 1 - Measurement is valid
double Measurement_Value (int id)	Gets the value of a measurement by its ID. Parameters: id - Measurement ID Returns: Value of the measurement 0 – if measurement does not exist 1 – if measurement exists
int Measurement_Decision (int id)	Gets the decision of a measurement by its ID. Parameters: ID - Measurement ID Returns: Decision of the measurement 0 – if measurement decision is false 1 – If measurement decision is true
int Measurement_NameExists(char* toolName, char* measurementName)	Determines if a measurement exist by name. Parameter: toolName – Tool name measurementName – Measurement name Returns: 0 – measurement does not exist 1 – measurement exists
int Measurement_Id (char* toolName, char* measurementName)	Gets the measurement ID by the measurement name. Parameters: toolName – Tool name measurementName – Measurement name Returns: -1 – measurement does not exist Other value – Measurement ID

Output Functions

Function	Description
void Output_Set (double value, int decision)	Sets the output value and decision on Output index 0. Only the last

Function	Description
	output value / decision in a script run is kept and passed to the Gocator output. To output an invalid value, the constant INVALID_VALUE can be used (e.g., Output_SetAt(0, INVALID_VALUE, 0)) Parameters: value - value output by the script decision - decision value output by the script. Can only be 0 or 1
void Output_SetAt(unsigned int index, double value, int decision)	Sets the output value and decision at the specified output index. To output an invalid value, the constant INVALID_VALUE can be used (e.g., Output_SetAt(0, INVALID_VALUE, 0)) Parameters: index – Script output index value – value output by the script decision – decision value output by the script. Can only be 0 or 1
void Output_SetId(int id, double value, int decision)	Sets the output value and decision at the specified script output ID. To output an invalid value, the constant INVALID_VALUE can be used (e.g., Output_SetId(0, INVALID_VALUE, 0)) Parameters: id – Script output ID

Memory Functions

Function	Description
void Memory_Set64s (int id, long long value)	Stores a 64-bit signed integer in persistent memory. Parameters: id - ID of the value value - Value to store
long long Memory_Get64s (int id)	Loads a 64-bit signed integer from persistent memory. Parameters: id - ID of the value Returns: value - Value stored in persistent memory
void Memory_Set64u (int id, unsigned long long value)	Stores a 64-bit unsigned integer in the persistent memory Parameters: id - ID of the value value - Value to store
unsigned long long Memory_Get64u (int id)	Loads a 64-bit unsigned integer from persistent memory. Parameters: id - ID of the value Returns:

Function	Description
	value - Value stored in persistent memory
void Memory_Set64f (int id, double value)	Stores a 64-bit double into persistent memory. Parameters: id - ID of the value value - Value to store
double Memory_Get64f (int id)	Loads a 64-bit double from persistent memory. All persistent memory values are set to 0 when the sensor starts. Parameters: id - ID of the value Returns: value - Value stored in persistent memory
int Memory_Exists (int id)	Tests for the existence of a value by ID. Parameters: id – Value ID Returns: 0 – value does not exist 1 – value exists
void Memory_Clear (int id)	Erases a value associated with an ID. Parameters: id – Value ID
void Memory_ClearAll()	Erases all values from persistent memory

Runtime Variable Functions

Function	Description
int RuntimeVariable_Count()	Returns the number of runtime variables that can be accessed. Returns: The count of runtime variables.
int RuntimeVariable_Get32s(int id)	Returns the value of the runtime variable at the given index. Parameters: Id – ID of the runtime variable Returns: Runtime variable value

Stamp Functions

Function	Description
long long Stamp_Frame()	Gets the frame number of the last frame.
long long Stamp_Time()	Gets the time stamp of the last frame.
long long Stamp_Encoder()	Gets the encoder position of the last frame when the image data was

Function	Description
	scanned/taken.
long long Stamp_EncoderZ()	Gets the encoder position at the time of the last index pulse of the last frame.
unsigned int Stamp_Inputs()	Gets the digital input state of the last frame. Returns a bit field representing digital input states.

Math Functions

Function	Description
float sqrt(float x)	Calculates square root of x
float sin(float x)	Calculates sin(x) (x in radians)
float cos(float x)	Calculates cos(x) (x in radians)
float tan(float x)	Calculates tan(x) (x in radians)
float asin(float x)	Calculates asin(x)
float acos(float x)	Calculates acos(x)
float atan(float x)	Calculates atan(x)
float pow (float x, float y)	Calculates the exponential value. x is the base, y is the exponent
float fabs(float x)	Calculates the absolute value of x

Example

The following example shows how to create a custom measurement that is based on the values from other measurements. The example calculates the 3-Dimensional Euclidean distance between the center of a hole and the center of the base of a nearby stud, then checks decision limits on this metric before sending the output.

```
/* Calculate the 3-Dimensional Euclidean distance between two points in 3D space*/

/* Retrieve 3D coordinate from Hole X, Y and Z tools (assumes these tools have been
configured as ID 0, ID 1 and ID 2 respectively) */
double HoleX = Measurement_Value(0);
double HoleY = Measurement_Value(1);
double HoleZ = Measurement_Value(2);

/* Retrieve 3D coordinate from StudBase X, Y and Z tools (assumes these tools have been
configured as ID 3, ID 4 and ID 5 respectively) */
double StudX = Measurement_Value(3);
double StudY = Measurement_Value(4);
double StudZ = Measurement_Value(5);

/* Calculate distance between points in 3D space */
double Distance = sqrt((HoleX - StudX)*(HoleX - StudX) + (HoleY - StudY)*(HoleY - StudY)
+ (HoleZ - StudZ)*(HoleZ - StudZ));
```

```
/* Min and Max Decision Limits */
/* Note that measurement values are in the unit of thousands of a millimeter in the
script */
/* In this example the distance is considered good if it's between 17.9 mm and 18.1 mm */
double MinDecisionLimit = 17.900;
double MaxDecisionLimit = 18.100;

if (Distance > MinDecisionLimit && Distance < MaxDecisionLimit)
{
   Output_Set(Distance, 1);
}
else
{
   Output_Set(Distance, 0);
}
```

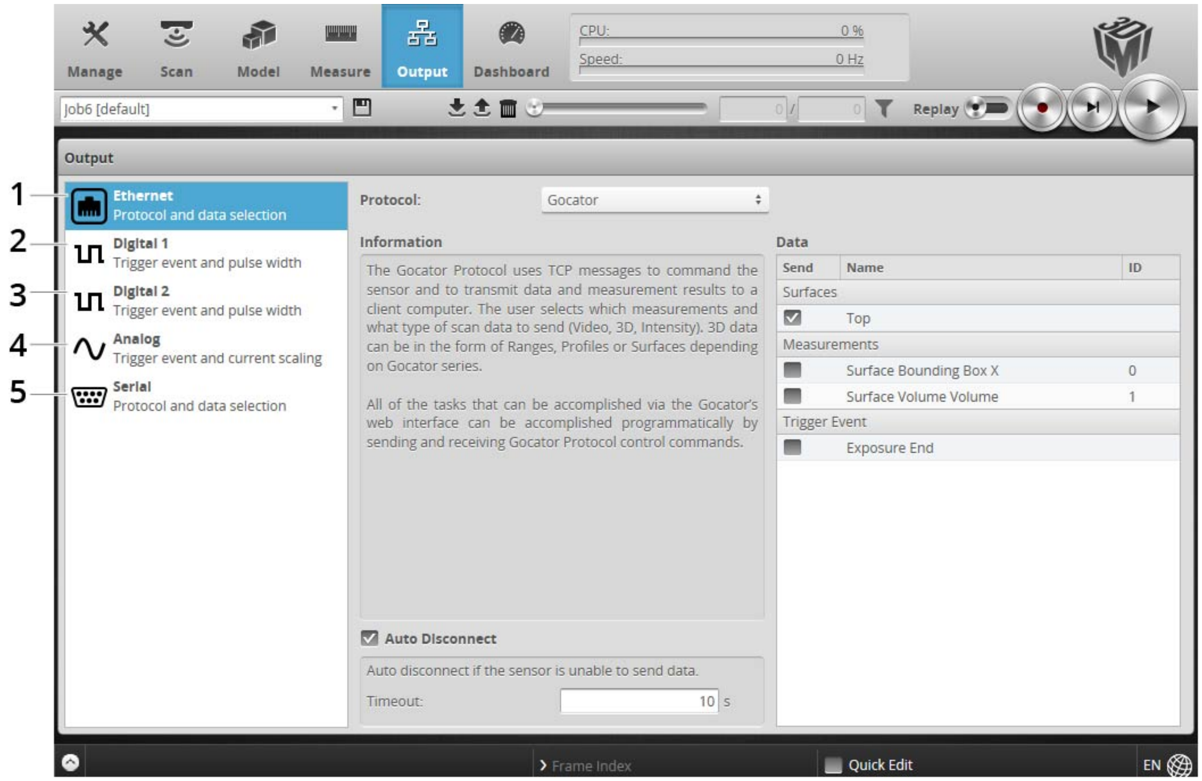
Output

The following sections describe the **Output** page.

Output Page Overview

Output configuration tasks are performed using the **Output** page. Gocator sensors can transmit data and measurement results to various external devices using several output interface options.

> Up to two outputs can have scheduling enabled with ASCII as the Serial output protocol. When Selcom is the current Serial output protocol, only one other output can have scheduling enabled.



	Category	Description
1	Ethernet	Used to select the data sources that will transmit data via Ethernet. See *Ethernet Output* on the next page.
2	Digital Output 1	Used to select the data sources that will be combined to produce a digital output pulse on Output 1. See *Digital Output* on page 516.
3	Digital Output 2	Used to select the data sources that will be combined to produce a digital output pulse on Output 2. See *Digital Output* on page 516.
4	Analog Panel	Used to convert a measurement value or decision into an analog output signal. See *Analog Output* on page 519.
5	Serial Panel	Used to select the measurements that will be transmitted via RS-485 serial output. See *Serial Output* on page 522.

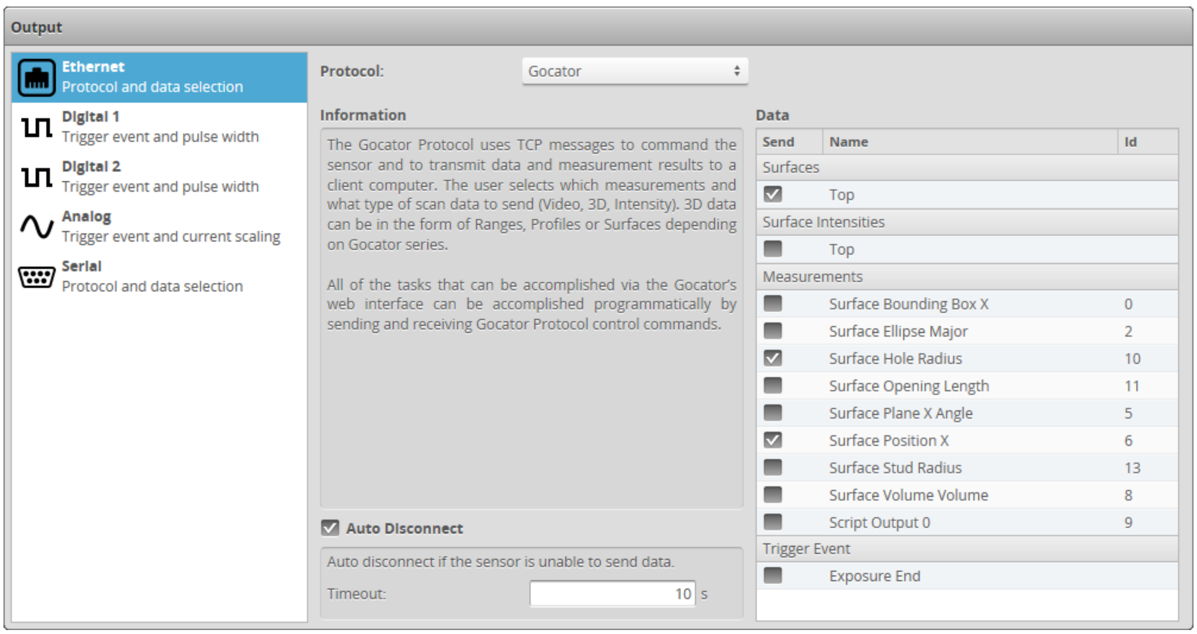
Ethernet Output

A sensor uses TCP messages (Gocator protocol) to receive commands from client computers, and to send video, 3D point clouds, intensity, and measurement results to client computers. The sensor can also receive commands from and send measurement results to a PLC using ASCII, Modbus TCP, PROFINET, or EtherNet/IP protocol.

See *Protocols* on page 650 for the specification of these protocols.

The specific protocols used with Ethernet output are selected and configured within the panel.

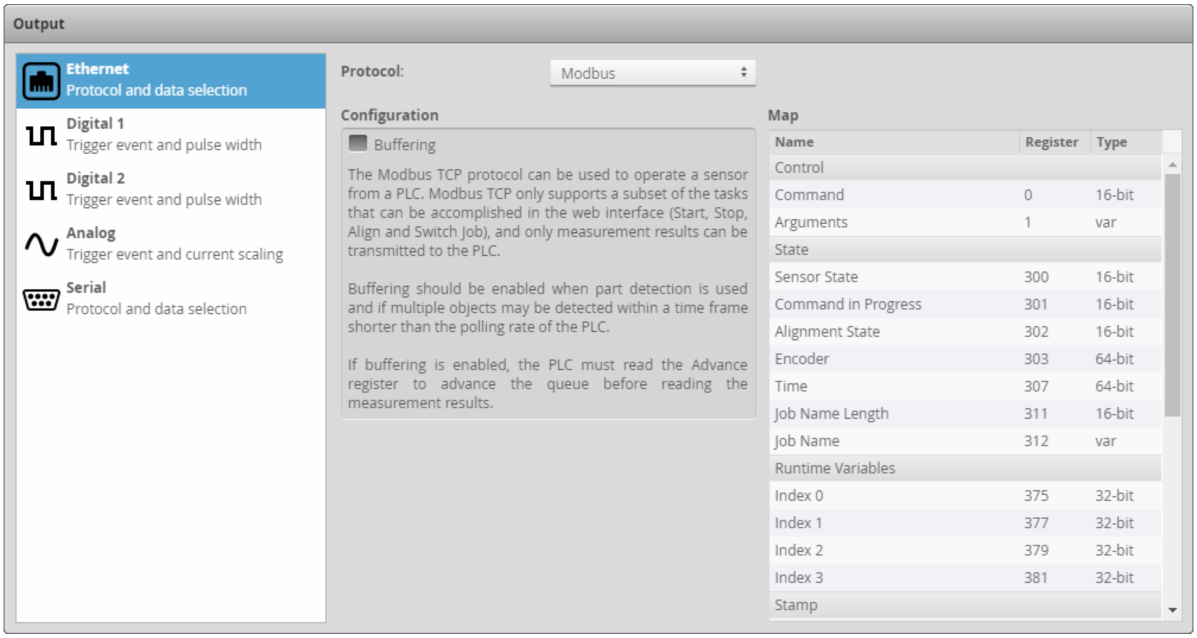
> The Gocator protocol is always on and its output is always available, regardless of the output you choose. This allows simultaneous connections via an SDK application and a PLC, letting you for example archive or display scan data on a PC while controlling equipment with a PLC.



To receive commands and send results using Gocator Protocol messages:

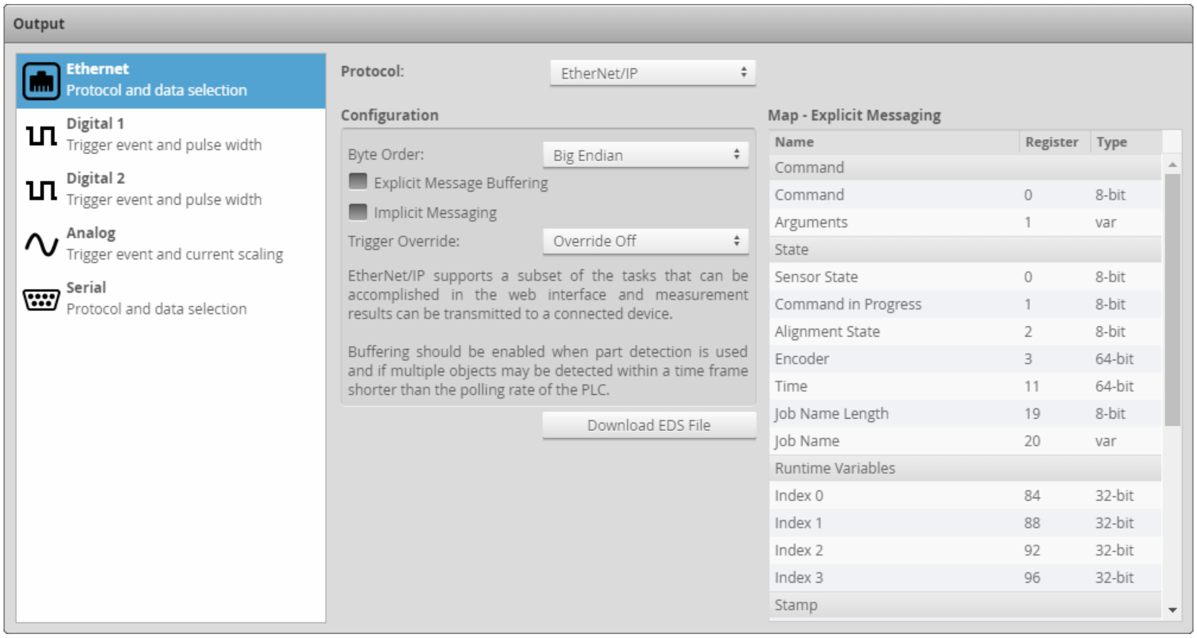
1. Go to the **Output** page.
2. Click on the **Ethernet** category in the **Output** panel.
3. Select **Gocator** as the protocol in the **Protocol** drop-down.
4. Check the video, data, intensity, or measurement items to send.
5. (Optional) Uncheck the Auto Disconnect setting.
   By default, this setting is checked, and the timeout is set to 10 seconds.

All of the tasks that can be accomplished with the Gocator's web interface (creating jobs, performing alignment, sending data and health information, and software triggering, etc.) can be accomplished programmatically by sending Gocator protocol control commands.



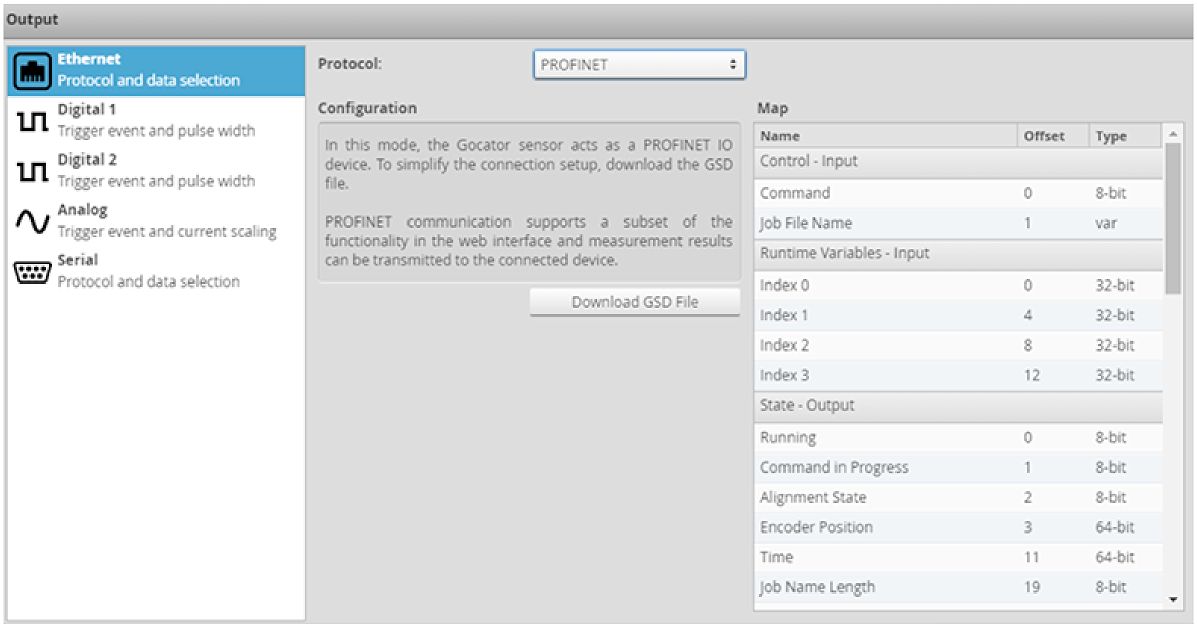
To receive commands and send results using Modbus TCP messages:

1. Go to the **Output** page.
2. Click on **Ethernet** in the **Output** panel.
3. Select **Modbus** as the protocol in the **Protocol** drop-down.
   Unlike the Gocator Protocol, you do not select which measurement items to output. The Ethernet panel will list the register addresses that are used for Modbus TCP communication.
   The Modbus TCP protocol can be used to operate a sensor. Modbus TCP only supports a subset of the tasks that can be performed in the web interface. A sensor can only process Modbus TCP commands when Modbus is selected in the **Protocol** drop-down.
4. Check the **Buffering** checkbox, if needed.
   Buffering is needed, for example, in Surface mode if multiple objects are detected within a time frame shorter than the polling rate of the PLC.
   If buffering is enabled with the Modbus protocol, the PLC must read the Advance register to advance the queue before reading the measurement results.



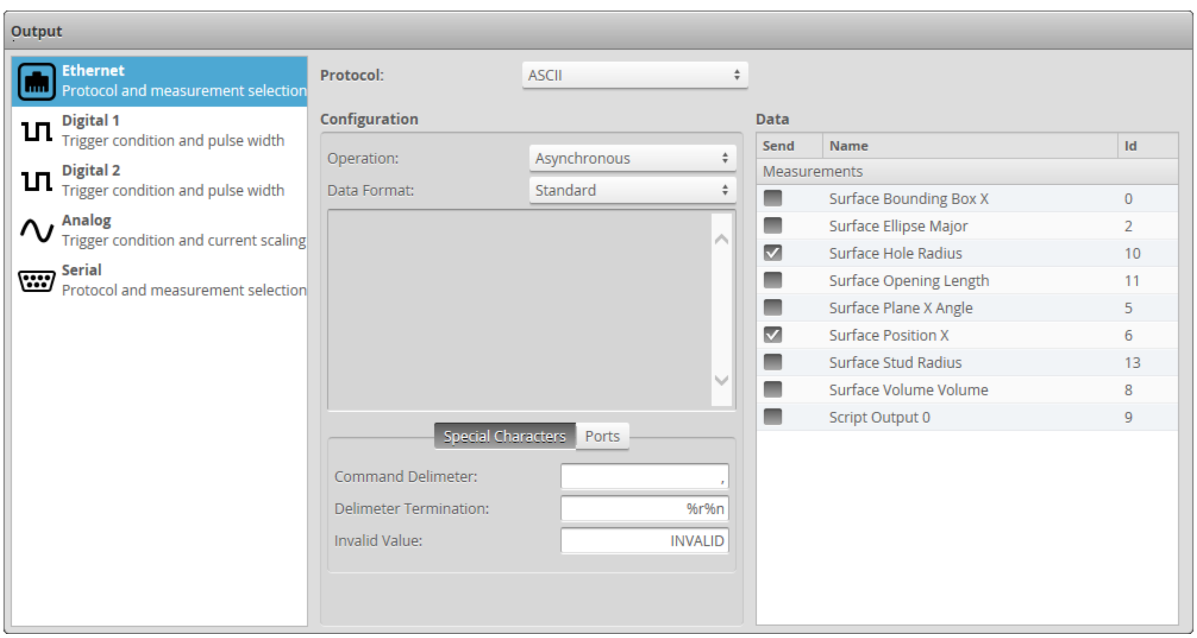
To receive commands and send results using EtherNet/IP messages:

1. Go to the **Output** page.
2. Click on **Ethernet** in the **Output** panel.
3. Select **EtherNet/IP** in the **Protocol** option.
   Unlike using the Gocator Protocol, you don't select which measurement items to output. The **Ethernet** panel will list the register addresses that are used for EtherNet/IP messages communication.
   The EtherNet/IP protocol can be used to operate a sensor. EtherNet/IP only supports a subset of the tasks that can be accomplished in the web interface. A sensor can only process EtherNet/IP commands when the EtherNet/IP is selected in the **Protocol** option.
4. Check the **Explicit Message Buffering** option, if needed.
   Buffering is needed, for example, in Surface mode if multiple objects are detected within a time frame shorter than the polling rate of the PLC. If buffering is enabled with the EtherNet/IP protocol, the buffer is automatically advanced when the Sample State Assembly Object is read (*Sample State Assembly* on page 726).
5. Check the **Implicit Messaging** option, if needed.
   Implicit messaging uses UDP and is faster than explicit messaging, so it is intended for time-critical applications. However, implicit messaging is layered on top of UDP. UDP is connectionless and data delivery is not guaranteed. For this reason, implicit messaging is only suitable for applications where occasional data loss is acceptable.
   For more information on setting up implicit messaging, see http://lmi3d.com/sites/default/files/APPNOTE_Implicit_Messaging_with_Allen-Bradley_PLCs.pdf.
6. Choose the byte order in the **Byte Order** dropdown.
7. Click the **Download EDS File** button to download an EDS file for use with your IDE.



To receive commands and send results using PROFINET messages:

1. Go to the **Output** page.
2. Click on **Ethernet** in the **Output** panel.
3. Select **PROFINET** in the **Protocol** option.
4. Click the **Download GSD File** button to download a GSD file for use with your IDE.



To receive commands and send results using ASCII messages:

1. Go to the **Output** page.

2. Click on **Ethernet** in the **Output** panel.
3. Select **ASCII** as the protocol in the **Protocol** drop-down.
4. Set the operation mode in the **Operation** drop-down.
   In asynchronous mode, the data results are transmitted when they are available. In polling mode, users send commands on the data channel to request the latest result. See *Polling Operation Commands (Ethernet Only)* on page 784 for an explanation of the operation modes.
5. Select the data format from the **Data Format** drop-down.
   Standard: The default result format of the ASCII protocol. Select the measurement to send by placing a check in the corresponding checkbox. See *Standard Result Format* on page 792 for an explanation of the standard result mode.
   Standard with Stamp: Select the measurement to send by placing a check in the corresponding checkbox. See *Standard Result Format* on page 792 for an explanation of the standard result mode.
   Custom: Enables the custom format editor. Use the replacement patterns listed in **Replacement Patterns** to create a custom format in the editor. C language *printf*-style formatting is also supported: for example, %sprintf[%09d, %value[0]]. This allows fixed length formatting for easier input parsing in PLC and robot controller logic.
6. Set the special characters in the **Special Characters** tab.
   Set the command delimiter, delimiter termination, and invalid value characters. Special characters are used in commands and standard-format data results.
7. Set the TCP ports in the **Ports** tab.
   Select the TCP ports for the control, data, and health channels. If the port numbers of two channels are the same, the messages for both channels are transmitted on the same port.

Digital Output

Gocator sensors can convert measurement decisions or software commands to digital output pulses, which can then be used to output to a PLC or to control external devices, such as indicator lights or air ejectors.

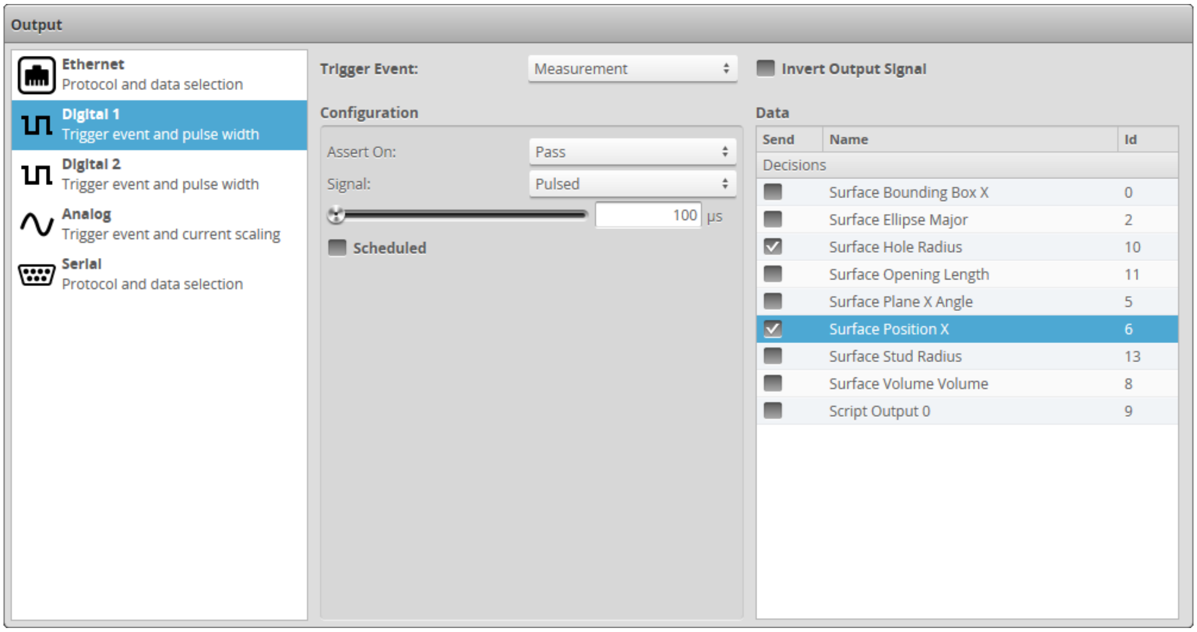
> Digital outputs cannot be used when taking scans using the Snapshot button, which takes a single scan and is typically used to test measurement tool settings. Digital outputs can only be used when a sensor is running, taking a continuous series of scans.

A digital output can act as a measurement valid signal to allow external devices to synchronize to the timing at which measurement results are output. In this mode, the sensor outputs a digital pulse when a measurement result is ready.

A digital output can also act as a strobe signal to allow external devices to synchronize to the timing at which the sensor exposes. In this mode, the sensor outputs a digital pulse when the sensor exposes.

Each sensor supports two digital output channels. See *Digital Outputs* on page 896 for information on wiring digital outputs to external devices.

Trigger conditions and pulse width are then configured within the panel.



To output measurement decisions:

1. Go to the **Output** page.
2. Click **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Measurement**.
4. In **Configuration**, set **Assert On** and select the measurements that should be combined to determine the output.
   If multiple measurement decisions are selected and **Assert On** is set to **Pass**, the output is activated when all selected measurements pass.
   If **Assert On** is set to **Fail**, the output is activated when any one of the selected measurements fails.
5. Set the **Signal** option.
   The signal type specifies whether the digital output is a continuous signal or a pulsed signal. If **Signal** is set to **Continuous**, the signal state is maintained until the next transition occurs. If **Signal** is set to is **Pulsed**, you must specify the pulse width and how it is scheduled.
6. Specify a pulse width using the slider.
   The pulse width is the duration of the digital output pulse, in microseconds.
7. Check the **Scheduled** option if the output needs to be scheduled; otherwise, leave it unchecked for immediate output.
   A scheduled output becomes active after the delay from the start of Gocator exposure. A scheduled output can be used to track the decisions for multiple objects as these objects travel from the sensor to the eject gates.
   The **Delay** setting specifies the distance from the sensor to the eject gates.
   An immediate output becomes active as soon as measurement results are available. The output activates after the sensor finishes processing the data. As a result, the time between the start of sensor

exposure and output activates can vary and is dependent on the processing latency. The latency is reported in the dashboard and in the health messages.

8. If you checked **Scheduled**, specify a delay and a delay domain.
   The **Delay** specifies the time or encoder distance between the start of sensor exposure and when the output becomes active. The delay should be larger than the time needed to process the data inside the sensor. It should be set to a value that is larger than the processing latency reported in the dashboard or in the health messages.
   The unit of the delay is configured with the **Delay Domain** setting.
9. If you want to invert the output signal, check **Invert Output Signal**.

To output a measurement valid signal:

1. Go to the **Output** page.
2. Click on **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Measurement**.
4. In **Configuration**, set **Assert On** to **Always**.
5. Select the measurements.
   The output activates when the selected decisions produce results. The output activates only once for each frame even if multiple decision sources are selected.
6. Specify a pulse width using the slider.
   The pulse width determines the duration of the digital output pulse, in microseconds.

To respond to software scheduled commands:

1. Go to the **Output** page.
2. Click **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Software**.
4. Specify a **Signal** type.
   The signal type specifies whether the digital output is a continuous signal or a pulsed signal. If the signal is continuous, its state is maintained until the next transition occurs. If the signal is pulsed, user specifies the pulse width and the delay.
5. Specify a **Pulse Width**.
   The pulse width determines the duration of the digital output pulse, in microseconds.
6. Specify if the output is immediate or scheduled.
   A pulsed signal can become active immediately or be scheduled. A continuous signal always becomes active immediately.
   Immediate output becomes active as soon as a scheduled digital output (*Schedule Digital Output* on page 676) is received.
   Scheduled output becomes active at a specific target time or position, given by the Scheduled Digital Output command. Commands that schedule an event in the past will be ignored. An encoder value is in

the future if the value will be reached by moving in the forward direction (the direction that encoder calibration was performed in).

To output an exposure signal:

1. Go to the **Output** page.
2. Click **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Exposure Begin** or **Exposure End**.
4. Set the **Pulse Width** option.
   The pulse width determines the duration of the digital output pulse, in microseconds.

To output an alignment signal:

1. Go to the **Output** page.
2. Click **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Alignment**.
   The digital output state is High if the sensor is aligned, and Low if not aligned. Whether the sensor is running does not affect the output.

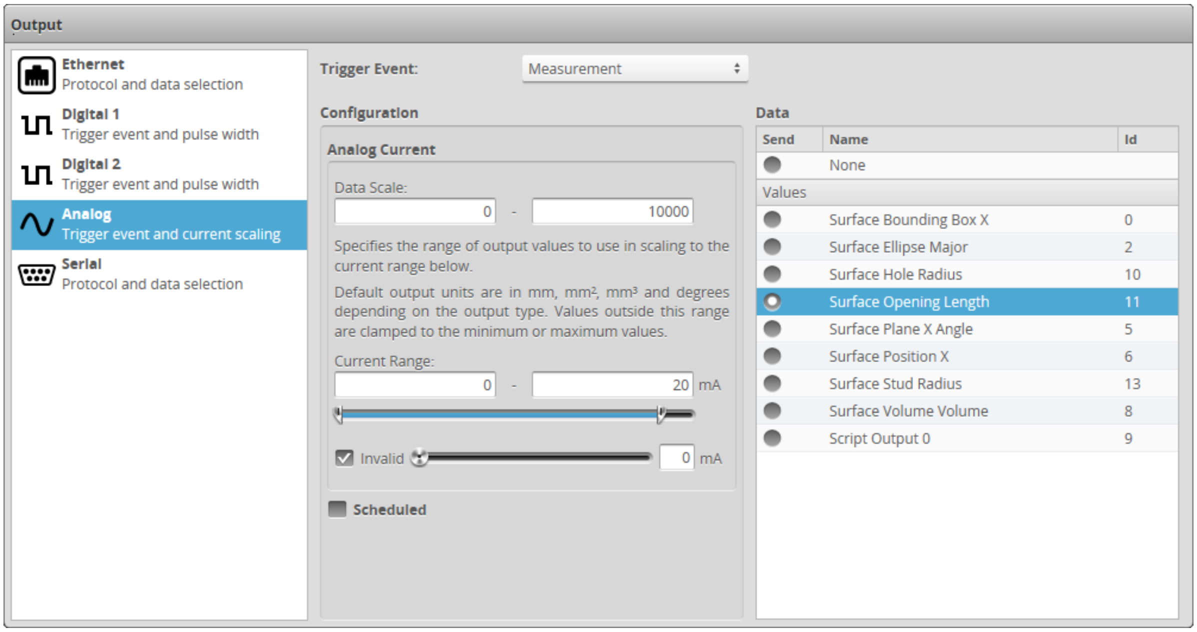
To respond to exposure begin/end:

1. Go to the **Output** page.
2. Click **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Exposure Begin** or **Exposure End**.

Analog Output

Gocator sensors can convert a measurement result or software request to an analog output.Each sensor supports one analog output channel.

For information on wiring analog output to an external device, see *Analog Output* on page 898



To output measurement value or decision:

1. Go to the **Output** page.
2. Click on **Analog** in the **Output** panel.
3. Set **Trigger Event** to **Measurement**.
4. Select the measurement that should be used for output.
   Only one measurement can be used for analog output. Measurements shown here correspond to measurements that have been programmed using the **Measurements** page.
5. Specify **Data Scale** values.
   The values specified here determine how measurement values are scaled to the minimum and maximum current output. The **Data Scale** values are specified in millimeters for dimensional measurements such as distance, square millimeters for areas, cubic millimeters for volumes, and degrees for angle results.
6. Specify **Current Range** and **Invalid** current values.
   The values specified here determine the minimum and maximum current values in milliamperes. If **Invalid** is checked, the current value specified with the slider is used when a measurement value is not valid. If **Invalid** is not checked, the output holds the last value when a measurement value is not valid.
7. Specify if the output is immediate or scheduled.
   An analog output can become active immediately or scheduled. Check the **Scheduled** option if the output needs to be scheduled.
   A scheduled output becomes active after a specified delay from the start of Gocator exposure. A scheduled output can be used to track the decisions for multiple objects as these objects travel from the sensor to the eject gates. The delay specifies the distance from the sensor to the eject gates.
   An Immediate output becomes active as soon as the measurement results are available. The output

activates after the Gocator finishes processing the data. As a result, the time between the start of Gocator exposure and output activates depends on the processing latency. The latency is reported in the dashboard and in the health messages.

8. Specify a delay.
   The delay specifies the time or spatial location between the start of Gocator exposure and the output becomes active. The delay should be larger than the time needed to process the data inside the Gocator. It should be set to a value that is larger than the processing latency reported in the dashboard and in the health messages.
   The unit of the delay is configured in the trigger panel. See *Triggers* on page 89 for details.

1. Go to the **Output** page.
2. Click on **Analog** in the **Output** panel.
3. Set **Trigger Event** to **Measurement**.
4. Select the measurement that should be used for output.
   Only one measurement can be used for analog output. Measurements shown here correspond to measurements that have been programmed using the **Measurements** page.
5. Specify **Data Scale** values.
   The values specified here determine how measurement values are scaled to the minimum and maximum current output. The **Data Scale** values are specified in millimeters for dimensional measurements such as distance, square millimeters for areas, cubic millimeters for volumes, and degrees for angle results.
6. Specify **Current Range** and **Invalid** current values.
   The values specified here determine the minimum and maximum current values in milliamperes. If **Invalid** is checked, the current value specified with the slider is used when a measurement value is not valid. If **Invalid** is not checked, the output holds the last value when a measurement value is not valid.
7. Specify if the output is immediate or scheduled.
   An analog output can become active immediately or scheduled. Check the **Scheduled** option if the output needs to be scheduled.
   A scheduled output becomes active after a specified delay from the start of Gocator exposure. A scheduled output can be used to track the decisions for multiple objects as these objects travel from the sensor to the eject gates. The delay specifies the distance from the sensor to the eject gates.
   An Immediate output becomes active as soon as the measurement results are available. The output activates after the Gocator finishes processing the data. As a result, the time between the start of Gocator exposure and output activates depends on the processing latency. The latency is reported in the dashboard and in the health messages.
8. Specify a delay.
   The delay specifies the time or spatial location between the start of Gocator exposure and the output becomes active. The delay should be larger than the time needed to process the data inside the Gocator. It should be set to a value that is larger than the processing latency reported in the dashboard and in the health messages.
   The unit of the delay is configured in the trigger panel. See *Triggers* on page 89 for details.

	The analog output takes about 75 us to reach 90% of the target value for a maximum change, then another ~40 us to settle completely.

To respond to software scheduled commands:

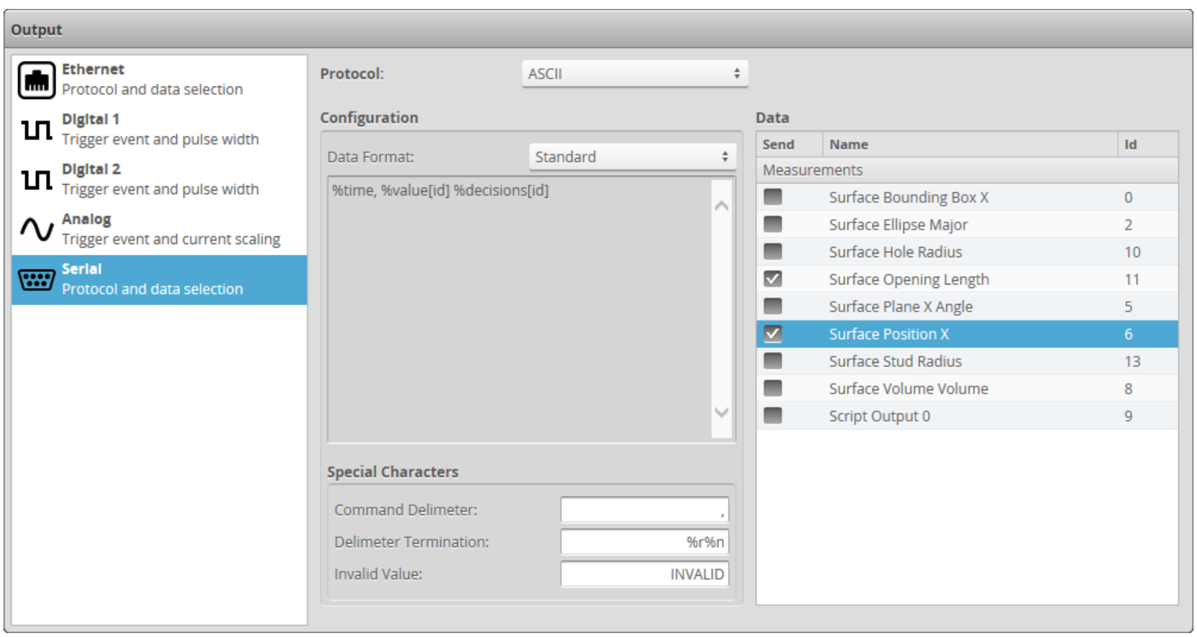
1. Go to the **Output** page.
2. Click on **Analog** in the **Output** panel.
3. Set **Trigger Event** to **Software**.
4. Specify if the output is immediate or scheduled.
   An analog output value becomes active immediately or scheduled. Immediate output becomes active as soon as a Scheduled Analog Output command (see *Schedule Analog Output* on page 677) is received. Software scheduled command can schedule an analog value to output at a specified future time or encoder value, or changes its state immediately. The Delay setting in the panel is ignored. Commands that schedule event in the past will be ignored. An encoder value is in future if the value will be reached by moving in the forward direction (the direction that encoder calibration was performed in).

Serial Output

Gocator's web interface can be used to select measurements to be transmitted via RS-485 serial output. Each sensor has one serial output channel.

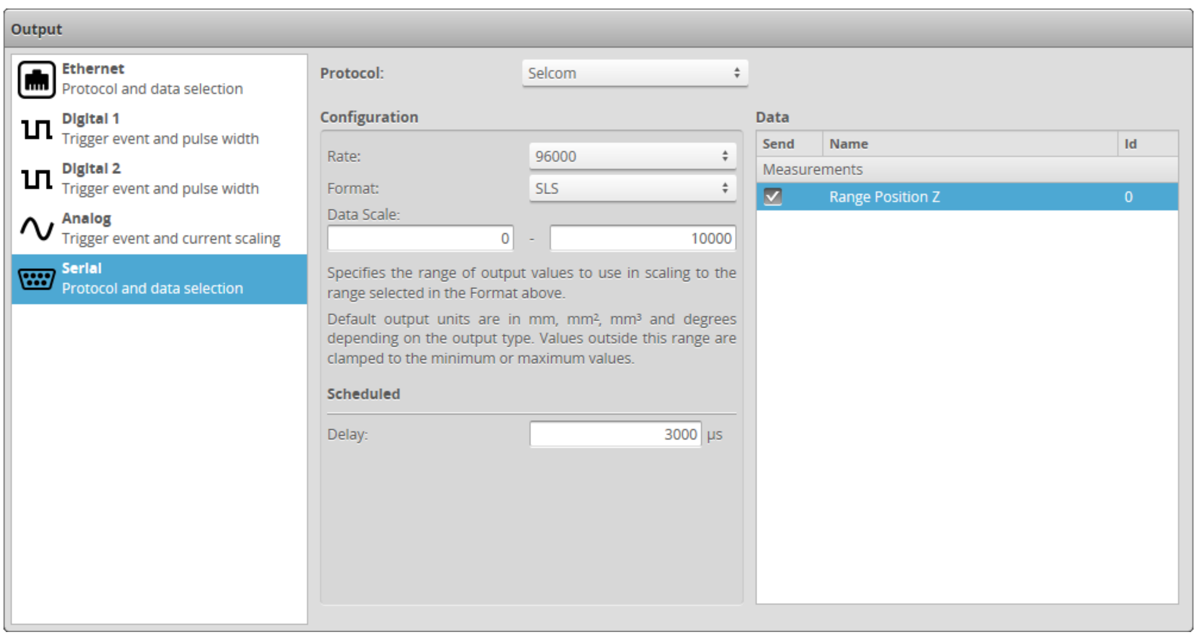
The ASCII protocol outputs data asynchronously using a single serial port. For information on the ASCII Protocol parameters and data formats, see *ASCII Protocol* on page 783.

For information on wiring serial output to an external device, see *Serial Output* on page 898.



To configure ASCII output:

1. Go to the **Output** page.
2. Click on **Serial** in the **Output** panel.
3. Select **ASCII** in the **Protocol** option.
4. Select the **Data Format**.
   Select **Standard** to use the default result format of the ASCII protocol. Select value and decision to send by placing a check in the corresponding check box. See *Standard Result Format* on page 792 for an explanation of the standard result mode.
   Select **Custom** to customize the output result. A data format box will appear in which you can type the format string. See *Custom Result Format* on page 792 for the supported format string syntax.
5. Select the measurments to send.
   Select measurements by placing a check in the corresponding check box.
6. Set the **Special Characters**.
   Select the delimiter, termination and invalid value characters. Special characters are used in commands and standard-format data results.



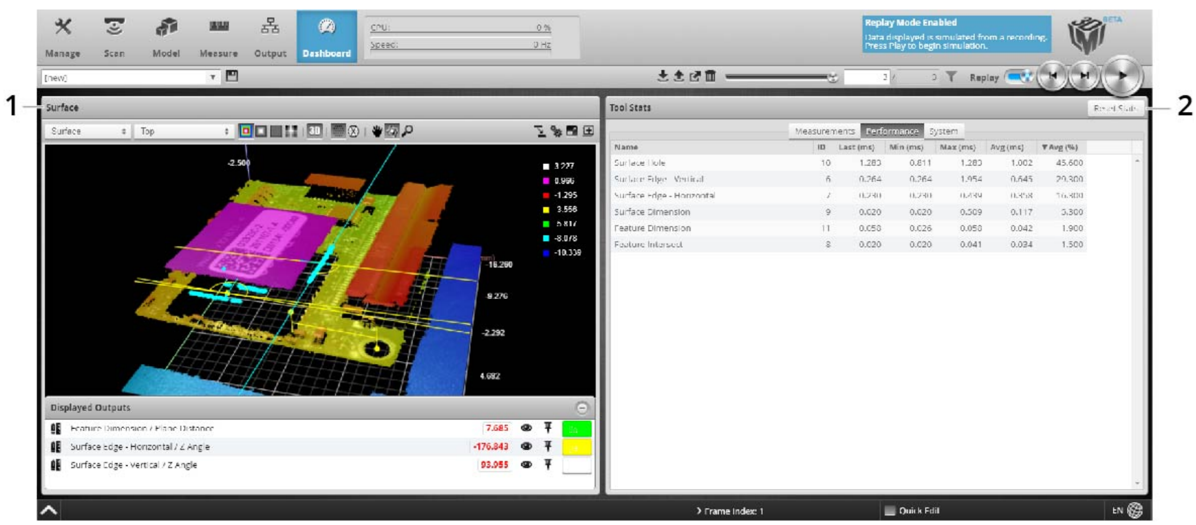
Dashboard

The following sections describe the **Dashboard** page.

Dashboard Page Overview

The **Dashboard** page summarizes sensor and system health information and provides tool and measurement statistics. It also provides tool performance statistics. The data viewer is available on the Dashboard page. This is especially useful for users accessing sensors via Technician accounts (which provide a simplified user interface, namely, with only the Scan and Dashboard pages). This lets any user monitor one or more measurements visually, on scan data, during troubleshooting and monitoring.

You can also pin multiple tool outputs such as measurements and geometric features so that they are displayed on the data viewer at the same time. Note however that pinned outputs in the data viewers on the Measure page and the Dashboard page are *not* independent: pinning or unpinning on either page affects the pinned outputs in both.



	Element	Description
1	Data viewer	Displays scan data and, if they are pinned to the main view, measurements and geometric features. For general information on the data viewer, see *Data Viewer* on page 115. For more information on pinning, see *Pinning Measurements and Features* on page 198.
2	Tool Stats	Displays measurement and tool performance statistics. See *Statistics* on the next page. Also displays sensor state and health information. See *State and Health Information* on page 526.

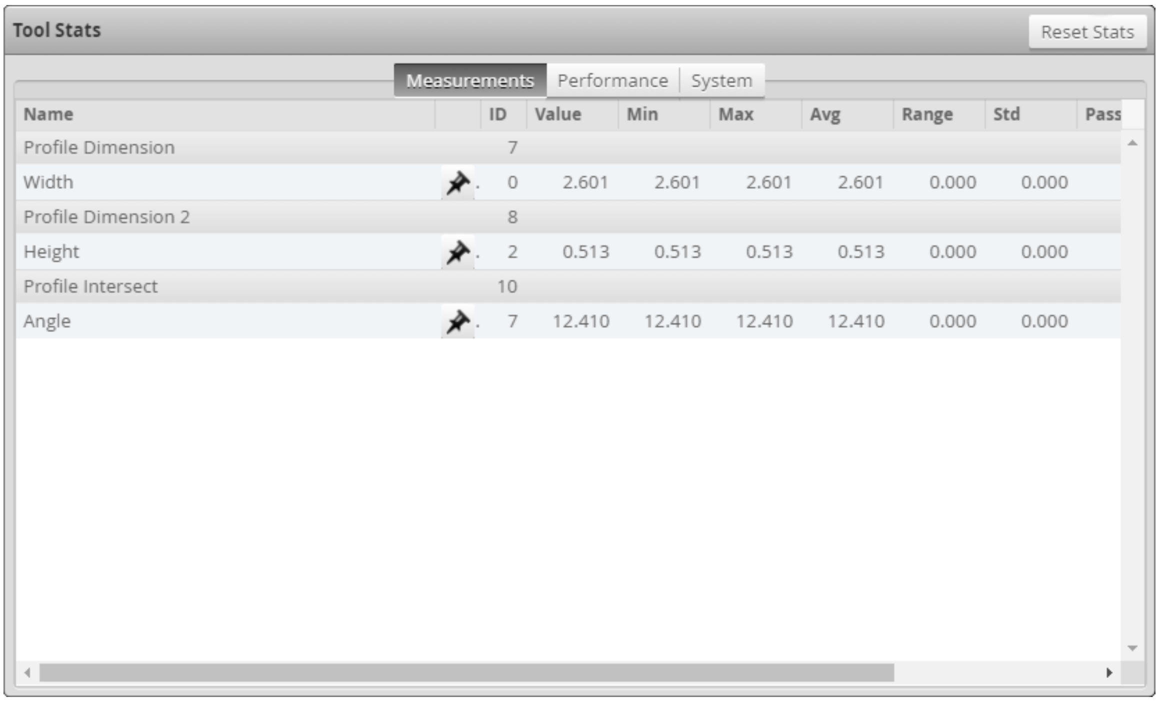
Statistics

In the **Tool Stats** panel, you can examine measurement and tool statistics in two tabs: **Measurements** and **Performance**.

To reset statistics in both tabs, use the **Reset Stats** button.

Measurements

The **Measurements** tab displays statistics for each measurement enabled in the **Measure** page, grouped by the tool that contains the measurement.



For each measurement, Gocator displays the following information:

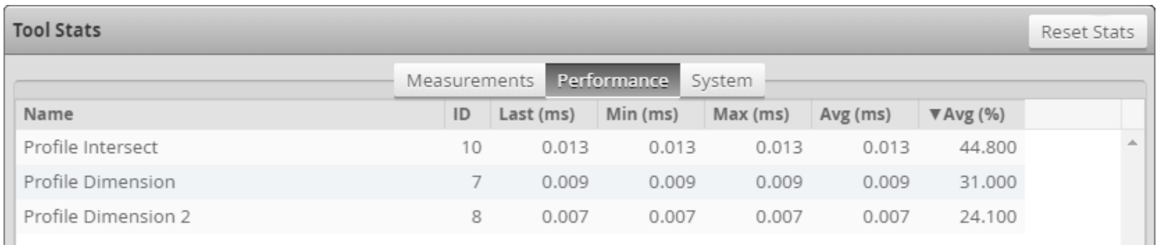
Measurement Statistics

Name	Description
ID	The measurement ID as set in the measurement's ID field on the Measure page.
Value	The most recent measurement value.
Min	The minimum measurement value that has been observed.
Max	The maximum measurement value that has been observed.
Avg	The average of all measurement values collected since the sensor was started.
Range	The difference between Max and Min.
Std	The standard deviation of all measurement values collected since the sensor was started.
Pass	The number of pass decisions the measurement has generated.

Name	Description
Fail	The number of fail decisions the measurement has generated.
Invalid	The number of frames that returned no valid measurement value.
Overflow	The number of frames that returned an overflow.

Performance

The **Performance** tab displays performance statistics (execution time) for each tool added in the **Measure** page.



For each tool, Gocator displays the following information:

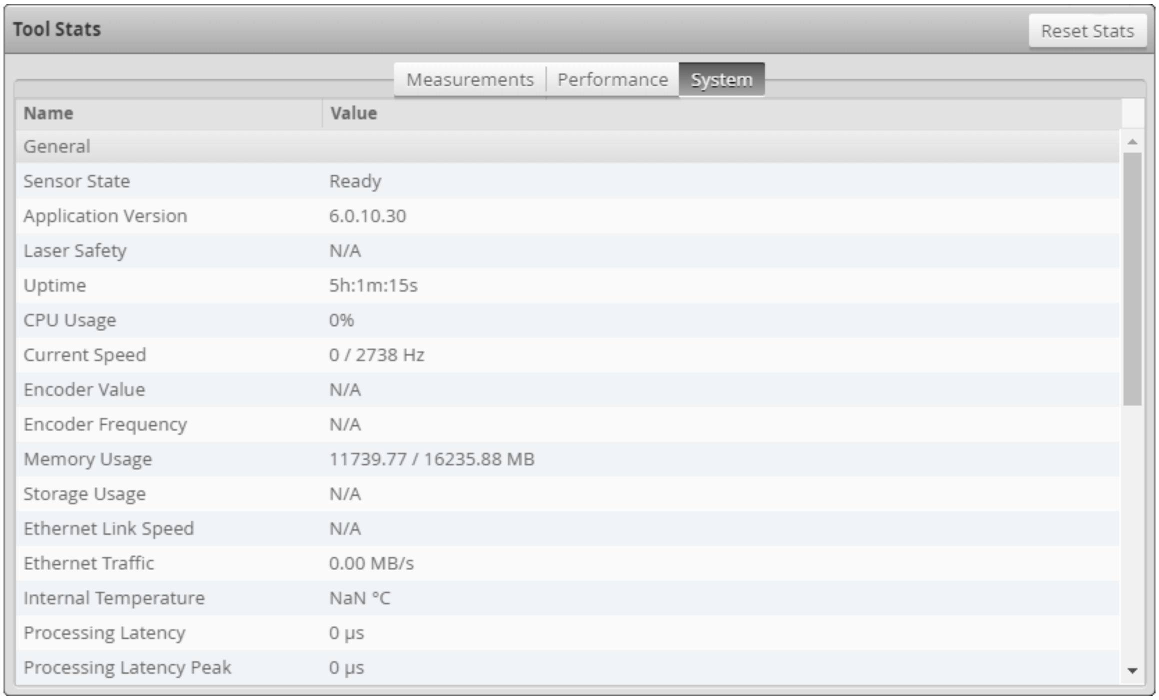
Performance Statistics

Name	Description
Last (ms)	The last execution time of the tool.
Min (ms)	The minimum execution time of the tool.
Max (ms)	The maximum execution time of the tool.
Avg (ms)	The average execution time of the tool.
Avg (%)	The average percentage the CPU the tool uses.

	Tools are sorted by the *Avg (%)* column in descending order.

State and Health Information

In the **Tool Stats** pane, you can examine state and health information.



The following information is available in the **System** tab on the **Dashboard** page:

Dashboard General System Values

Name	Description
Sensor State*	Current sensor state (Conflict, Ready, or Running).
Application Version	Sensor firmware version.
Laser Safety	Whether Safety is enabled. With snapshot sensors, enabling safety is not required in order to scan.
Uptime	Length of time since the sensor was power-cycled or reset.
CPU Usage	Sensor CPU utilization.
Current Speed*	Current speed of the sensor.
Encoder Value	Current encoder value (ticks).
Encoder Frequency	Current encoder frequency (Hz).
Memory Usage	Sensor memory utilization (MB used / MB total available).
Storage Usage	Sensor flash storage utilization (MB used / MB total available).
Ethernet Link Speed	Speed of the Ethernet link (Mbps).
Ethernet Traffic	Network output utilization (MB/sec).
Internal Temperature	Internal sensor temperature.
Processing Latency	Last delay from camera exposure start to when the results are ready for output.
Processing Latency Peak	Peak delay from camera exposure start to when the results are ready for output.
Alignment State	Whether the sensor or sensor system has been aligned.
Over Temperature State	Whether the internal temperature of the sensor is over a predetermined level.

Name	Description
Over Temperature Duration	The amount of time that the internal temperature of the sensor has been over a predetermined level.

Dashboard History Values

Name	Description
Scan Count*	Number of scans performed since sensor state last changed to Running.
Trigger Drop**	Count of camera frames dropped due to excessive trigger speed.
Processing Drop**	The sum of various indicators related to processing drop including drops due to insufficient CPU and buffer overflows.
Ethernet Output Drop**	Count of frame drops due to slow Ethernet link.
Analog Output Drop**	Count of analog output drops because last output has not been completed.
Serial Output Drop**	Count of serial output drops because last output has not been completed.
Digital Output 1 Drop**	Count of digital output drops because last output has not been completed.
Digital Output 2 Drop**	Count of digital output drops because last output has not been completed.
Digital Output 1 High Count	Count of high states on digital output.
Digital Output 2 High Count	Count of high states on digital output.
Digital Output 1 Low Count	Count of low states on digital output.
Digital Output 2 Low Count	Count of low states on digital output.
Anchor Invalid Count**	Count of invalid anchors.
Valid Spot Count	Count of valid spots detected in the last frame.
Max Spot Count*	Maximum number of spots detected since sensor was started.
Camera Search Count	Not applicable to these sensors.

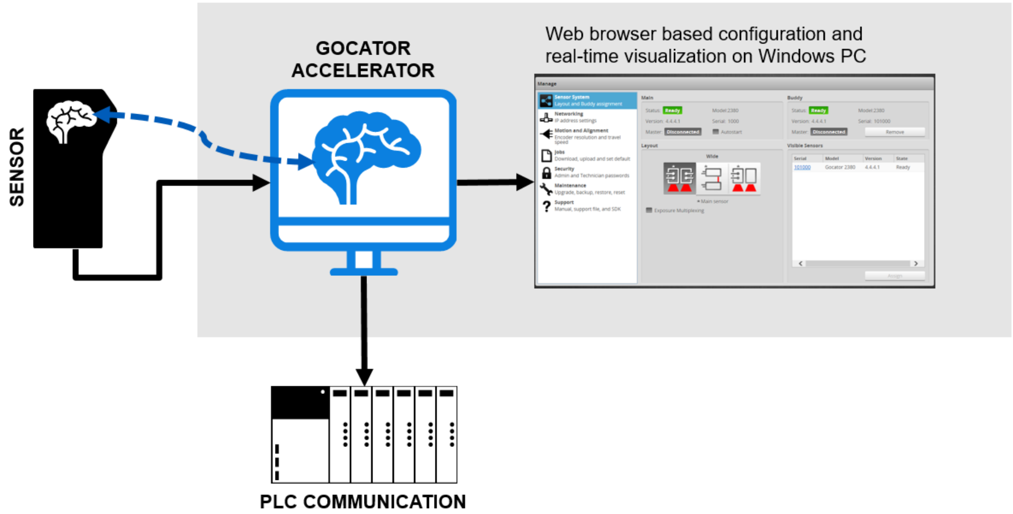
* When the sensor is accelerated, the indicator's value is reported from the accelerating PC.

** When the sensor is accelerated, the indicator's value is the sum of the values reported from the sensor and the accelerating PC.

Gocator Acceleration

Gocator sensors are all-in-one devices, combining scanning, measurement, and control capabilities in a single housing. However, to achieve higher scan rates and measurement performance in very high density data scenarios, you may wish to use one of two acceleration methods.

For information on the ports acceleration uses (for example, in order to ensure ports are not blocked over your network), see *Required Ports* on page 39.



Acceleration improves a sensor system's processing capability by transferring the processing to a dedicated processing device in the system. The accelerator can accelerate one or more standalone sensors or multi-sensor systems. LMI provides two acceleration solutions:

- A hardware Smart Vision Accelerator called GoMax
- PC-based acceleration software (available either as a standalone utility or via the SDK)

For estimated performance and scan rates, see *Estimated Performance and Scan Rates* on page 535.

	The Gocator emulator and accelerator (software and GoMax) do not support the PROFINET protocol.

The web interface of an accelerated sensor is identical to the interface of an unaccelerated sensor. The Ethernet-based output protocols (Gocator, EtherNet/IP, ASCII, and Modbus) are also identical to those found on an unaccelerated sensor, and are fully supported.

	Accelerators support digital, analog, and serial output from sensors. However, because output must be passed to the accelerator and then back to the sensor, network latency will have an impact on performance.

When a sensor is accelerated, it sends data directly to the accelerating device. You access the web interface using the IP address of the accelerating device, rather than the IP of the sensor. SDK applications can interface to the accelerator in the same way as is possible with a physical sensor, although the IP of the accelerating device must be used for the connection.

Benefits

Accelerated sensors provide several benefits.

Acceleration is completely transparent: because the output protocols of an accelerated sensor are identical to those of an unaccelerated sensor, SDK and PLC applications require no changes whatsoever for controlling accelerated sensors and receiving health information and data.

Measurement latency is reduced on accelerated sensors, which results in shorter cycle times. This means a sensor can scan more targets in a given time period.

The memory of accelerated sensors is limited only by the memory of the accelerating device. Accelerated sensors can therefore handle large 3D point clouds more effectively.

Dashboard and Health Indicators

After a sensor is accelerated, the values of some health indicators come from the accelerating PC instead of the sensor. Others come from a combination of the accelerated sensor and the accelerating PC.

- For information on which indicators are affected in the Dashboard in the web interface, see *State and Health Information* on page 526.
- For information on which indicators accessed through the Gocator protocol are affected, see *Health Results* on page 708.

Hardware Acceleration: GoMax

The GoMax Smart Vision Accelerator is a dedicated, small form factor device that can accelerate one or more sensors. Using GoMax to accelerate a sensor system rather than a PC greatly simplifies implementation and maintenance, providing a plug-and-play experience. And GoMax better handles continuous 3D data streams over Ethernet. Finally, GoMax automatically recovers from temporary power losses or system disconnects.

For more information on GoMax, see the product's user manual.

Software-Based Acceleration

You can implement acceleration capabilities in client applications that you create using the Gocator SDK. You can also use the provided standalone utility (GoAccelerator.exe) that you can use to instantly accelerate systems.

> The firmware version of the sensor you want to accelerate must match the version of the SDK used to build an accelerator-based application (or the version of the GoAccelerator utility).

System Requirements and Recommendations

Minimum System Requirements

The following are the minimum system requirements for accelerating a single sensor with the accelerator PC application:

PC

- Processor: Intel Core i3 or equivalent (32- or 64-bit)
- RAM: 4 GB
- Hard drive: 128 GB
- Operating system: Windows 7 or higher (32- or 64-bit)

To accelerate more sensors or run the system at higher speeds, use a computer with greater system resources.

Graphics Card

For additional acceleration of snapshot sensors, the accelerator can make use of a graphics card with CUDA support. However, the following requirements must be met:

- A CUDA compatible graphics card.
- A graphics card with a "Compute Capability" greater than 5.2. To determine whether a card fulfills this requirement, see NVIDIA's *CUDA GPUs* page.
- For Gocator 6.0, the minimum NVIDIA graphics card driver version is 441.22. For Gocator 5.3, the minimum driver version is 411.31. LMI recommends downloading the driver directly from the NVIDIA website to get the latest driver, rather than using the Windows Device Manager.

If PC acceleration performance is substantially lower when using Gocator 6.0, verify that cuda is enabled for the PC acceleration. If it is not, then try upgrading the graphics card.

Recommendations

The following are general recommendations:

- Purchase a PC based on the hardware specifications described in *Estimated Performance and Scan Rates* on page 535.
- Run only the accelerator application on the PC: third-party applications can consume system resources in unpredictable ways and at random times.
- Limit background Windows processes such as drive optimization (defragmentation) or virus scans, or schedule them so that they don't interfere with scanning sessions.
- Ensure that sufficient overhead in the system's resources is available. You can review the PC's resources with the Windows Task Manager and Resource Monitor applications. We recommend that you leave at least 20% network bandwidth, CPU, memory and disk utilization at all times.
- To verify system stability and robustness, perform long-term testing over multiple days.

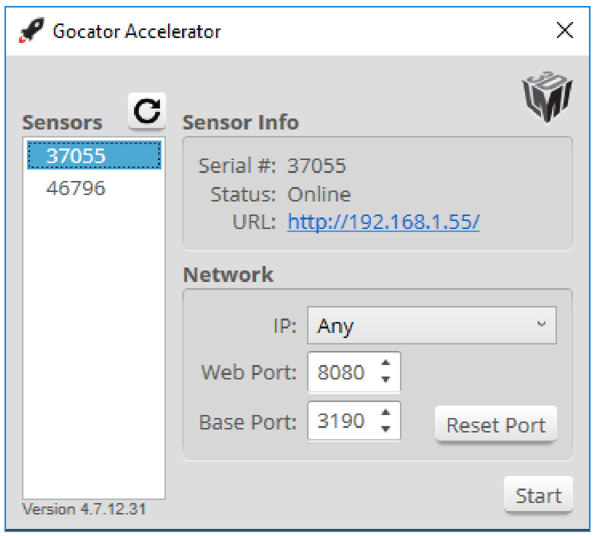
Installation

To get the necessary packages, go to https://downloads.lmi3d.com/, choose your product from the Product Downloads section, and download it from the Download Center.

- For the GoAccelerator utility, download the 14405-x.x.x.x_SOFTWARE_GO_Utilities.zip package.
- For the SDK libraries and DLL for integrating acceleration into a client application, download the 14400-X.X.X.X_SOFTWARE_GO_SDK.zip.

Gocator Accelerator Utility

The Accelerator utility accelerates the standalone sensors or multi-sensor systems you choose.

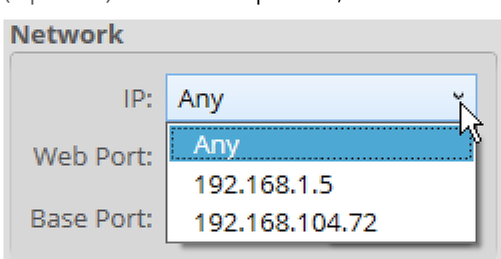


To accelerate a sensor using the Accelerator utility:

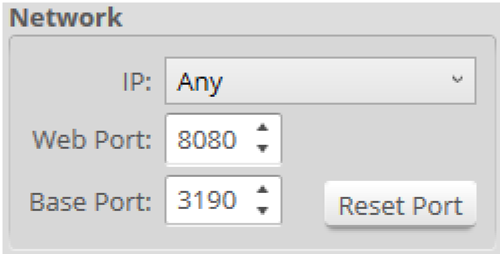
1. Power up the sensor system you want to accelerate.
2. Launch the Accelerator utility.
3. If a Windows Security alert asks whether you want to allow GoAccelerator.exe to communicate on networks, make sure **Public** and **Private** are checked, and then click **Allow Access**.
4. In the **Sensors** list, click the sensor you want to accelerate.

   If you do not see the sensor, you may need to wait a few seconds and then click the Refresh button ().

   In multi-sensor systems, only the Main sensor is listed.
5. (Optional) In the **IP** drop-down, choose an IP or choose **Any** to let the application choose.



6. (Optional) Set **Web Port** to a port for use with the accelerated sensor's URL.



> If port 8080 is already in use, set **Web Port** to an unused port.

7. (Optional) If you are accelerating multiple systems, click on another sensor in the **Sensors** list, and repeat the steps above.

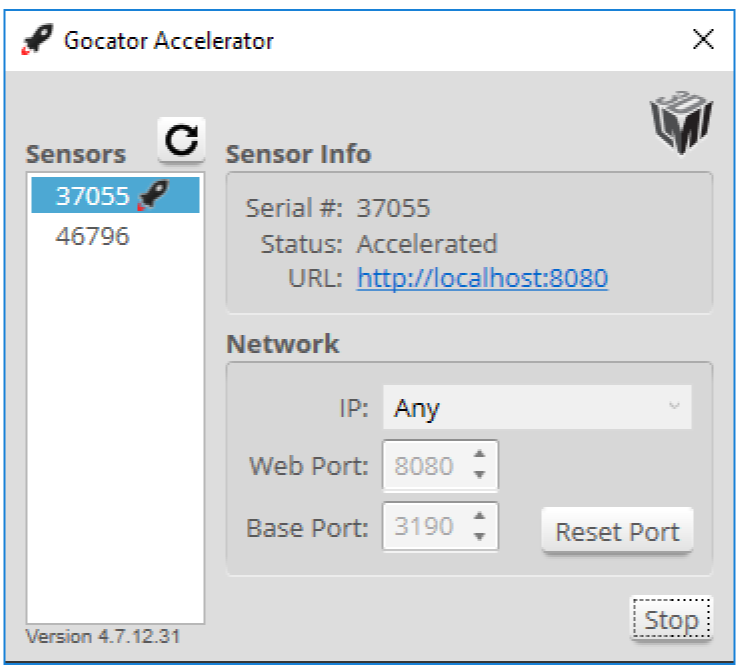
   The application uses **Base Port** as an offset for several communication port numbers.

   To avoid port conflicts, you should increment the base port number by at least 10 for each accelerated sensor.

   Port 3190 is the default base port number, allowing connections from SDK-based applications and the web UI without manually specifying ports.

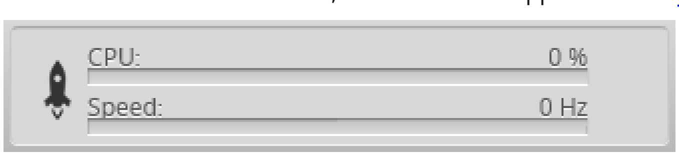
8. Click **Start**.

   The sensor system is now accelerated. An icon appears next to the accelerated sensor in the **Sensors** list to indicate this.



9. To open the accelerated sensor's web interface, in the Accelerator application, click the link next to **URL**.

   When a sensor is accelerated, a "rocket" icon appears in the metrics area.



If you restart an accelerated sensor, the sensor will continue to be accelerated when it restarts.

To stop an accelerated sensor in the Accelerator application:

1. Select the sensor in the **Sensors** list.
2. Click **Stop**.

To exit the Accelerator application:

1. Right-click the icon Accelerator icon () in the notification tray.

   Clicking the X icon in the application only minimizes the application.

2. Choose **Exit**.

SDK Application Integration

Sensor acceleration can be fully integrated into an SDK application. Users simply need to instantiate the GoAccelerator object and connect it to a sensor object.

```
GoAccelerator accelerator = kNULL;

// obtain GoSensor object by sensor IP address
if ((status = GoSystem_FindSensorByIpAddress(system, &ipAddress, &sensor)) != kOK)
{
       printf("Error: GoSystem_FindSensorByIpAddress:%d\n", status);
       return;
}

// construct accelerator
if ((status = GoAccelerator_Construct(&accelerator, kNULL)) != kOK)
{
       printf("Error: GoAccelerator_Construct:%d\n", status);
       return;
}
// start accelerator
if ((status = GoAccelerator_Start(accelerator)) != kOK)
{
       printf("Error: GoAccelerator_Start:%d\n", status);
       return;
}
printf ("GoAccelerator_Start completed\n");
if ((status = GoAccelerator_Attach(accelerator, sensor)) != kOK)
{
       printf("Error: GoAccelerator_Attach:%d\n", status);
       return;
}

// create connection to GoSensor object
if ((status = GoSensor_Connect(sensor)) != kOK)
{
       printf("Error: GoSensor_Connect:%d\n", status);
       return;
}
```

After, the SDK application can control an accelerated sensor in the same way as an unaccelerated sensor.

Estimated Performance and Scan Rates

This section provides estimated scan rates and measurement tool performance.

Note that the estimates are based on tests done using Gocator firmware 5.2 SR1.

The following tables provide estimates of the scan rates of Gocator 3210, 3506, and 3504 under different field-of-view and resolution settings.

For these estimates, the following settings were used:

- Exposure was set to 4 milliseconds.
- No measurement tools added.
- **Reduce Occlusion** and **Acquire Intensity** were disabled.
- **Uniform Spacing** was enabled.

The estimated PC-based performance and scan rates (the "With PC Accelerator" columns) are based on the following hardware specifications:

PC

- Processor: Intel i7 5960X
- RAM: 16 GB
- Operating system: Windows 8.1 Pro

Graphics Card (only applies to the "+ CUDA" column)

- Processor: NVIDIA GeForce GTX 970
- RAM: 12 GB DDR5

Gocator 3210 Estimated Scan Rates

Field of View (X x Y x MR)	Spacing (mm)	Sensor-only (Hz)	With GoMax (Hz)	With PC Accelerator (Hz)	With PC Accelerator + CUDA (Hz)
100x154x110	0.5	6	10	10	10
100x154x110	0.2	1.8	10	10	10
100x154x110	0.1	0.5	4.7	10	10
100x154x110	**0.08**	0.3	4.0	10	10
100x154x110	0.05	0.15	2.1	5.4	7.2

Gocator 3506 Estimated Scan Rates

Field of View (X x Y x MR)	Spacing (mm)	Sensor-only (Hz)	With GoMax (Hz)	With PC Accelerator (Hz)	With PC Accelerator + CUDA (Hz)
30x45x25	0.2	4.3	8.5	8.5	8.5

Field of View (X x Y x MR)	Spacing (mm)	Sensor-only (Hz)	With GoMax (Hz)	With PC Accelerator (Hz)	With PC Accelerator + CUDA (Hz)
30x45x25	0.1	3.2	8.5	8.5	8.5
30x45x25	**0.02**	0.2	2.7	7.7	8.5

Gocator 3504 Estimated Scan Rates

Field of View (X x Y x MR)	Spacing (mm)	Sensor-only (Hz)	With GoMax (Hz)	With PC Accelerator (Hz)	With PC Accelerator + CUDA (Hz)
14x18x7	0.05	2.9	7.5	7.5	7.5
14x18x7	0.02	1	6.4	7.5	7.5
14x18x7	0.01	0.3	2.9	6.4	7.4
14x18x7	**0.006**	0.15	2.0	5.4	6

The following table lists the scan rate speed increase factor when accelerating Gocator 3210, 3506, and 3504 with GoMax. Sensors are set at their default resolutions. The same settings listed above were used for these results.

Gocator 3210, 3506 & 3504 Scan Rate Increase Factors

Sensor	Field of View (X x Y x MR)	Spacing (mm)	Sensor-only (Hz)	With GoMax (Hz)	Speed Increase Factor
3210	100x154x110	0.08	0.3	4.0	13.3
3506	30x45x25	0.02	0.2	2.7	13.5
3504	14x18x7	0.006	0.15	2.0	13.3

The following table lists the running time of various measurement tools, with and without GoMax, as well as the performance increase factor when running with GoMax.

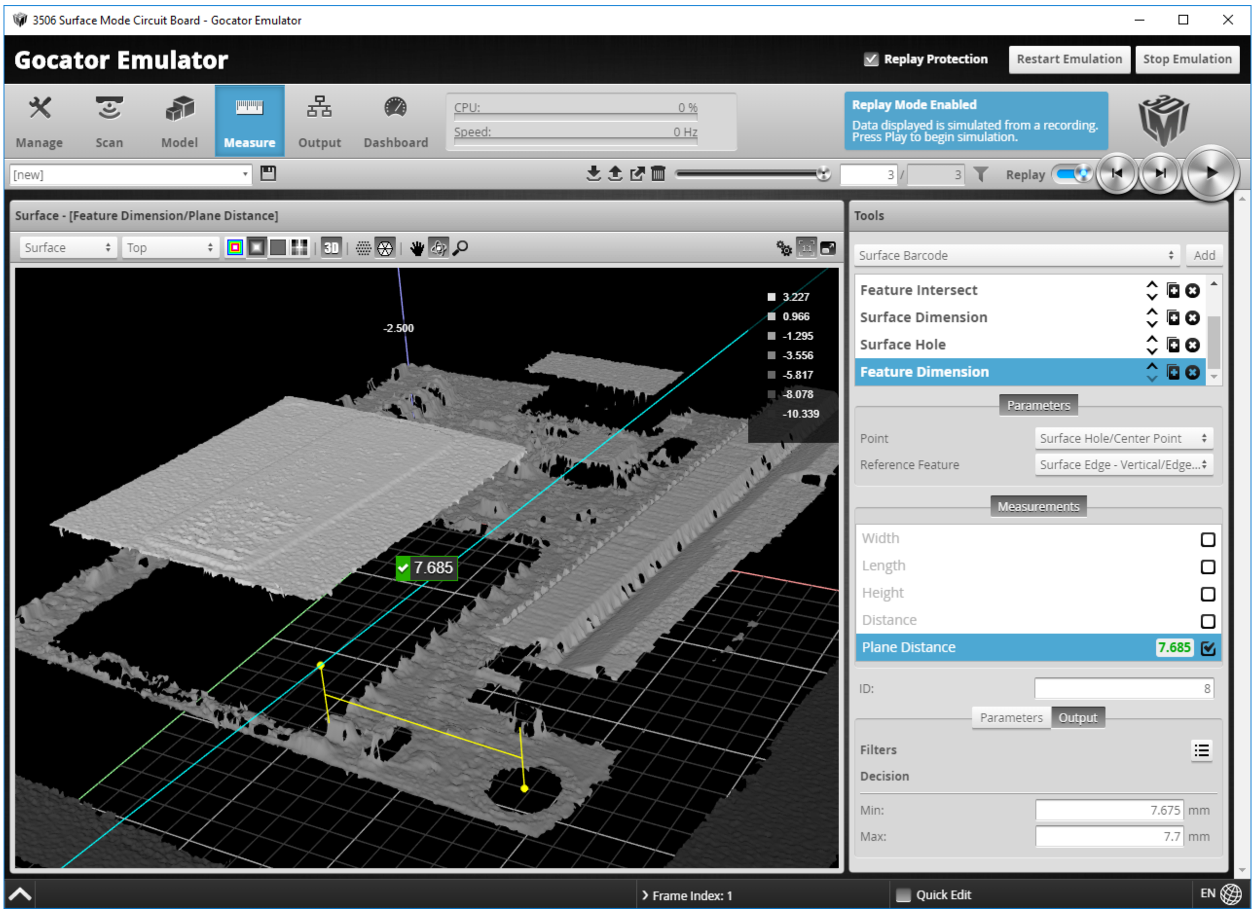
Note that although sensor models and job file configurations will affect running times, the performance increase factor for tools should be consistent across models and configurations.

Gocator 3210 Performance Increase Factors

Measurement Tool	Running Time on Sensor (ms)	Running Time with GoMax (ms)	Performance Increase Factor
Cylinder Head Volume	15	2.2	6.8

Gocator Emulator

The emulator is a stand-alone application that lets you run a "virtual" sensor, encapsulated in a "scenario." When running a scenario, you can test jobs, evaluate data, and even learn more about new features, rather than take a physical device off the production line to do this. You can also use a scenario to familiarize yourself with the overall interface if you are new to Gocator.



Emulator showing a part in recorded data.
A measurement is applied to the recorded data.

System Requirements

The following are the system requirements for the software:

PC

- Processor: Intel Core i3 or equivalent (64-bit)
- RAM: 4 GB

- Hard drive: 500 GB
- Operating system: Windows 7 or higher (64-bit)

Limitations

In most ways, a scenario behaves like a real sensor, especially when visualizing data, setting up models and part matching, and adding and configuring measurement tools. The following are some of the limitations:

- Changes to job files in the emulator are *not* persistent (they are lost when you close or restart the emulator). However, you can keep a modified job by first saving it and then downloading it from the **Jobs** list on the **Manage** page to a client computer. The job file can then be loaded into the emulator at a later time or even onto a physical sensor for final testing.
- Performing alignment in the emulator has no effect and will never complete.
- The emulator does not support the PROFINET protocol.

For information on saving and loading jobs in the emulator, see *Creating, Saving, and Loading Jobs* on page 543.

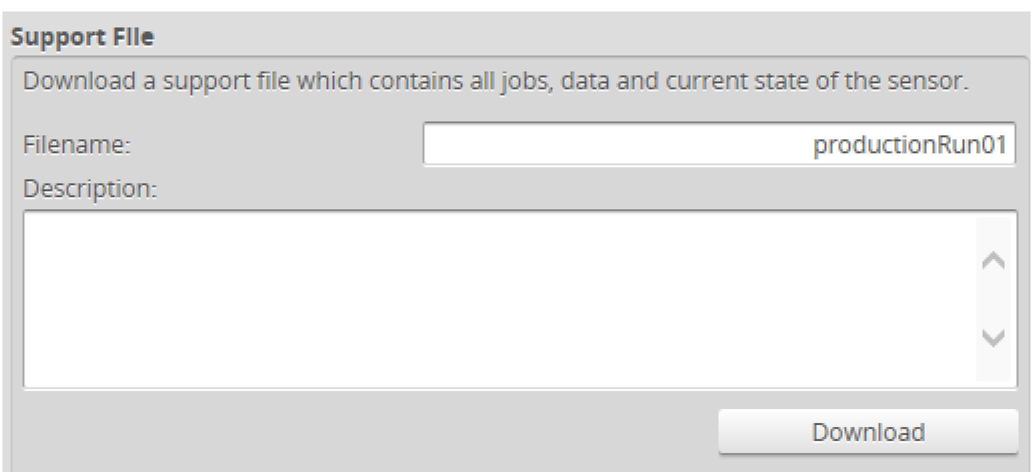
For information on uploading and downloading jobs between the emulator and a computer, and performing other job file management tasks, see *Downloading and Uploading Jobs* on page 547.

Downloading a Support File

The emulator provides several preinstalled scenarios.

You can also create scenarios yourself by downloading a support file from a physical sensor and then adding it to the emulator.

Support files can contain jobs, letting you configure systems and add measurements in an emulated sensor. Support files can also contain replay data, letting you test measurements and some configurations on real data. Dual-sensor systems are supported.



To download a support file:

1. Go to the **Manage** page and click on the **Support** category.
2. In **Filename**, type the name you want to use for the support file.

   When you create a scenario from a support file in the emulator, the filename you provide here is displayed in the emulator's scenario list.

   Support files end with the .gs extension, but you do not need to type the extension in **Filename**.
3. (Optional) In **Description**, type a description of the support file.

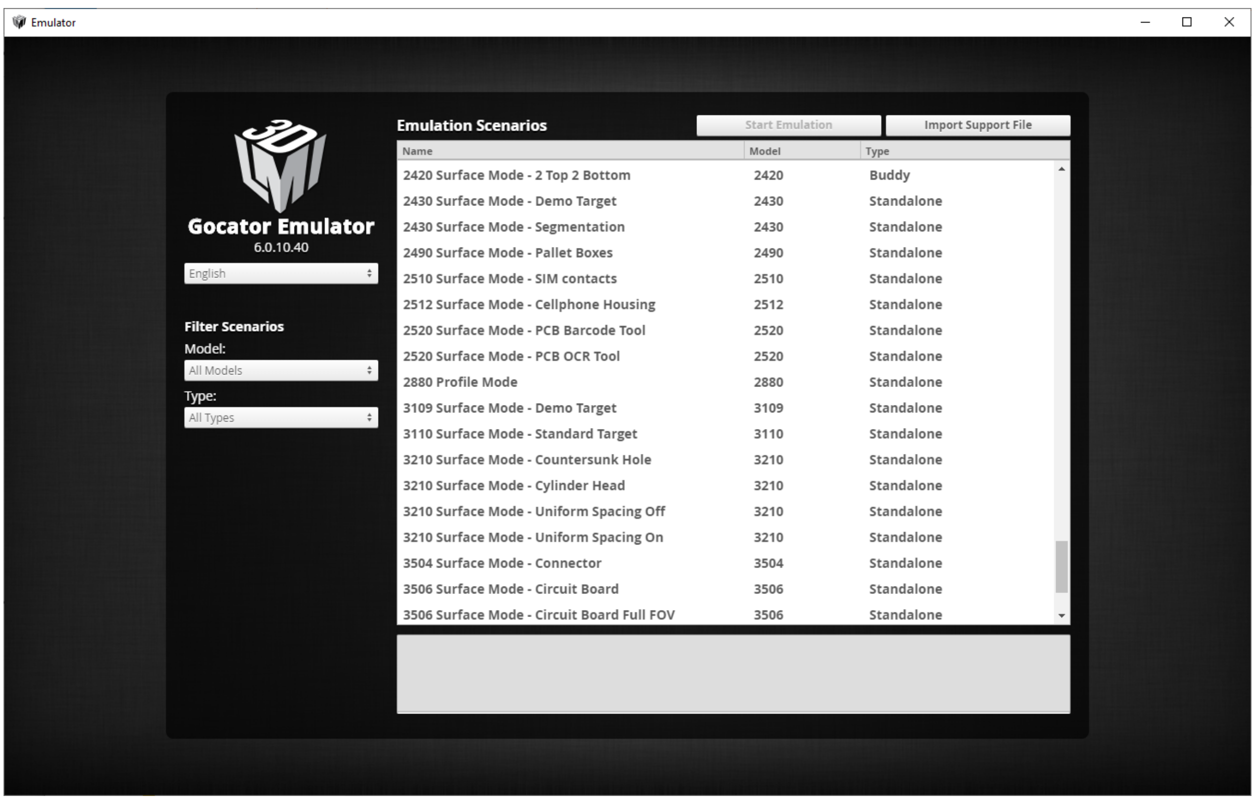
   When you create a scenario from a support file in the emulator, the description is displayed below the emulator's scenario list.
4. Click **Download**, and then when prompted, click **Save**.

⚠	Downloading a support file stops the sensor.

Running the Emulator

The emulator is contained in the utilities package (14405-x.x.x.x_SOFTWARE_GO_Utilities.zip). To get the package, go to http://lmi3d.com/support, choose your product from the Product Downloads section, and download the package from the Download Center.

To run the emulator, unzip the package and double-click the *GoEmulator* link in the unzipped *Emulator and Accelerator* subfolder.



Emulator launch screen

You can change the language of the emulator's interface from the launch screen. To change the language, choose a language option from the top drop-down:

Selecting the emulator interface language

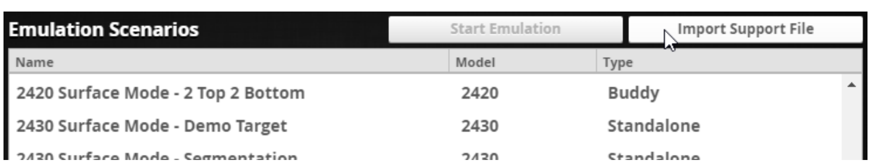
Adding a Scenario to the Emulator

To simulate a physical sensor using a support file downloaded from a sensor, you must add it as a scenario in the emulator.

> You can add support files downloaded from any series of Gocator sensors to the emulator.

To add a scenario:

1. Launch the emulator if it isn't running already.
2. Click the **Add** button and choose a previously saved support file (.gs extension) in the **Choose File to Upload** dialog.

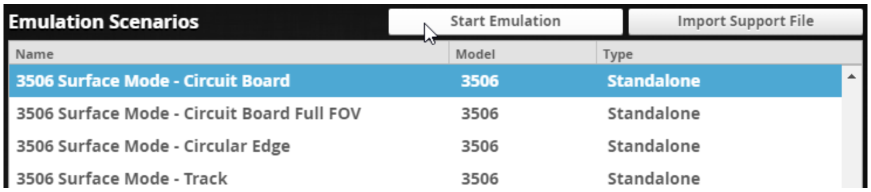


3. (Optional) In the field below the list, type a description.

> You can only add descriptions for user-added scenarios.

Running a Scenario

After you have added a virtual sensor by uploading a support file to the emulator, you can run it from the **Available Scenarios** list on the emulator launch screen. You can also run any of the scenarios included in the installation.



To run a scenario:

1. If you want to filter the scenarios listed in **Available Scenarios**, do one or both of the following:
   - Choose a model family in the **Model** drop-down.
   - Choose **Standalone** or **Buddy** to limit the scenarios to single-sensor or dual-/multi-sensor scenarios, respectively.
2. Select a scenario in the **Available Scenarios** list and double-click it in the list or click **Start**.

Removing a Scenario from the Emulator

You can easily remove a scenario from the emulator.

You can only remove user-added scenarios.

To remove a scenario:

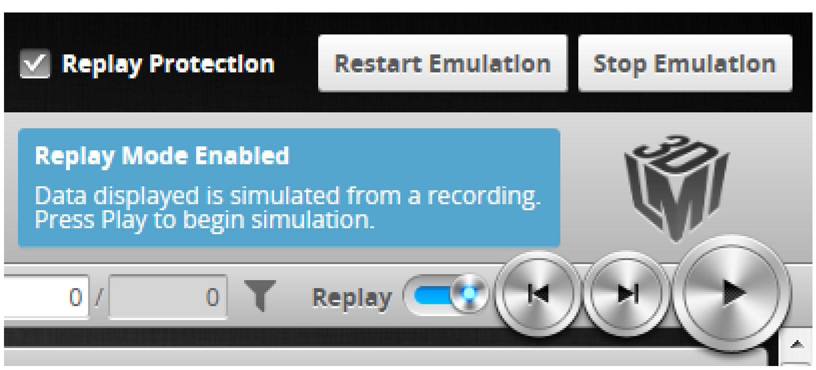
1. If the emulator is running a scenario, click 

to stop it.
2. In the **Available Scenarios** list, scroll to the scenario you want to remove.
3. Click the  button next to the scenario you want to remove.

   The scenario is removed from the emulator.

Using Replay Protection

Making changes to certain settings on the **Scan** page causes the emulator to flush replay data. The **Replay Protection** option protects replay data by preventing changes to settings that affect replay data. Settings that do not affect replay data can be changed.



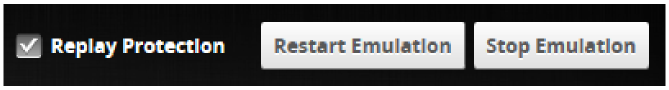
If you try to uncheck **Replay Protection**, you must confirm that you want to disable it.

Replay Protection is on by default.

Stopping and Restarting the Emulator

To stop the emulator:

- Click **Stop Emulation**.



Stopping the emulator returns you to the launch screen.

To restart the emulator when it is running:

- Click **Restart Emulation**.

Restarting the emulator restarts the currently running simulation.

Running the Emulator in Default Browser

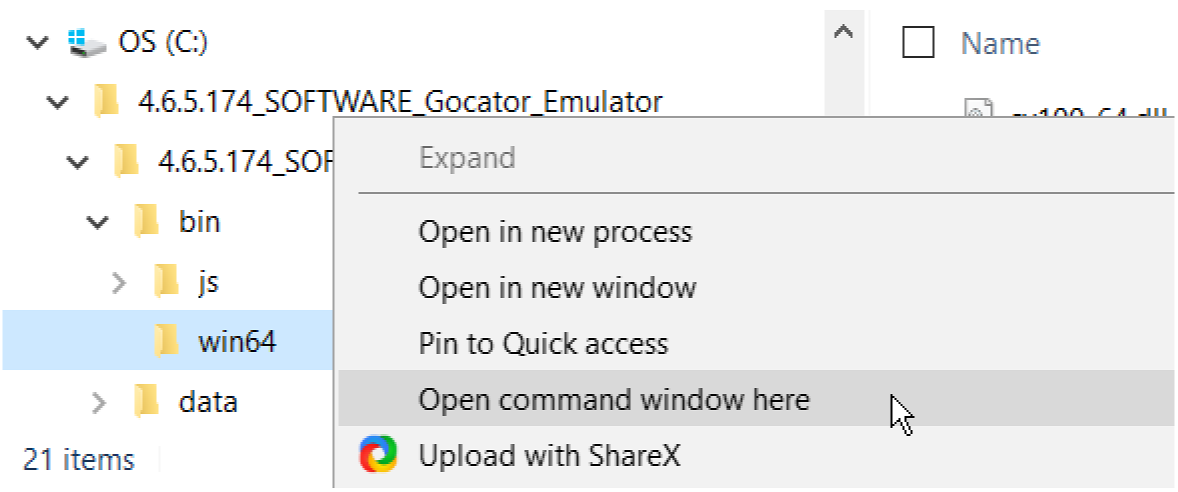
When you use the `/browser` command line parameter, the emulator application launches normally but also launches in your default browser. This provides additional flexibility when using the emulator. For example, you can resize the emulator running in a browser window.

To run the emulator in your default browser:

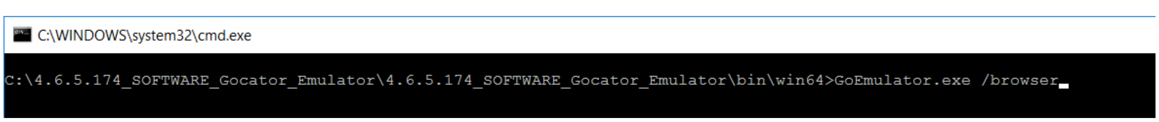
1. In Windows Explorer (Windows 7) or File Explorer (Windows 8 or 10), browse to the location of the emulator.

   The emulator is under `bin\win64`, in the location in which you installed the emulator.

2. Press and hold `Shift`, right-click the `win64` folder containing the emulator, and choose **Open command window here** (or **Open PowerShell window here**).



3. In the command prompt, type `GoEmulator.exe /browser` (or `.\GoEmulator.exe /browser` for PowerShell).



   After the emulator application starts, the emulator also launches in your default browser.

Working with Jobs and Data

The following topics describe how to work with jobs and replay data (data recorded from a physical sensor) in a scenario running on the emulator.

Creating, Saving, and Loading Jobs

Changes saved to job files in the emulator are *not* persistent (they are lost when you close or restart the emulator). To keep jobs permanently, you must first save the job in the emulator and then download the job file to a client computer. See below for more information on creating, saving, and switching jobs. For information on downloading and uploading jobs between the emulator and a computer, see *Downloading and Uploading Jobs* on page 547.

The job drop-down list in the toolbar shows the jobs available in the emulator. The job that is currently active is listed at the top. The job name will be marked with "[unsaved]" to indicate any unsaved changes.

To create a job:

1. Choose **[New]** in the job drop-down list and type a name for the job.
2. Click the **Save** button or press **Enter** to save the job.
   The job is saved to the emulator using the name you provided.

To save a job:

- Click the **Save** button.

  The job is saved to the emulator.

To load (switch) jobs:

- Select an existing file name in the job drop-down list.

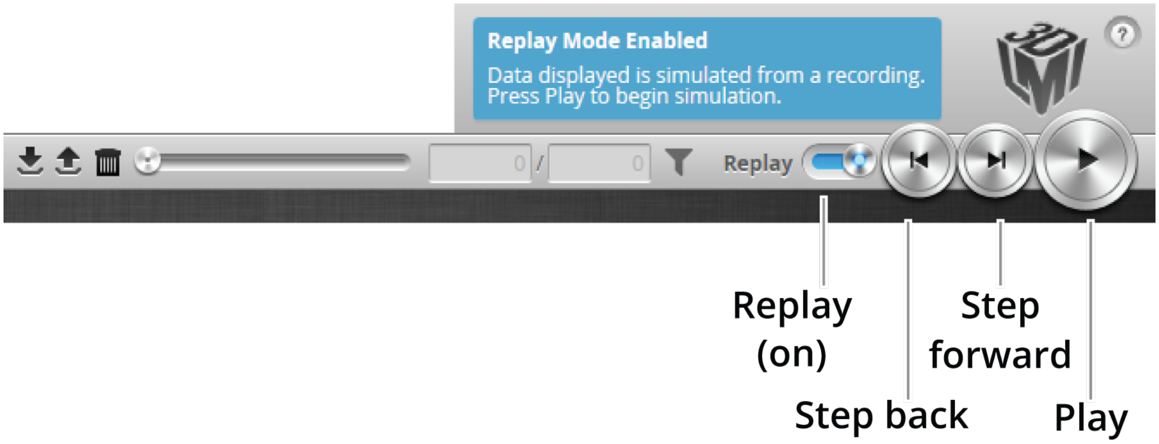
  The job is activated. If there are any unsaved changes in the current job, you will be asked whether you want to discard those changes.

Playback and Measurement Simulation

The emulator can replay scan data previously recorded by a physical sensor, and also simulate measurement tools on recorded data. This feature is most often used for troubleshooting and fine-tuning measurements, but can also be helpful during setup.

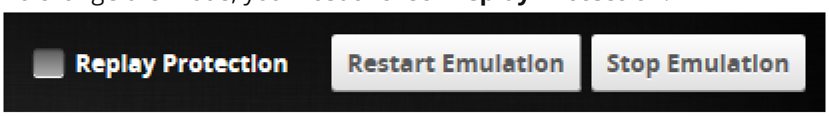
Playback is controlled using the toolbar controls.

	Recording is not functional in the emulator.



Playback controls when replay is on

To replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue.
   To change the mode, you must uncheck **Replay Protection**.
   

2. Use the **Replay** slider or the **Step Forward**, **Step Back**, or **Play** buttons to review data.
   The **Step Forward** and **Step Back** buttons move the current replay location forward and backward by a single frame, respectively.
   The **Play** button advances the replay location continuously, animating the playback until the end of the replay data.
   The **Stop** button (replaces the **Play** button while playing) can be used to pause the replay at a particular location.
   The **Replay** slider (or **Replay Position** box) can be used to go to a specific replay frame.

To simulate measurements on replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue.
   To change the mode, **Replay Protection** must be unchecked.

2. Go to the **Measure** page.
   Modify settings for existing measurements, add new measurement tools, or delete measurement tools as desired. For information on adding and configuring measurements, see *Measurement and Processing* on page 146.

3. Use the **Replay Slider**, **Step Forward**, **Step Back**, or **Play** button to simulate measurements.
   Step or play through recorded data to execute the measurement tools on the recording.
   Individual measurement values can be viewed directly in the data viewer. Statistics on the measurements that have been simulated can be viewed in the **Dashboard** page; for more information on the dashboard, see *Dashboard* on page 524.

To clear replay data:

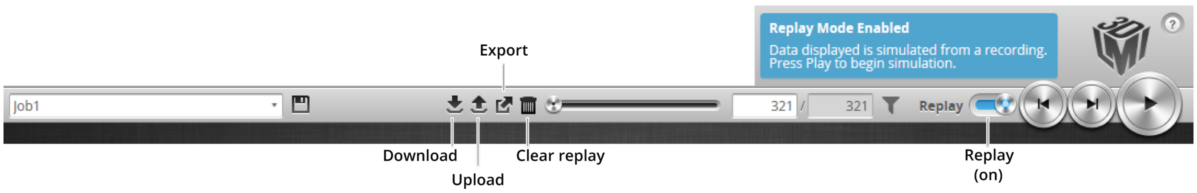
- Click the **Clear Replay Data** button.

Downloading, Uploading, and Exporting Replay Data

Replay data (recorded scan data) can be downloaded from the emulator to a client computer, or uploaded from a client computer to the emulator.

Data can also be exported from the emulator to a client computer in order to process the data using third-party tools.

You can only upload replay data to the same sensor model that was used to create the data.



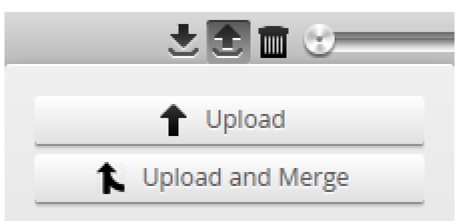
Replay data is not loaded or saved when you load or save jobs.

To download replay data:

1. Click the Download button.
2. In the **File Download** dialog, click **Save**.
3. In the **Save As...** dialog, choose a location, optionally change the name, and click **Save**.

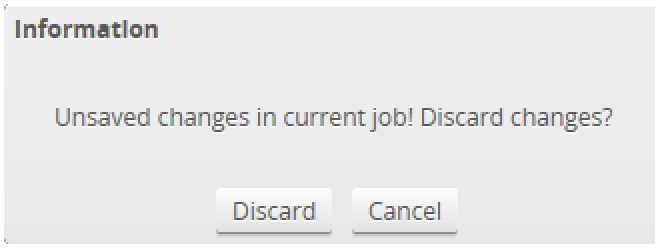
To upload replay data:

1. Click the Upload button.
   The Upload menu appears.



2. In the Upload menu, choose one of the following:
   - **Upload**: Unloads the current job and creates a new unsaved and untitled job from the content of the replay data file.
   - **Upload and merge**: Uploads the replay data and merges the data's associated job with the current job. Specifically, the settings on the **Scan** page are overwritten, but all other settings of the current job are preserved, including any measurements or models.

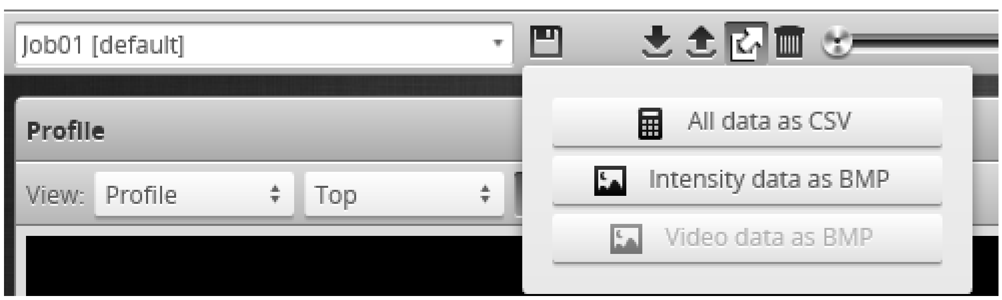
   If you have unsaved changes in the current job, the firmware asks whether you want to discard the changes.



3. Do one of the following:
   - Click **Discard** to discard any unsaved changes.
   - Click **Cancel** to return to the main window to save your changes.
4. If you clicked **Discard**, navigate to the replay data to upload from the client computer and click **OK**.
   The replay data is loaded, and a new unsaved, untitled job is created.

Replay data can be exported using the CSV format.

	Surface intensity data cannot be exported to the CSV format. It can only be [exported separately as a bitmap]().



To export replay data in the CSV format:

1. In the **Scan Mode** panel, switch to Profile or Surface.
2. Switch to Replay mode.
3. Click the Export button and select **All Data as CSV**.
   Only data at the current replay location is exported.
   Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Playback and Measurement Simulation* on page 543.
4. (Optional) Convert exported data to another format using the CSV Converter Tool. For information on this tool, see *CSV Converter Tool* on page 862.

	The decision values in the exported data depend on the *current* state of the job, not the state during recording. For example, if you record data when a measurement returns a *pass* decision, change the measurement's settings so that a *fail* decision is returned, and then export to CSV, you will see a *fail* decision in the exported data.

Recorded intensity data can be exported to a bitmap (.BMP format). **Acquire Intensity** must be checked in the **Scan Mode** panel while data was being recorded in order to export intensity data.

To export recorded intensity data to the BMP format:

- Switch to Replay mode and click the **Export** button and select **Intensity data as BMP**.

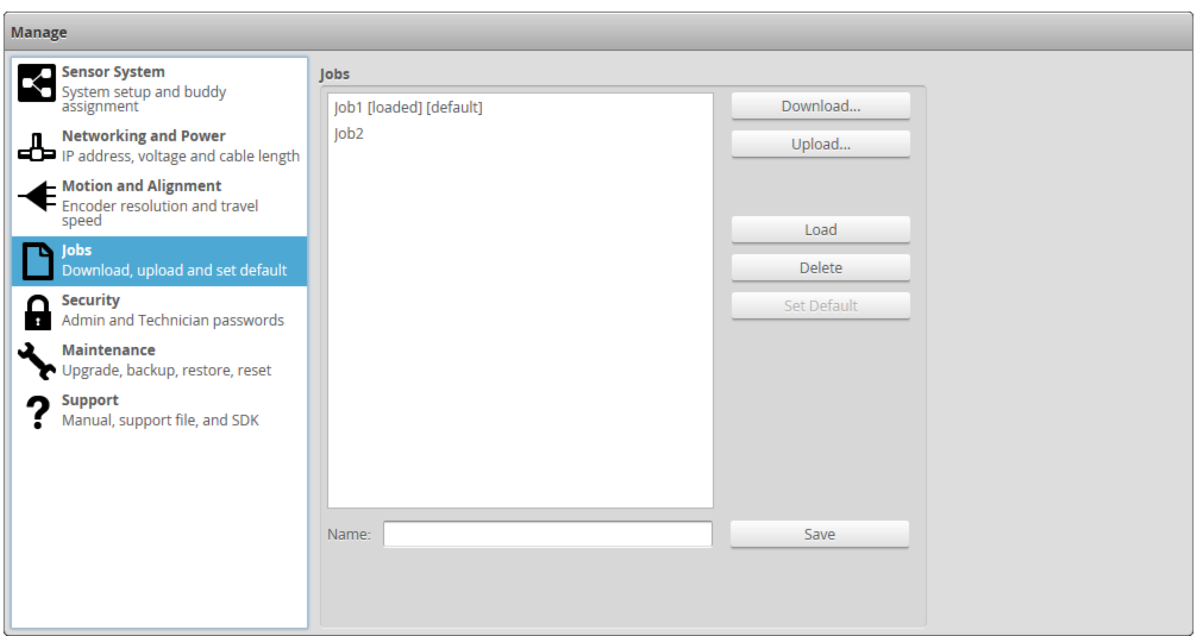
  Only the intensity data in the current replay location is exported.
  Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Playback and Measurement Simulation* on page 543.

To export video data to a BMP file:

1. In the **Scan Mode** panel, switch to Video mode.
   Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Playback and Measurement Simulation* on page 543.
2. Switch to Replay mode.
3. Click the Export button and select **Video data as BMP**.

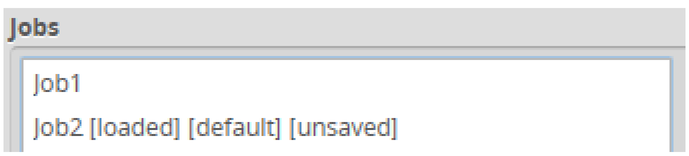
Downloading and Uploading Jobs

The **Jobs** category on the **Manage** page lets you manage the jobs in the emulator.



Element	Description
Name field	Used to provide a job name when saving files.
Jobs list	Displays the jobs that are currently saved in the emulator.
Save button	Saves current settings to the job using the name in the **Name** field. Changes to job files are not persistent in the emulator. To keep changes, first save changes in the job file, and then download the job file to a client computer. See the procedures below for instructions.
Load button	Loads the job that is selected in the job list. Reloading the current job discards any unsaved changes.
Delete button	Deletes the job that is selected in the job list.
Set as Default button	Setting a different job as the default is not persistent in the emulator. The job set as default when the support file (used to create a virtual sensor) was downloaded is used as the default whenever the emulator is started.
Download... button	Downloads the selected job to the client computer.
Upload... button	Uploads a job from the client computer.

Unsaved jobs are indicated by "[unsaved]".



> Changes to job files in the emulator are *not* persistent (they are lost when you close or restart the emulator). However, you can keep modified jobs by first saving them and then downloading them to a client computer.

To save a job:

1. Go to the **Manage** page and click on the **Jobs** category.
2. Provide a name in the **Name** field.
   To save an existing job under a different name, click on it in the **Jobs** list and then modify it in the **Name** field.
3. Click on the **Save** button or press **Enter**.

To download, load, or delete a job, or to set one as a default, or clear a default:

1. Go to the **Manage** page and click on the **Jobs** category.
2. Select a job in the **Jobs** list.
3. Click on the appropriate button for the operation.

Scan, Model, and Measurement Settings

The settings on the **Scan** page related to actual scanning will clear the buffer of any scan data that is uploaded from a client computer, or is part of a support file used to create a virtual sensor. If **Replay Protection** is checked, the emulator will indicate in the log that the setting can't be changed because the change would clear the buffer. For more information on Replay Protection, see *Using Replay Protection* on page 541.

Other settings on the **Scan** page related to the post-processing of data can be modified to test their influence on scan data, without modifying or clearing the data, for example edge filtering (page 113), and filters on the X axis (page 108). Note that modifying the Y filters causes the buffer to be cleared.

For information on creating models and setting up part matching, see *Models* on page 126. For information on adding and configuring measurement tools, see *Measurement and Processing* on page 146.

Calculating Potential Maximum Frame Rate

You can use the emulator to calculate the potential maximum frame rate you can achieve with different settings.

For example, when you reduce the active area, in the **Active Area** tab on the **Sensor** panel, the maximum frame rate displayed on the **Trigger** panel is updated to reflect the increased speed that would be available in a physical sensor. (See *Active Area* on page 94 for more information on active area.)

Similarly, you can adjust exposure on the **Exposure** tab on the **Sensor** panel to see how this affects the maximum frame rate. (See *Exposure* on page 96 for more information on exposure.)

	To adjust active area in the emulator, **Replay Protection** must be turned off. See *Using Replay Protection* on page 541 for more information.

	Saving changes to active area causes replay data to be flushed.

Protocol Output

The emulator simulates output for all of Gocator's Ethernet-based protocols, with the exception of PROFINET.

- Gocator
- ASCII
- Modbus
- EtherNet/IP

Clients (such as PLCs) can connect to the emulator to access the simulated output and use the protocols as they would with a physical sensor.

The emulator allows connections to emulated sensors on localhost (127.0.0.1). You can also allow connections to emulated sensors on your computer's network card; for more information, see *Remote Operation* on the next page.

Remote Operation

You can specify the IP address of one of your computer's network cards to allow clients to connect remotely to an emulated sensor using the `/ip` command line parameter. When the `/ip` parameter is not used, emulated sensors are only available on the local machine (that is, 127.0.0.1 or localhost).

> Clients can only connect to emulated sensors, not to the emulator's launch page.

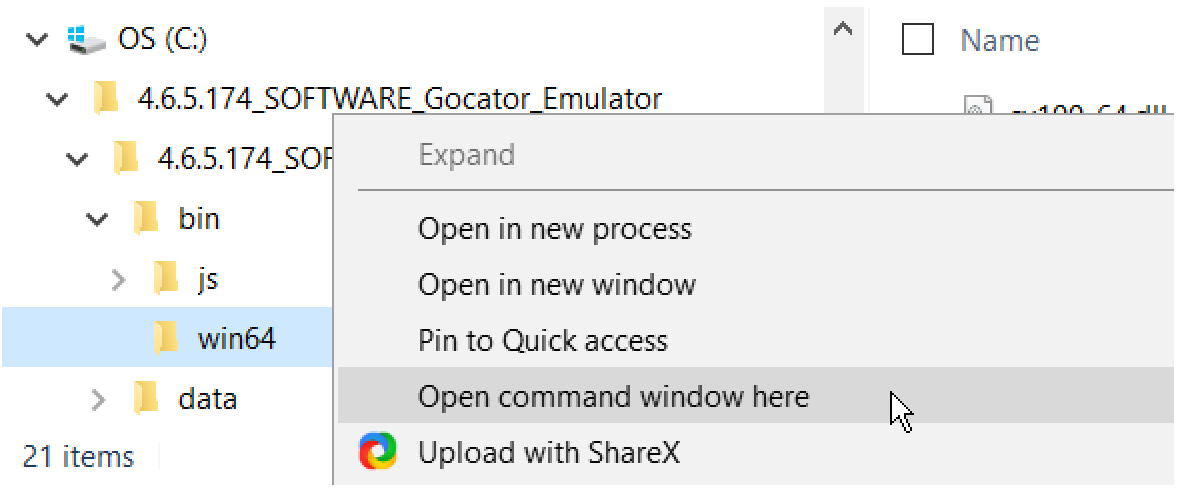
> You may need to contact your network administrator to allow connections to the computer running the emulated sensor.

To allow remote connections to an emulated sensor:

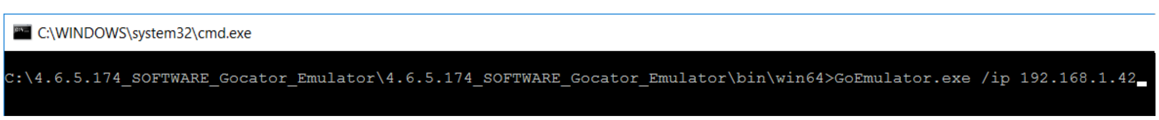
1. In Windows Explorer (Windows 7) or File Explorer (Windows 8 or 10), browse to the location of the emulator.

   The emulator is under `bin\win64`, in the location in which you installed the emulator.

2. Press and hold `Shift`, right-click the `win64` folder containing the emulator, and choose **Open command window here** (or **Open PowerShell window here**).



3. In the command prompt, type `GoEmulator.exe /ip`, followed by a valid IPV4 address on your network.



   The emulator application starts.

   > The emulator does not check that the IP address is valid.

4. From the emulator launch page, start a scenario.

   For more information, see *Running a Scenario* on page 540.

5. Provide the IP address you used with the `/ip` parameter, followed by port number 3191, to users who want to connect to the emulated sensor, for example:

```
192.168.1.42:3191
```

Sensor Device Files

This section describes the user-accessible device files stored on a sensor.

Live Files

Various "live" files stored on a sensor represent the sensor's active settings and transformations (represented together as "job" files), the active replay data (if any), and the sensor log.

By changing the live job file, you can change how the sensor behaves. For example, to make settings and transformations active, write to or copy to the _live.job file. You can also save active settings or transformations to a client computer, or to a file on the sensor, by reading from or copying these files, respectively.

	The live files are stored in volatile storage. Only user-created job files are stored in non-volatile storage.

The following table lists the live files:

Live Files

Name	Read/Write	Description
_live.job	Read/Write	The active job. This file contains a Configuration component containing the current settings. If Alignment Reference in the active job is set to Dynamic, it also contains a Transform component containing transformations. For more information on job files (live and user-created), accessing their components, and their structure, see *Job File Structure* on the next page.
_live.cfg	Read/Write	A standalone representation of the Configuration component contained in _ live.job. Used primarily for backwards compatibility.
_live.tfm	Read/Write	**If Alignment Reference of the active job is set to Dynamic:** A copy of the Transform component in _live.job. Used primarily for backwards compatibility. **If Alignment Reference of the active job is set to Fixed:** The transformations that are used for *all* jobs whose Alignment Reference setting is set to Fixed.
_live.log	Read	A sensor log containing various messages. For more information on the log file, see *Log File* below.
_live.rec	Read/Write	The active replay simulation data.
ExtendedId.xml	Read	Sensor identification.

Log File

The log file contains log messages generated by the sensor. The root element is *Log*.

To access the log file, use the Read File command, passing "_live.log" to the command. The log file is read-only.

Log Child Elements

Element	Type	Description
@idStart	64s	Identifier of the first log.
@idEnd	64s	Identifier of the final log.
List of (Info | Warning | Error)	List	An ordered list of log entries. This list is empty if idEnd < idStart.

Log/Info | Log/Warning | Log/Error Elements

Element	Type	Description
@time	64u	Log time, in uptime (μs).
@source	32u	The serial number of the sensor the log was produced by.
@id	32u	The Indentifier, or index, of the log
@value	String	Log content; may contain printf-style format specifiers (e.g. %u).
List of (IntArg | FloatArg | Arg)	List	An ordered list of arguments: IntArg – Integer argument FloatArg – Floating-point argument Arg – Generic argument

The arguments are all sent as strings and should be applied in order to the format specifiers found in the content.

Job File Structure

The following sections describe the structure of job files.

Job files, which are stored in a sensor's internal storage, control system behavior when a sensor is running. Job files contain the settings and potentially the transformations and models associated with the job (if Alignment Reference is set to Dynamic).

There are two kinds of job files:

- A special job file called "_live.job." This job file contains the *active* settings and potentially the transformations and models associated with the job. It is stored in volatile storage.
- Other job files that are stored in non-volatile storage.

Job File Components

A job file contains components that can be loaded and saved as independent files. The following table lists the components of a job file:

Job File Components

Component	Path	Description
Configuration	config.xml	The job's configurations. This component is always present. For more

Component	Path	Description
		information, see *Configuration* below.
Transform	transform.xml	Transformation values. Present only if Alignment Reference is set to Dynamic. For more information, see *Transform* on page 645.
Part model	<name>.mdl	One or more part model files. Part models are created using models and part matching. For more information, see *Part Models* on page 647.

Elements in the components contain three types of values: settings, constraints, and properties. Settings are input values that can be edited. Constraints are read-only limits that define the valid values for settings. Properties are read-only values that provide supplemental information related to sensor setup.

When a job file is received from a sensor, it will contain settings, constraints, and properties. When a job file is sent to a sensor, any constraints or properties in the file will be ignored.

Changing the value of a setting can affect multiple constraints and properties. After you upload a job file, you can download the job file again to access the updated values of the constraints and properties.

Accessing Files and Components

Job file components can be accessed individually as XML files using path notation. For example, the configurations in a user-created job file called *productionRun01.job* can be read by passing "productionRun01.job/config.xml" to the Read File command. In the same way, the configurations in the active job could be read using "_live.job/config.xml".

> If Alignment Reference is set to Fixed, the active job file (_live.job) will not contain transformations. To access transformations in this case, you must access them via _live.tfm.

> The following sections correspond to the XML structure used in job file components.

Configuration

The Configuration component of a job file contains settings that control how a sensor behaves.

You can access the Configuration component of the active job as an XML file, either using path notation, via "_live.job/config.xml", or directly via "_live.cfg".

You can access the Configuration component in user-created job files in non-volatile storage, for example, "productionRun01.job/config.xml". You can only access configurations in user-created job files using path notation.

See the following sections for the elements contained in this component.

All sensors share a common job file structure and settings for all features are included in job files, regardless of the model.

> If a setting in a job file is not used by a sensor, the setting's *used* property is set to 0.

Configuration Child Elements

Element	Type	Description
@version	32u	Configuration version (101).
@versionMinor	32u	Configuration minor version (9).
Setup	Section	For a description of the Setup elements, see *Setup* below.
Replay	Section	Contains settings related to recording filtering (see *Replay* on page 578).
Streams	Section	Read-only collection of available data streams (see *Streams/Stream (Read-only)* on page 579).
ToolOptions	Section	List of available tool types and their information. See *ToolOptions* on page 580 for details.
Tools	Collection	Collection of sections. Each section is an instance of a tool and is named by the type of the tool it describes. For more information, see the sections for each tool under *Tools* on page 583.
Tools.options	String (CSV)	Deprecated. Replaced by ToolOptions.
Outputs	Section	For a description of the Output elements, see *Output* on page 638.

Setup

The Setup element contains settings related to system and sensor setup.

Setup Child Elements

Element	Type	Description
TemperatureSafetyEnabled	Bool	Enables laser temperature safety control. Only applies to certain laser-based sensors.
TemperatureSafetyEnabled.used	Bool	Whether or not this property is used.
ScanMode	32s	The default scan mode.
ScanMode options	String (CSV)	List of available scan modes.
OcclusionReductionEnabled	Bool	Enables occlusion reduction.
OcclusionReductionEnabled.used	Bool	Whether or not property is used.
OcclusionReductionEnabled.value	Bool	Actual value used if not configurable.
OcclusionReductionAlg	32s	The Algorithim to use for occlusion reduction: 0 – Standard 1 – High Quality
OcclusionReductionAlg.used	Bool	Whether or not property is used
OcclusionReductionAlg.value	Bool	Actual value used if not configurable
UniformSpacingEnabled	Bool	Enables uniform spacing.
UniformSpacingEnabled.used	Bool	Whether or not property is used.
UniformSpacingEnabled.readonly	Bool	Whether or not property can be modified.
UniformSpacingEnabled.value	Bool	Actual value used if not configurable.
IntensityEnabled	Bool	Enables intensity data collection.

Element	Type	Description
IntensityEnabled.used	Bool	Whether or not property is used.
IntensityEnabled.value	Bool	Actual value used if not configurable.
FlickerFreeModeEnabled	Bool	Enables flicker-free operation.
FlickerFreeModeEnabled.used	Bool	Whether flicker-free operation can be used on this sensor.
ExternalInputZPulseEnabled	Bool	Enables the External Input based encoder Z Pulse feature.
ExternalInputZPulseIndex	32u	Input index to use for the input triggered z pulse feature.
ExternalInputZPulseEnabled.used	Bool	Whether the index can be set.
BackgroundSuppression	Section	See *BackgroundSuppression* below.
Filters	Section	See *Filters* below.
Trigger	Section	See *Trigger* on page 558.
Layout	Section	See *Layout* on page 560.
Alignment	Section	See *Alignment* on page 561.
Devices	Collection	A collection of two Device sections (with roles main and buddy). See *Devices / Device* on page 563.
SurfaceGeneration	Section	See *SurfaceGeneration* on page 571. Used by profile sensors.
SurfaceSections	Section	See *SurfaceSections* on page 572.
ProfileGeneration	Section	See *ProfileGeneration* on page 573. Used by Gocator displacement sensors.
PartDetection	Section	See *PartDetection* on page 574.
PartMatching	Section	See *PartMatching* on page 576.
Custom	Custom	Used by specialized sensors.

BackgroundSuppression

The BackgroundSuppression element contains settings related to background suppression.

BackgroundSuppression Child Elements

Element	Type	Description
Enabled	Bool	Enables background suppression.
FrameRatio	64f	Ratio of background frames to calibration frames

Filters

The Filters element contains settings related to post-processing profiles before they are output or used by measurement tools.

XSmoothing

XSmoothing Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YSmoothing

YSmoothing Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

XGapFilling

XGapFilling Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YGapFilling

YGapFilling Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

XMedian

XMedian Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.

Element	Type	Description
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YMedian

YMedian Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

XDecimation

XDecimation Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YDecimation

YDecimation Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

XSlope

> This filter is only available on displacement sensors.

XSlope Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YSlope

This filter is only available on displacement sensors.

YSlope Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

Trigger

The Trigger element contains settings related to trigger source, speed, and encoder resolution.

Trigger Child Elements

Element	Type	Description
Source	32s	Trigger source: 0 – Time 3 – Software
Source.options	32s (CSV)	List of available source options.
ExternalInputIndex	32s	Index of external input when Source (above) is set to 2 – Digital Input and connected to a Master. 0 – first digital input 1 – second digital input 2 – third digital input 3 – fourth digital input
ExternalInputIndex.options	32s (CSV)	List of available external input indices.
ExternalInputIndex.used	Bool	Whether the external input index used.
Units	32s	Sensor triggering units when source is not clock or encoder: 0 – Time 1 – Encoder
FrameRate	64f	Frame rate for time trigger (Hz).
FrameRate.min	64f	Minimum frame rate (Hz).

Element	Type	Description
FrameRate.max	64f	Maximum frame rate (Hz).
FrameRate.maxSource	32s	Source of maximum frame rate limit: 0 – Imager 1 – Surface generation
TracheidRate	64f	The frame rate of Tracheid data (Read Only)
TracheidRate.used	Bool	Whether the sensor has a Tracheid data rate.
FrameDataRate	64f	The frame rate of normal (range/profile/surface) data (Read Only)
FrameDataRate.used	Bool	Whether the sensor has a separate FrameDataRate
EncoderSpacing.min	64f	Minimum encoder spacing (mm).
EncoderSpacing.max	64f	Maximum encoder spacing (mm).
EncoderSpacing.minSource	32s	Source of minimum encoder spacing: 0 – Resolution 1 – Surface generation
EncoderSpacing.used	Bool	Whether or not this parameter is configurable.
EncoderTriggerMode	32s	Encoder triggering mode: 0 – Tracking backward 1 – Bidirectional 2 – Ignore backward
Delay	64f	Trigger delay (µs or mm).
Delay.min	64f	Minimum trigger delay (µs or mm).
Delay.max	64f	Maximum trigger delay (µs or mm).
GateEnabled	Bool	Enables digital input gating.
GateEnabled.used	Bool	True if this parameter can be configured.
GateEnabled.value	Bool	Actual value if the parameter cannot be configured.
BurstEnabled	Bool	Enables burst triggering.
BurstEnabled.Used	Bool	Whether or not this parameter is configurable.
BurstCount	32u	Number of scans to take during burst triggering.
BurstCount.used	Bool	Whether or not this parameter is configurable.
BurstCount.max	32u	Maximum burst count.
ReversalDistanceAutoEnabled	Bool	Whether or not to use auto-calculated value.
ReversalDistanceAutoEnabled.used	Bool	Whether or not this parameter can be configured.
ReversalDistance	64f	Encoder reversal threshold (for jitter handling)
ReversalDistance.used	Bool	Whether or not this parameter is used.
ReversalDistance.value	64f	Actual value.
LaserSleepMode.used	Bool	Whether or not this feature can be configured. Laser sleep mode settings are not used by snapshot sensors.

Element	Type	Description
LaserSleepMode/Enabled	Bool	Enables or disables the feature.
LaserSleepMode/IdleTime	64u	Idle time before laser is turned off (µs).
LaserSleepMode/WakupEncoderTravel	64u	Minimum amount of encoder movement before laser turns on (mm).

Layout

Layout Child Elements

Element	Type	Description
DataSource	32s	Data source of the layout output (read-only): 0 – Top 1 – Bottom 2 – Top left 3 – Top right 4 – Top Bottom 5 – Left Right
XSpacingCount	32u	Number of points along X when data is resampled.
YSpacingCount	32u	Number of points along Y when data is resampled.
TransformedDataRegion	Region3D	Transformed data region of the layout output.
Orientation	32s	Sensor orientation: 0 – Normal (single-sensor system) / Wide (dual-sensor system) 1 – Opposite 2 – Reverse 3 – Grid
Grid	Grid	Grid representation of the multi-sensor layout.
Orientation.options	32s (CSV)	List of available orientation options.
Orientation.value	32s	Actual value used if not configurable.
MultiplexBuddyEnabled	Bool	Enables multiplexing for buddies.
MultiplexSingleEnabled	Bool	Enables multiplexing for a single sensor configuration.
MultiplexSingleExposureDuration	64f	Exposure duration in µs (currently rounded to integer when read by the sensor)
MultiplexSingleDelay	64f	Delay in µs. (Currently gets rounded up when read by the sensor.)
MultiplexSinglePeriod	64f	Period in µs. (Currently gets rounded up when read by the sensor.)
MultiplexSinglePeriod.min	64f	Minimum period in µs.

Region3D Child Elements

Element	Type	Description
X	64f	X start (mm).
Y	64f	Y start (mm).
Z	64f	Z start (mm).

Element	Type	Description
Width	64f	X extent (mm).
Length	64f	Y extent (mm).
Height	64f	Z extent (mm).
ZAngle	64f	Z Angle start (degrees).
ZAngle.used	Bool	Whether or not this property is used.

Grid Elements

Element	Type	Description
ColumnCount	32u	Column count.
ColumnCount.value	32u	Column count value.

Alignment

The Alignment element contains settings related to alignment and encoder calibration.

Alignment Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
InputTriggerEnabled	Bool	Enables digital input-triggered alignment operation.
InputTriggerEnabled.used	Bool	Whether or not this feature can be enabled. This feature is available only on some sensor models.
InputTriggerEnabled.value	Bool	Actual feature status.
Type	32s	Type of alignment operation: 0 – Stationary 1 – Moving
Type.options	32s (CSV)	List of available alignment types.
StationaryTarget	32s	Stationary alignment target: 0 – None 1 – Disk 2 – Bar 3 – Plate
StationaryTarget.options	32s (CSV)	List of available stationary alignment targets.
MovingTarget	32s	Moving alignment target: 1 – Disk 2 – Bar
MovingTarget.options	32s (CSV)	List of available moving alignment targets.
EncoderCalibrateEnabled	Bool	Enables encoder resolution calibration.
Disk	Section	See *Disk* on the next page.
Bar	Section	See *Bar* on the next page.
Plate	Section	See *Plate* on the next page.

Element	Type	Description
Polygon	Section	See *Polygon* below.

Disk

Disk Child Elements

Element	Type	Description
Diameter	64f	Disk diameter (mm).
Height	64f	Disk height (mm).

Bar

Bar Child Elements

Element	Type	Description
Width	64f	Bar width (mm).
Height	64f	Bar height (mm).
HoleCount	32u	Number of holes.
HoleCount.value	32u	Actual number of holes expected by system.
HoleCount.used	Bool	Whether the hole count with be used in the bar alignment proceudure.
HoleDistance	64f	Distance between holes (mm).
HoleDistance.used	Bool	Whether the hole distance will be used in the bar alignment procedure.
HoleDiameter	64f	Diameter of holes (mm).
HoleDiameter.used	Bool	Whether the hold diameter will be used in the bar alignment procedure.
DegreesOfFreedom	32s	Degrees of freedom (DOF) to align: 42 – 3 DOF: x, z, y angle 58 – 4 DOF: x, y, z, y angle 59 – 5 DOF: x, y, z, y angle, z angle

Plate

Plate Child Elements

Element	Type	Description
Height	64f	Plate height (mm).
HoleCount	32u	Number of holes.
RefHoleDiameter	64f	Diameter of reference hole (mm).
SecHoleDiameter	64f	Diameter of secondary hole(s) (mm).

Polygon

Polygon Child Elements

Element	Type	Description
Corners	List	Contains a list of Corners (described below).
Corners.minCount	32s	Minimum number of corners.

Polygon/Corner

Corner Child Elements

Element	Type	Description
X	64f	X Position
Y	64f	Y Position
Devices	List of 32u	List of devices this corner is assigned to.
Devices.options	List of 32u	List of valid options for this field.

Devices / Device

Devices / Device Child Elements

Element	Type	Description
@index	32u	Ordered index of devices in device list.
@role	32s	Sensor role: 0 – Main
Layout	Layout	Multiplexing bank settings.
DataSource	32s	Data source of device output (read-only): 0 – Top
XSpacingCount	32u	Number of resampled points along X (read-only).
YSpacingCount	32u	Number of resampled points along Y (read-only).
ActiveArea	Region3D	Active area. (Contains min and max attributes for each element.)
TransformedDataRegion	Region3D	Active area after transformation (read-only).
FrontCamera	Window	Front camera window (read-only).
BackCamera	Window	Back camera window (read-only).
BackCamera.used	Bool	Whether or not this field is used.
PatternSequenceType	32s	The projector pattern sequence to display when a projector equipped device is running. The following types are possible: -1 – None 0 – Default 100 – Nine Lines 101 – Focus 102 – Standard Sequence
PatternSequenceType.options	32s	List of available pattern sequence types.
PatternSequenceType.used	Bool	Whether or not this field is used.
PatternSequenceIndex	32u	The index of the pattern sequence to display. Choose the pattern that produces the best data. The indices represent Phase Pattern Sequences, followed by Stripe Pattern Sequences in reverse order. The lower indices are the higher frequency phase code patterns, and the higher indices are the lower frequency binary patterns.

Element	Type	Description
		Index 1 [Phase Pattern Sequence Image 5]: Highest frequency sinusoid. Index 2 [Phase Pattern Sequence Image 4] [...] Index 5 [Phase Pattern Sequence Image 1]: Lowest frequency sinusoid. Index 6 [Stripe Pattern Sequence Image 7]: Highest bar count. Index 7 [Stripe Pattern Sequence Image 6] [...] Index 12 [Stripe Pattern Sequence Image 1]: Lowest bar count) Index 13 [Reference Image 1]
PatternSequenceIndex.min	32u	The minimum index (inclusive)
PatternSequenceIndex.max	32u	The maximum index (inclusive)
PatternSequenceIndex.used	Bool	Whether or not the pattern sequence index should be displayed
PatternSequenceIndex	32u	The index of the pattern sequence to display.
PatternSequenceIndex.min	32u	The minimum index (inclusive).
PatternSequenceIndex.max	32u	The maximum index (inclusive).
PatternSequenceIndex.used	Bool	Whether or not the pattern sequence index should be displayed.
PatternSequenceCount	32u	Number of frames in the active sequence (read-only).
ExposureMode	32s	Exposure mode: 0 – Single exposure 1 – Multiple exposures
ExposureMode.options	32s (CSV)	List of available exposure modes.
Exposure	64f	Single exposure (μs).
Exposure.min	64f	Minimum exposure (μs).
Exposure.max	64f	Maximum exposure (μs).
Exposure.used	Bool	Whether or not this field is used.
DynamicExposureMin	64f	Dynamic exposure range minimum (μs).
DynamicExposureMax	64f	Dynamic exposure range maximum (μs).
ExposureSteps	64f (CSV)	Mutiple exposure list (μs).
ExposureSteps.countMin	32u	Minimum number of exposure steps.
ExposureSteps.countMax	32u	Maximum number of exposure steps.
IntensitySource	32s	Intensity source: 0 – Both cameras 1 – Front camera 2 – Back camera

Element	Type	Description
IntensitySource.options	32s (CSV)	List of available intensity sources.
IntensityMode	32s	Intensity Mode: 0 – Auto 1 - Preserve
IntensityMode.used	Bool	Whether intensity mode is used
ZSubsampling	32u	Subsampling factor in Z.
ZSubsampling.options	32u (CSV)	List of available subsampling factors in Z.
SpacingInterval	64f	Uniform spacing interval (mm).
SpacingInterval.min	64f	Minimum spacing interval (mm).
SpacingInterval.max	64f	Maximum spacing interval (mm).
SpacingInterval.used	Bool	Whether or not field is used.
SpacingInterval value	64f	Actual value used.
SpacingIntervalType	32s	Spacing interval type: 0 – Maximum resolution 1 – Balanced 2 – Maximum speed 3 – Custom
SpacingIntervalType.used	Bool	Whether or not this field is used.
Tracking	Section	See *Tracking Child Elements* on the next page.
Material	Section	See *Material Child Elements* on the next page.
Tracheid	Section	See *Tracheid Child Elements* on page 571.
IndependentExposures	Section	See *IndependentExposures Child Elements* on page 570
Custom	Custom	Used by specialized sensors.

Region3D Child Elements

Element	Type	Description
X	64f	X start (mm).
Y	64f	Y start (mm).
Z	64f	Z start (mm).
Width	64f	X extent (mm).
Length	64f	Y extent (mm).
Height	64f	Z extent (mm).
ZAngle	64f	Z Angle start (degrees).
ZAngle.used	Bool	Whether or not this property is used.

Window Child Elements

Element	Type	Description
X	32u	X start (pixels).
Y	32u	Y start (pixels).
Width	32u	X extent (pixels).
Height	32u	Y extent (pixels).

Layout Child Elements

Element	Type	Description
Grid	Grid	Layout grid information.
MultiplexingBank	32u	Multiplexing bank ID
MultiplexingBank.used	32u	Whether or not this field can be specified
MultiplexingBank.value	32u	Actual value used by system

Grid Child Elements

Element	Type	Description
@used	Bool	Whether or not this section is used.
Row	32s	Device row position in grid layout.
Row.value	32s	Value in use by the sensor, useful for determining value when used is false.
Column	32s	Device column position in grid layout.
Column.value	32s	Value in use by the sensor, useful for determining value when used is false.
Direction	32s	Sensor orientation direction.
Direction.value	32s	Value in use by the sensor, useful for determining value when used is false.

Tracking is only available on Gocator 2300 and 2400 series sensors.

Tracking Child Elements

Element	Type	Description
Enabled	Bool	Enables tracking.
Enabled.used	Bool	Whether or not this field is used.
SearchThreshold	64f	Percentage of spots that must be found to remain in track.
Height	64f	Tracking window height (mm).
Height.min	64f	Minimum tracking window height (mm).
Height.max	64f	Maximum tracking window height (mm).

Material Child Elements

Element	Type	Description
Type	32s	Type of Material settings to use. 0 – Custom

Element	Type	Description
		1 – Diffuse 3 – Reflective
Type.used	Bool	Determines if the setting's value is currently used.
Type.value	32s	Value in use by the sensor, useful for determining value when used is false.
Type.options	32u (CSV)	List of available material types.
SpotThreshold	32s	Spot detection threshold.
SpotThreshold.min	32s	The minimum spot detection threshold possible.
SpotThreshold.max	32s	The maximum spot detection threshold possible.
SpotThreshold.used	Bool	Determines if the setting's value is currently used.
SpotThreshold.value	32s	Value in use by the sensor, useful for determining value when used is false.
SpotThreshold.readonly	Bool	Whether or not property can be modified. If set, the value should be considered read-only by the client. Only has meaning if "used" is also set.
SpotWidthMax	32s	Spot detection maximum width.
SpotWidthMax.used	Bool	Determines if the setting's value is currently used.
SpotWidthMax.value	32s	Value in use by the sensor, useful for determining value when used is false.
SpotWidthMax.min	32s	Minimum allowed spot detection maximum value.
SpotWidthMax.max	32s	Maximum allowed spot detection maximum value.
SpotSelectionType	32s	Spot selection type 0 – Best. Picks the strongest spot in a given column. 1 – Top. Picks the spot which is most Top/Left on the imager 2 – Bottom. Picks the spot which is most Bottom/Right on the imager 3 – None. All spots are available. This option may not be available in some configurations. 4 – Continuity. Picks the most continuous spot.
SpotSelectionType.used	Bool	Determines if the setting's value is currently used.
SpotSelectionType.value	32s	Value in use by the sensor, useful for determining value when used is false.
SpotSelectionType.options	32s (CSV)	List of available spot selection types.
SpotWidthMin	32s	Spot detection minimum width.
SpotWidthMin.used	32s	Determines if the setting's value is currently used and to be displayed.
SpotWidthMin.value	Bool	Value in use by the sensor, useful for determining value

Element	Type	Description
		when "used" is false.
SpotWidthMin.min	32s	Minimum allowed spot detection minimum value.
SpotWidthMin.max	32s	Maximum allowed spot detection minimum value.
SpotWidthMin.readonly	Bool	Whether or not property can be modified. If set, the value should be considered read-only by the client. Only has meaning if “used” is also set.
WidthThreshold	32s	Spot detection width threshold.
WidthThreshold.used	Bool	Determines if the setting’s value is currently used and to be displayed.
WidthThreshold.value	32s	Value in use by the sensor, useful for determining value when "used" is false.
WidthThreshold.min	32s	Minimum allowed spot detection width threshold value.
WidthThreshold.max	32s	Maximum allowed spot detection width threshold value.
WidthThreshold.readonly	Bool	Whether or not property can be modified. If set, the value should be considered read-only by the client. Only has meaning if “used” is also set.
SpotSumMin	32s	Spot detection minimum sum.
SpotSumMin.used	Bool	Determines if the setting’s value is currently used and to be displayed.
SpotSumMin.value	32s	Value in use by the sensor, useful for determining value when "used" is false.
SpotSumMin.min	32s	Minimum allowed spot detection sum minimum value.
SpotSumMin.max	32s	Maximum allowed spot detection sum minimum value.
SpotSumMin.readonly	Bool	Whether or not property can be modified. If set, the value should be considered read-only by the client. Only has meaning if “used” is also set.
SobelEdgeWindow	32s	Sobel spot detection edge window size.
SobelEdgeWindow.used	Bool	Determines if the setting’s value is currently used and to be displayed.
SobelEdgeWindow.value	32s	Value in use by the sensor, useful for determining value when "used" is false.
SobelEdgeWindow min	32s	Minimum allowed Sobel spot detection edge window value.
SobelEdgeWindow max	32s	Maximum allowed Sobel spot detection edge window value.
SobelEdgeWindow.readonly	Bool	Whether or not property can be modified. If set, the value should be considered read-only by the client. Only has

Element	Type	Description
		meaning if “used” is also set.
CameraGainAnalog	64f	Analog camera gain factor.
CameraGainAnalog.used	Bool	Determines if the setting’s value is currently used.
CameraGainAnalog.value	64f	Value in use by the sensor, useful for determining value when used is false.
CameraGainAnalog.min	64f	Minimum value.
CameraGainAnalog.max	64f	Maximum value.
CameraGainDigital	64f	Digital camera gain factor.
CameraGainDigital.used	Bool	Determines if the setting’s value is currently used.
CameraGainDigital.value	64f	Value in use by the sensor, useful for determining value when used is false.
CameraGainDigital.min	64f	Minimum value.
CameraGainDigital.max	64f	Maximum value.
DynamicSensitivity	64f	Dynamic exposure control sensitivity factor. This can be used to scale the control setpoint.
DynamicSensitivity.used	Bool	Determines if the setting’s value is currently used.
DynamicSensitivity.value	64f	Value in use by the sensor, useful for determining value when used is false.
DynamicSensitivity.min	64f	Minimum value.
DynamicSensitivity.max	64f	Maximum value.
DynamicThreshold	32s	Dynamic exposure control threshold. If the detected number of spots is fewer than this number, the exposure will be increased.
DynamicThreshold.used	Bool	Determines if the setting’s value is currently used.
DynamicThreshold.value	32s	Value in use by the sensor, useful for determining value when used is false.
DynamicThreshold.min	32s	Minimum value.
DynamicThreshold.max	32s	Maximum value.
SensitivityCompensationEnabled	Bool	Sensitivity compensation toggle. Used in determining analog and digital gain, along with exposure scale.
SensitivityCompensationEnabled.used	Bool	Determines if the setting’s value is currently used.
SensitivityCompensationEnabled.value	Bool	Value in use by the sensor, useful for determining value when used is false.
GammaType	32s	Gamma type.
GammaType used	Bool	Determines if the setting’s value is currently used.
GammaType value	32s	Value in use by the sensor. Useful for determining value when used is false.
SpotContinuitySorting	Section	See *SpotContinuitySorting Child Elements* on the next page.

Element	Type	Description
SurfaceEncoding	32s	Surface encoding type: 0 – Standard 1 – Interreflection (advanced use only)
SurfaceEncoding.used	Bool	Determines if the setting’s value is currently used.
SurfaceEncoding.value	Bool	Value in use by the sensor, useful for determining value when used is false.
SurfaceEncoding.readonly	Bool	Whether or not property can be modified. If set, the value should be considered read-only by the client. Only has meaning if “used” is also set.
SurfacePhaseFilter	32s	Surface phase filter (correction type) 0 – None 1 – Reflective 2 - Translucent
SurfacePhaseFilter.used	Bool	Determines if the setting’s value is currently used.
SurfacePhaseFilter.value	Bool	Value in use by the sensor, useful for determining value when used is false.
ContrastThreshold	32s	Contrast detection threshold.
ContrastThreshold.min	32s	The minimum contrast detection threshold possible.
ContrastThreshold.max	32s	The maximum contrast detection threshold possible.
ContrastThreshold.used	Bool	Determines if the setting’s value is currently used.
ContrastThreshold.value	32s	Value in use by the sensor, useful for determining value when used is false.
ContrastThreshold.readonly	Bool	Whether or not property can be modified. If set, the value should be considered read-only by the client. Only has meaning if “used” is also set.

SpotContinuitySorting Child Elements

Element	Type	Description
MinimumSegmentSize	32u	Smallest continuous segment considered in continuity sorting.
SearchWindow/X	32u	X component of continuity sorting search window size.
SearchWindow/Y	32u	Y component of continuity sorting search window size.

IndependentExposures settings are only supported by 3x00 series sensors.

IndependentExposures Child Elements

Element	Type	Description
@used	Bool	Whether this field is used
Enabled	Bool	Whether to allow using separate exposure values for each camera

Element	Type	Description
FrontCameraExposure	64f	The exposure value to use for the front camera
FrontCameraExposure.min	64f	The minimum exposure value possible for front camera
FrontCameraExposure.max	64f	The maximum exposure value possible for back camera
BackCameraExposure	64f	The exposure value to use for the front camera
BackCameraExposure.min	64f	The minimum exposure value possible for front camera
BackCameraExposure.max	64f	The maximum exposure value possible for back camera

Tracheid settings are only supported by Gocator 200 series multi-point sensors.

Tracheid Child Elements

Element	Type	Description
@used	Bool	Whether this field is used
TracheidExposureEnabled	Bool	Whether to use a unique exposure for tracheid capture
TracheidExposure	64f	The exposure value to use for tracheid measurements
TracheidExposure.min	64f	The minimum exposure value possible tracheid measurements
TracheidExposure.max	64f	The maximum exposure value possible for tracheid measurements
Camera0Threshold	32u	The tracheid threshold for camera 0
Camera1Threshold	32u	The tracheid threshold for camera 1

SurfaceGeneration

The SurfaceGeneration element contains settings related to surface generation.

This element is used by laser profile sensors.

SurfaceGeneration Child Elements

Element	Type	Description
Type	32s	Surface generation type: 0 – Continuous 1 – Fixed length 2 – Variable length 3 – Rotational
Type.options	32s (CSV)	List of available generation types
Type.value	32s	Value in use by the sensor
FixedLength	Section	See *FixedLength* on the next page.
VariableLength	Section	See *VariableLength* on the next page.
Rotational	Section	See *Rotational* on the next page.

FixedLength

FixedLength Child Elements

Element	Type	Description
StartTrigger	32s	Start trigger condition: 0 – Sequential 1 – Digital input 2 – Software triggered
ExternalInputIndex	32s	Index of external input when Source (above) is set to 1 – Digital Input and connected to a Master. 0 – first digital input 1 – second digital input 2 – third digital input 3 – fourth digital input
ExternalInputIndex.options	32s (CSV)	List of available external input indices.
ExternalInputIndex.used	Bool	Is the external input index in use.
Length	64f	Surface length (mm).
Length.min	64f	Minimum surface length (mm).
Length.max	64f	Maximum surface length (mm).

VariableLength

VariableLength Child Elements

Element	Type	Description
MaxLength	64f	Maximum surface length (mm).
MaxLength.min	64f	Minimum value for maximum surface length (mm).
MaxLength.max	64f	Maximum value for maximum surface length (mm).

Rotational

Rotational Child Elements

Element	Type	Description
Circumference	64f	Circumference (mm).
Circumference.min	64f	Minimum circumference (mm).
Circumference.max	64f	Maximum circumference (mm).

SurfaceSections

SurfaceSections Child Elements

Element	Type	Description
@used	Bool	Whether surface sectioning is enabled.
@xMin	64f	The minimum valid X value to be used for section definition.
@xMax	64f	The maximum valid X value to be used for section definition.

Element	Type	Description
@yMin	64f	The minimum valid Y value to be used for section definition.
@yMax	64f	The maximum valid Y value to be used for section definition.
Section	Collection	A series of Section elements.

Section Child Elements

Element	Type	Description
@id	32s	The ID assigned to the surface section.
@name	String	The name associated with the surface section.
StartPoint	Point64f	The beginning point of the surface section.
EndPoint	Point64f	The end point of the surface section.
CustomSpacingIntervalEnabled	Bool	Indicates whether a user specified custom spacing interval is to be used for the resulting section.
SpacingInterval	64f	The user specified spacing interval.
SpacingInterval.min	64f	The spacing interval limit minimum.
SpacingInterval.max	64f	The spacing interval limit maximum.
SpacingInterval.value	64f	The current spacing interval used by the system.

ProfileGeneration

The ProfileGeneration element contains settings related to profile generation.

This element is used by laser displacement sensors.

ProfileGeneration Child Elements

Element	Type	Description
Type	32s	Profile generation type: 0 – Continuous 1 – Fixed length 2 – Variable length 3 – Rotational
Type.options	32s (CSV)	List of available generation types
Type.value	32s	Value in use by the sensor
FixedLength	Section	See *FixedLength* below.
VariableLength	Section	See *VariableLength* on the next page.
Rotational	Section	See *Rotational* on the next page.

FixedLength

FixedLength Child Elements

Element	Type	Description
StartTrigger	32s	Start trigger condition: 0 – Sequential

Element	Type	Description
		1 – Digital input 2 – Software triggered
ExternalInputIndex	32s	Index of external input when Source (above) is set to 1 – Digital Input and connected to a Master. 0 – first digital input 1 – second digital input 2 – third digital input 3 – fourth digital input
ExternalInputIndex.options	32s (CSV)	List of available external input indices.
ExternalInputIndex.used	Bool	Is the external input index in use.
Length	64f	Profile length (mm).
Length.min	64f	Minimum profile length (mm).
Length.max	64f	Maximum profile length (mm).

VariableLength

VariableLength Child Elements

Element	Type	Description
MaxLength	64f	Maximum surface length (mm).
MaxLength.min	64f	Minimum value for maximum profile length (mm).
MaxLength.max	64f	Maximum value for maximum profile length (mm).

Rotational

Rotational Child Elements

Element	Type	Description
Circumference	64f	Circumference (mm).
Circumference.min	64f	Minimum circumference (mm).
Circumference.max	64f	Maximum circumference (mm).

PartDetection

PartDetection Child Elements

Element	Type	Description
Enabled	Bool	Enables part detection.
Enabled.used	Bool	Whether or not this field is used.
Enabled value	Bool	Actual value used if not configurable.
MinArea	64f	Minimum area (mm^2).
MinArea.min	64f	Minimum value of minimum area.
MinArea.max	64f	Maximum value of minimum area.

Element	Type	Description
MinArea.used	Bool	Whether or not this field is used.
GapWidth	64f	Gap width (mm).
GapWidth.min	64f	Minimum gap width (mm).
GapWidth.max	64f	Maximum gap width (mm).
GapWidth.used	Bool	Whether or not this field is used.
GapLength	64f	Gap length (mm).
GapLength.min	64f	Minimum gap length (mm).
GapLength.max	64f	Maximum gap length (mm).
GapLength.used	Bool	Whether or not this field is used.
PaddingWidth	64f	Padding width (mm).
PaddingWidth.min	64f	Minimum padding width (mm).
PaddingWidth.max	64f	Maximum padding width (mm).
PaddingWidth.used	Bool	Whether or not this field is used.
PaddingLength	64f	Padding length (mm).
PaddingLength.min	64f	Minimum padding length (mm).
PaddingLength.max	64f	Maximum padding length (mm).
PaddingLength.used	Bool	Whether or not this field is used.
MinLength	64f	Minimum length (mm).
MinLength.min	64f	Minimum value of minimum length (mm).
MinLength.max	64f	Maximum value of minimum length (mm).
MinLength.used	Bool	Whether or not this field is used.
MaxLength	64f	Maximum length (mm).
MaxLength.min	64f	Minimum value of maximum length (mm).
MaxLength.max	64f	Maximum value of maximum length (mm).
MaxLength.used	Bool	Whether or not this field is used.
Threshold	64f	Height threshold (mm).
Threshold.min	64f	Minimum height threshold (mm).
Threshold.max	64f	Maximum height threshold (mm).
ThresholdDirection	32u	Threshold direction: 0 – Above 1 – Below
FrameOfReference	32s	Part frame of reference: 0 – Sensor 1 – Scan 2 – Part
FrameOfReference.used	Bool	Whether or not this field is used.

Element	Type	Description
FrameOfReference.value	32s	Actual value.
IncludeSinglePointsEnabled	Bool	Enables preservation of single data points in Top+Bottom layout
IncludeSinglePointsEnabled.used	Bool	Whether or nto this field is available to be modified
EdgeFiltering	Section	See *EdgeFiltering* below.

EdgeFiltering

EdgeFiltering Child Elements

Element	Type	Description
@used	Bool	Whether or not this section is used.
Enabled	Bool	Enables edge filtering.
PreserveInteriorEnabled	Bool	Enables preservation of interior.
ElementWidth	64f	Element width (mm).
ElementWidth.min	64f	Minimum element width (mm).
ElementWidth.max	64f	Maximum element width (mm).
ElementLength	64f	Element length (mm).
ElementLength.min	64f	Minimum element length (mm).
ElementLength.max	64f	Maximum element length (mm).

PartMatching

The PartMatching element contains settings related to part matching.

PartMatching Child Elements

Element	Type	Description
Enabled	Bool	Enables part matching.
Enabled.used	Bool	Whether or not this field is used.
MatchAlgo	32s	Match algorithm. 0 – Edge points 1 – Bounding Box 2 – Ellipse
Edge	Section	See *Edge* below.
BoundingBox	Section	See *BoundingBox* on the next page.
Ellipse	Section	See *Ellipse* on the next page.

Edge

Edge Child Elements

Element	Type	Description
ModelName	String	Name of the part model to use. Does not include the .mdl extension.
Acceptance/Quality/Min	64f	Minimum quality value for a match.

BoundingBox

BoundingBox Child Elements

Element	Type	Description
ZAngle	64f	Z rotation to apply to bounding box (degrees).
AsymmetryDetectionType	32s	Determine whether to use asymmetry detection and, if enabled, which dimension is the basis of detection. The possible values are: 0 – None 1 – Length 2 - Width
Acceptance/Width/Min	64f	Minimum width (mm).
Acceptance/Width/Max	64f	Maximum width (mm).
Acceptance/Width/Tolerance	64f	Width acceptance tolerance value
Acceptance/Width/Tolerance.deprecated	Bool	Whether this tolerance field is deprecated
Acceptance/Length/Min	64f	Minimum length (mm).
Acceptance/Length/Max	64f	Maximum length (mm).
Acceptance/Length/Tolerance	64f	Length acceptance tolerance value
Acceptance/Length/Tolerance.deprecated	Bool	Whether this tolerance field is deprecated
X	64f	X value
X.deprecated	Bool	Whether this X field is deprecated
Y	64f	Y value
Y.deprecated	Bool	Whether this Y field is deprecated
Width	64f	Width value
Width.deprecated	Bool	Whether this width field is deprecated
Length	64f	Length value
Length.deprecated	Bool	Whether this length field is deprecated

Ellipse

Ellipse Child Elements

Element	Type	Description
ZAngle	64f	Z rotation to apply to ellipse (degrees).
AsymmetryDetectionType	32s	Determine whether to use asymmetry detection and, if enabled, which dimension is the basis of detection. The possible values are: 0 – None 1 – Major 2 - Minor
Acceptance/Major/Min	64f	Minimum major length (mm).

Element	Type	Description
Acceptance/Major/Max	64f	Maximum major length (mm).
Acceptance/Major/Tolerance	64f	Major acceptance tolerance value
Acceptance/Major/Tolerance.deprecated	Bool	Whether this tolerance field is deprecated
Acceptance/Minor/Min	64f	Minimum minor length (mm).
Acceptance/Minor/Max	64f	Maximum minor length (mm).
Acceptance/Minor/Tolerance	64f	Minor acceptance tolerance value
Acceptance/Minor/Tolerance.deprecated	Bool	Whether this tolerance field is deprecated
X	64f	X value
X.deprecated	Bool	Whether this X field is deprecated
Y	64f	Y value
Y.deprecated	Bool	Whether this Y field is deprecated
Width	64f	Width value
Width.deprecated	Bool	Whether this width field is deprecated
Length	64f	Length value
Length.deprecated	Bool	Whether this length field is deprecated

Replay

Contains settings related to recording filtering.

RecordingFiltering

RecordingFiltering Child Elements

Element	Type	Description
ConditionCombineType	32s	0 – Any: If any enabled condition is satisfied, the current frame is recorded. 1 – All: All enabled conditions must be satisfied for the current frame to be recorded.
Conditions	Collection	A collection of AnyMeasurement, AnyData, or Measurement conditions.

Conditions/AnyMeasurement

Conditions/AnyMeasurement Elements

Element	Type	Description
Enabled	Bool	Indicates whether the condition is enabled.
Result	32s	The measurement decision criteria to be included in the filter. Possible values are: 0 – Pass 1 – Fail 2 – Valid 3 – Invalid

Conditions/AnyData

Conditions/AnyData Elements

Element	Type	Description
Enabled	Bool	Indicates whether the condition is enabled.
RangeCountCase	32s	The case under which to record data: 0 – Range count at or above threshold of valid data points. 1 – Range count below threshold.
RangeCountThreshold	32u	The threshold for the number of range points that are valid.

Conditions/Measurement

Conditions/Measurement Elements

Element	Type	Description
Enabled	Bool	Indicates whether the condition is enabled.
Result	32s	The measurement decision criteria for the selected ID to be included in the filter. Possible values are: 0 – Pass 1 – Fail 2 – Valid 3 – Invalid
Ids	32s	The ID of the measurement to filter.

Streams/Stream (Read-only)

Streams/Stream Child Elements

Element	Type	Description
Step	32s	The data step of the stream being described. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Id	32u	The stream ID.
CadenceId	32u	Represents a stage in the data processing pipeline. The greater the number, the farther removed from the initial acquisition stage. One of the following: 0 – Primary 1 – Auxiliary 10 - Diagnostic
DataType	32s	The stream data type 0 – None 4 – Uniform Profile

Element	Type	Description
		16 – Uniform Surface
ColorEncoding	32s	The color encoding type. Only appears for Video stream steps (1). 0 – None 1 – Bayer BGGR 2 – Bayer GBRG 3 – Bayer RGGB 4 – Bayer GRBG
IntensityEnabled	Bool	Whether the stream includes intensity data
Sources	Collection	A collection of Source elements as described below.

Source Child Elements

Element	Type	Description
Id	32s	The ID of the data source. Possible values are: 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right 4 – Top Bottom 5 – Left Right 100 to 131 – G2 buddy sensor device indices for configurations with 2 to 31 buddy G2 sensors to identify a particular sensor's scan data. Main sensor is 100. First buddied sensor is 101. Second buddied sensor is 102 and so on.
Capability	32s	The capability of the data stream source. Possible values are: 0 – Full 1 – Diagnostic only 2 - Virtual
Region	Region3d	The region of the given stream source.
AdditionalRegions	Collection	Collection of additional regions (for example, for the second camera).
AdditionalRegions/Region	Region3d	Additional regions.

ToolOptions

The ToolOptions element contains a list of available tool types, their measurements, features, and data output types, and settings for related information.

ToolOptions Child Elements

Element	Type	Description
<Tool Names>	Collection	A collection of tool name elements. An element for each tool type is present.

Tool Name Child Elements

Element	Type	Description
@displayName	String	Display name of the tool.
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
MeasurementOptions	Collection	See *MeasurementOptions* below.
FeatureOptions	Collection	See *FeatureOptions* below.
StreamOptions	Collection	See *StreamOptions* on the next page.
ToolDataOutputOptions	Collection	See *ToolDataOutputOptions* on the next page.
DefinedSourcesOptions	Collection	See *DefinedSouresOptions* on page 583.

MeasurementOptions

MeasurementOptions Child Elements

Element	Type	Description
<Measurement Names>	Collection	A collection of measurement name elements. An element for each measurement is present.

<Measurement Name> Child Elements

Element	Type	Description
@displayName	String	Display name of the tool.
@minCount	32u	Minimum number of instances in a tool.
@maxCount	32u	Maximum number of instances in a tool.

FeatureOptions

FeatureOptions Child Elements

Element	Type	Description
<Feature Names>	Collection	A collection of feature name elements. An element for each measurement is present.

<Feature Name> Child Elements

Element	Type	Description
@displayName	String	Display name of the feature.
@minCount	32u	Minimum number of instances in a tool.
@maxCount	32u	Maximum number of instances in a tool.
@dataType	String	The data type of the feature. One of: – PointFeature – LineFeature

Element	Type	Description
		– CircleFeature – PlaneFeature

StreamOptions

StreamOptions Child Elements

Element	Type	Description
@step	32s	The data step of the stream being described. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
@ids	CSV	A list representing the available IDs associated with the given step.

ToolDataOutputOptions

ToolDataOutputOptions Child Elements

Element	Type	Description
@displayName	String	Display name of the tool.
@dataType	32s	The data type of this data output from the tool. Possible values are: 1 – None 2 – Range 3 – Uniform (Resampled) Profile 4 – Profile Point Cloud (Unresampled Profile) 5 – Uniform (Resampled) Surface 6 – Surface Point Cloud (Unresampled Surface) 7 – Reserved 8 – Video 9 – Tracheid 10 – Measurement 0x201 – Feature Point 0x202 – Feature Line 0x203 – Feature Circle 0x204 – Feature Plane 0x80000000 – 0xFFFFFFFF – Generic Data
@minCount	32u	Minimum number of instances in a tool.
@maxCount	32u	Maximum number of instances in a tool.

DefinedSouresOptions

DefinedSourcesOptions Child Elements

Element	Type	Description
@options	32s	Defines all the sensor positions that can be an input data source to this tool. The allowable sources are specified during tool definition time.

Tools

The Tools element contains measurement tools. The following sections describe each tool and its available measurements.

Tools Child Elements

Element	Type	Description
@options	String (CSV)	A list of the tools available in the currently selected scan mode.
<ToolType>	Section	An element for each added tool.

Profile Types

The following types are used by various measurement tools.

ProfileFeature

An element of type ProfileFeature defines the settings for detecting a feature within an area of interest.

ProfileFeature Child Elements

Element	Type	Description
Type	32s	Determine how the feature is detected within the area: 0 – Max Z 1 – Min Z 2 – Max X 3 – Min X 4 – Corner 5 – Average 6 – Rising Edge 7 – Falling Edge 8 – Any Edge 9 – Top Corner 10 – Bottom Corner 11 – Left Corner 12 – Right Corner 13 – Median
RegionEnabled	Bool	Indicates whether feature detection applies to the defined Region or to the entire active area.
Region	ProfileRegion2D	Element for feature detection area.

ProfileLine

An element of type ProfileLine defines measurement areas used to calculate a line.

ProfileLine Child Elements

Element	Type	Description
RegionCount	32s	Count of the regions.
Regions	(Collection)	The regions used to calculate a line. Contains one or two Region elements of type ProfileRegion2D, with RegionEnabled fields for each.

ProfileRegion2d

An element of type ProfileRegion2d defines a rectangular area of interest.

ProfileRegion2d Child Elements

Element	Type	Description
X	64f	Setting for profile region X position (mm).
Z	64f	Setting for profile region Z position (mm).
Width	64f	Setting for profile region width (mm).
Height	64f	Setting for profile region height (mm).

Surface Types

The following types are used by the various measurement tools.

Region3D

An element of type Region3D defines a rectangular area of interest in 3D.

Region3D Child Elements

Element	Type	Description
X	64f	Volume X position (mm).
Y	64f	Volume Y position (mm).
Z	64f	Volume Z position (mm).
Width	64f	Volume width (mm).
Length	64f	Volume length (mm).
Height	64f	Volume height (mm).

SurfaceFeature

An element of type SurfaceFeature defines the settings for detecting a feature within an area of interest.

SurfaceFeature Child Elements

Element	Type	Description
Type	32s	Setting to determine how the feature is detected within the area: 0 – Average (formerly Centroid 2d) 1 – Centroid (formerly Centroid 3d) 2 – X Max

Element	Type	Description
		3 – X Min 4 – Y Max 5 – Y Min 6 – Z Max 7 – Z Min 8 – Median
RegionEnabled	Boolean	Setting to enable/disable region: 0 – Disable 1 – Enable
Region	Region3D	Element for feature detection volume.

SurfaceRegion2d

An element of type SurfaceRegion2d defines a rectangular area of interest on the X-Y plane.

SurfaceRegion2d Child Elements

Element	Type	Description
X	64f	Setting for surface region X position (mm).
Y	64f	Setting for surface region Y position (mm).
Width	64f	Setting for region width (mm).
Length	64f	Setting for region length (mm).

Geometric Feature Types

The Geometric Feature type is used by various measurement tools.

Feature Child Elements

Element	Type	Description
@id	32s	The identifier of the geometric feature. -1 if unassigned.
@dataType	String	The data type of the feature. One of: – PointFeature – LineFeature
@type	String	Type name of feature.
Name	String	The display name of the feature.
Enabled	Bool	Whether the given feature output is enabled.
Pinned	Boolean	Whether the feature is pinned to main renderer.
Parameters	Collection	Collection of GdkParam elements.

Parameter Types

The following types are used by internal and custom (user-created) GDK-based tools.

For the list of attributes of these types, see *GDK Parameter Child Elements* on page 587.

GDK Parameter Bool Type

Element	Type	Description
	Bool	Boolean value of parameter.

GDK Parameter Int Type

Element	Type	Description
	32s	Integer value of parameter of integer type.

GDK Parameter Float Type

Element	Type	Description
	64f	Floating point value of parameter.

GDK Parameter String Type

Element	Type	Description
	String	String value of parameter.

GDK Parameter Profile Region Type

Element	Type	Description
X	64f	X value of region.
Z	64f	Z value of region.
Width	64f	Width value of region.
Height	64f	Height value of region.

GDK Parameter Surface Region 2D Type

Element	Type	Description
X	64f	X value of region.
X	64f	X value of region.
Y	64f	Y value of region.
Width	64f	Width value of region.
Length	64f	Length value of region.

GDK Parameter Surface Region 3D Type

Element	Type	Description
X	64f	X value of region.
Y	64f	Y value of region.
Z	64f	Z value of region.
Width	64f	Width value of region.
Length	64f	Length value of region.
Height	64f	Height value of region.
ZAngle	64f	ZAngle value of region.

GDK Parameter Geometric Feature Type

Element	Type	Description
	32s	Geometric feature Id for parameter.

GDK Parameter Child Elements

Element	Type	Description
@label	String	Parameter label.
@type	String	Type of parameter. It is one of the following (see tables below for elements found in each type): - Bool - Int - Float - ProfileRegion - SurfaceRegion2d - SurfaceRegion3d - GeometricFeature - DataInput
@units	String	Parameter units name.
@options	Variant (CSV)	Options available for this parameter.
@optionNames	String (CSV)	Names
@used	String (CSV)	Parameter currently in use if true. Optional (defaults to true if not explicitly set)
@dataTypes	k32s	For DataInput parameters, it lists all the data types accepted by this parameter.

ProfileArea

A ProfileArea element defines settings for a profile area tool and one or more of its measurements.

ProfileArea Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *ProfileArea* above.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.

Element	Type	Description
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
Type	Boolean	Area to measure: 0 – Object (convex shape above the baseline) 1 – Clearance (concave shape below the baseline)
Type.used	Boolean	Whether or not field is used.
Baseline	Boolean	Baseline type: 0 – X-axis 1 – Line
Baseline.used	Boolean	Whether or not field is used.
RegionEnabled	Boolean	If enabled, the defined region is used for measurements. Otherwise, the full active area is used.
Region	ProfileRegion2d	Measurement region.
Line	ProfileLine	Line definition when Baseline is set to Line.
Measurements\Area	Area tool measurement	Area measurement.
Measurements\CentroidX	Area tool measurement	CentroidX measurement.
Measurements\CentroidZ	Area tool measurement	CentroidZ measurement.
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature.

Area Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable

Element	Type	Description
		1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

ProfileBoundingBox

A ProfileBoundingBox element defines settings for a profile bounding box tool and one or more of its measurements.

ProfileBoundingBox Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *ProfileBoundingBox* above.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.

Element	Type	Description
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Measurement region.
Measurements\X	Bounding Box tool measurement	X measurement.
Measurements\Z	Bounding Box tool measurement	Z measurement.
Measurements\Width	Bounding Box tool measurement	Width measurement.
Measurements\Height	Bounding Box tool measurement	Height measurement.
Measurements\GlobalX	Bounding Box tool measurement	GlobalX measurement
Measurements\GlobalY	Bounding Box tool measurement	GlobalY measurement
Measurements\GlobalAngle	Bounding Box tool measurement	GlobalAngle measurement
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature.
Features\CornerPoint	GeometricFeature	CornerPoint PointFeature.

Bounding Box Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.

Element	Type	Description
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

ProfileCircle

A ProfileCircle element defines settings for a profile circle tool and one or more of its measurements.

ProfileCircle Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *ProfileCircle* above.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Measurement region.
Measurements\X	Circle tool measurement	X measurement.
Measurements\Z	Circle tool measurement	Z measurement.
Measurements\Radius	Circle tool measurement	Radius measurement.
Measurements\StdDev	CircleMeasurement	Standard deviation measurement

Element	Type	Description
Measurements\MinError	CircleMeasurement	Minimum error measurement
Measurements\MinErrorX	CircleMeasurement	Minimum error X measurement
Measurements\MinErrorZ	CircleMeasurement	Minimum error Z measurement
Measurements\MaxError	CircleMeasurement	Maximum error measurement
Measurements\MaxErrorX	CircleMeasurement	Maximum error X measurement
Measurements\MaxErrorZ	CircleMeasurement	Maximum error Z measurement
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature.

Circle Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

ProfileDimension

A ProfileDimension element defines settings for a profile dimension tool and one or more of its measurements.

ProfileDimension Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
RefFeature	ProfileFeature	Reference measurement region.
Feature	ProfileFeature	Measurement region.
Measurements\Width	Dimension tool measurement	Width measurement.
Measurements\Height	Dimension tool measurement	Height measurement.
Measurements\Distance	Dimension tool measurement	Distance measurement.
Measurements\CenterX	Dimension tool measurement	CenterX measurement.
Measurements\CenterZ	Dimension tool measurement	CenterZ measurement.

Dimension Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).

Element	Type	Description
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
Absolute *(Width and Height measurements only)*	Boolean	Setting for selecting absolute or signed result: 0 – Signed 1 – Absolute

ProfileGroove

A ProfileGroove element defines settings for a profile groove tool and one or more of its measurements.

The profile groove tool is dynamic, meaning that it can contain multiple measurements of the same type in the Measurements element.

ProfileGroove Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.

Element	Type	Description
Features	Collection	Not used.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
Shape	32s	Shape: 0 – U-shape 1 – V-shape 2 – Open
MinDepth	64f	Minimum depth.
MinWidth	64f	Minimum width.
MaxWidth	64f	Maximum width.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Measurement region.
Measurements\X	Groove tool measurement	X measurement.
Measurements\Z	Groove tool measurement	Z measurement.
Measurements\Width	Groove tool measurement	Width measurement.
Measurements\Depth	Groove tool measurement	Depth measurement.

Groove Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable

Element	Type	Description
		1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
SelectType	32s	Method of selecting a groove when multiple grooves are found: 0 – Max depth 1 – Ordinal, from left 2 – Ordinal, from right
SelectIndex	32s	Index when SelectType is set to 1 or 2.
Location *(X and Z measurements only)*	32s	Setting for groove location to return from: 0 – Bottom 1 – Left corner 2 – Right corner

ProfileIntersect

A ProfileIntersect element defines settings for a profile intersect tool and one or more of its measurements.

ProfileIntersect Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool

Element	Type	Description
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *ProfileIntersect* on the previous page.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
RefType	32s	Reference line type: 0 – Fit 1 – X Axis
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
RefLine	ProfileLine	Definition of reference line. Ignored if RefType is not 0.
Line	ProfileLine	Definition of line.
Measurements\X	Intersect tool measurement	X measurement.
Measurements\Z	Intersect tool measurement	Z measurement.
Measurements\Angle	Intersect tool measurement	Angle measurement.
Features\IntersectPoint	GeometricFeature	IntersectPoint PointFeature.
Features\Line	GeometricFeature	Line LineFeature.
Features\BaseLine	GeometricFeature	BaseLine LineFeature.

Intersect Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable

Element	Type	Description
		1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
Absolute *(Angle measurement only)*	Boolean	Setting for selecting the angle range: 0 – A range of -90 to 90 degrees is used. 1 – A range of 0 to 180 degrees is used.

ProfileLine

A ProfileLine element defines settings for a profile line tool and one or more of its measurements.

ProfileLine Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *ProfileLine* above.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.

Element	Type	Description
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Measurement region.
FittingRegions	ProfileLine	ProfileLine describing up to 2 regions to fit to.
FittingRegionsEnabled	Bool	Whether the fitting regions are enabled.
Measurements\StdDev	Line tool measurement	StdDev measurement.
Measurements\MaxError	Line tool measurement	MaxError measurement.
Measurements\MinError	Line tool measurement	MinError measurement.
Measurements\Percentile	Line tool measurement	Percentile measurement.
Measurements\Offset	Line tool measurement	Offset measurement.
Measurements\Angle	Line tool measurement	Angle measurement.
Measurements\MinErrorX	Line tool measurement	Minimum Error in Z measurement.
Measurements\MinErrorZ	Line tool measurement	Minimum Error in Z measurement.
Measurements\MaxErrorX	Line tool measurement	Maximum Error in X measurement.
Measurements\MaxErrorZ	Line tool measurement	Maximum Error in Z measurement.
Features\Line	GeometricFeature	Line LineFeature.
Features\ErrorMinPoint	GeometricFeature	ErrorMinPoint PointFeature.
Features\ErrorMaxPoint	GeometricFeature	ErrorMaxPoint PointFeature.

Line Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable

Element	Type	Description
		1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
Percent *(Percentile measurement only)*	64f	Error percentile.

ProfilePanel

A ProfilePanel element defines settings for a profile panel tool and one or more of its measurements.

ProfilePanel Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range

Element	Type	Description
		3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
RefSide	32s	Setting for reference side to use.
MaxGapWidth	64f	Setting for maximum gap width (mm).
LeftEdge	ProfilePanelEdge	Element for left edge configuration.
RightEdge	ProfilePanelEdge	Element for right edge configuration.
Measurements\Gap	Gap/Flush measurement	Gap measurement.
Measurements\Flush	Gap/Flush measurement	Flush measurement.
Measurements\LeftGapX	Gap/Flush measurement	Left Gap X measurement.
Measurements\LeftGapZ	Gap/Flush measurement	Left Gap Z measurement.
Measurements\LeftFlushX	Gap/Flush measurement	Left Flush X measurement.
Measurements\LeftFlushZ	Gap/Flush measurement	Left Flush Z measurement.
Measurements\LeftSurfaceAngle	Gap/Flush measurement	Left Surface Angle measurement.
Measurements\RightGapX	Gap/Flush measurement	Right Gap X measurement.
Measurements\RightGapZ	Gap/Flush measurement	Right Gap Z measurement.
Measurements\RightFlushX	Gap/Flush measurement	Right Flush X measurement.
Measurements\RightFlushZ	Gap/Flush measurement	Right Flush Z measurement.
Measurements\RightSurfaceAngle	Gap/Flush measurement	Right Surface Angle measurement.

ProfilePanelEdge

Element	Type	Description
EdgeType	32s	Edge type: 0 – Tangent 1 – Corner
MinDepth	64f	Minimum depth.
MaxVoidWidth	64f	Maximum void width.

Element	Type	Description
SurfaceWidth	64f	Surface width.
SurfaceOffset	64f	Surface offset.
NominalRadius	64f	Nominal radius.
EdgeAngle	64f	Edge angle.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Edge region.

Gap/Flush Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
Axis *(Gap measurement only)*	32s	Measurement axis: 0 – Edge 1 – Surface 2 – Distance
Absolute *(Flush measurement only)*	Boolean	Setting for selecting absolute or signed result: 0 – Signed 1 – Absolute

ProfilePosition

A ProfilePosition element defines settings for a profile position tool and one or more of its measurements.

ProfilePosition Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *ProfilePosition* above.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
Feature	ProfileFeature	Element for feature detection.
Measurements\X	Position tool measurement	X measurement.
Measurements\Z	Position tool measurement	Z measurement.
Features\Point	GeometricFeature	Point PointFeature

Position Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state:

Element	Type	Description
		0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.

ProfileRoundCorner

A ProfileRoundCorner element defines settings for a profile round corner tool and one or more of its measurements.

ProfileRoundCorner Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.

Element	Type	Description
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
RefDirection	32s	Setting for reference side to use: 0 – Left 1 – Right
Edge	ProfilePanelEdge	Element for edge configuration
Measurements\X	Round Corner tool measurement	X measurement.
Measurements\Z	Round Corner tool measurement	Z measurement.
Measurements\Angle	Round Corner tool measurement	Angle measurement.
Features\CenterPoint	Geometric Feature	Circle Center PointFeature.
Features\EdgePoint	Geometric Feature	Edge PointFeature.

ProfilePanelEdge

Element	Type	Description
EdgeType	32s	Edge type: 0 – Tangent 1 – Corner
MinDepth	64f	Minimum depth.
MaxVoidWidth	64f	Maximum void width.
SurfaceWidth	64f	Surface width.
SurfaceOffset	64f	Surface offset.
NominalRadius	64f	Nominal radius.
EdgeAngle	64f	Edge angle.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Edge region.

Round Corner Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.

Element	Type	Description
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

ProfileStrip

A ProfileStrip element defines settings for a profile strip tool and one or more of its measurements.

The profile strip tool is dynamic, meaning that it can contain multiple measurements of the same type in the Measurements element.

ProfileStrip Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.

Element	Type	Description
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
BaseType	32s	Setting for the strip type: 0 – None 1 – Flat
LeftEdge	Bitmask	Setting for the left edge conditions: 1 – Raising 2 – Falling 4 – Data End 8 – Void
RightEdge	Bitmask	Setting for the right edge conditions: 1 – Raising 2 – Falling 4 – Data End 8 – Void
TiltEnabled	Boolean	Setting for tilt compensation: 0 – Disabled 1 – Enabled
SupportWidth	64f	Support width of edge (mm).
TransitionWidth	64f	Transition width of edge (mm).
MinWidth	64f	Minimum strip width (mm).
MinHeight	64f	Minimum strip height (mm).
MaxVoidWidth	64f	Void max (mm).
Region	ProfileRegion2d	Region containing the strip.
Measurements\X	Strip tool measurement	X measurement.
Measurements\Z	Strip tool measurement	Z measurement.
Measurements\Width	Strip tool measurement	Width measurement.
Measurements\Height	Strip tool measurement	Width measurement.

Strip Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
SelectType	32s	Method of selecting a groove when multiple grooves are found: 0 – Best 1 – Ordinal, from left 2 – Ordinal, from right
SelectIndex	32s	Index when SelectType is set to 1 or 2.
Location *(X, Z, and Height measurements only)*	32s	Setting for groove location to return from: 0 – Left 1 – Right 2 – Center

Script

A Script element defines settings for a script measurement.

Script Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Code	String	Script code.
Measurements\Output	(Collection)	Dynamic list of Output elements.

Output

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.

SurfaceBoundingBox

A SurfaceBoundingBox element defines settings for a surface bounding box tool and one or more of its measurements.

SurfaceBoundingBox Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *SurfaceBoundingBox* above.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.

Element	Type	Description
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
ZRotationEnabled	Boolean	Setting to enable/disable rotation of bounding box
AsymmetryDetectionType	32s	Determine whether to use asymmetry detection and if enabled, which dimension would be the basis of detection. The possible values are: 0 – None 1 – Length 2 – Width
RegionEnabled	Boolean	Setting to enable/disable region.
Region	Region3D	Measurement region.
Measurements\X	Bounding Box tool measurement	X measurement.
Measurements\Y	Bounding Box tool measurement	Y measurement.
Measurements\Z	Bounding Box tool measurement	Z measurement.
Measurements\Width	Bounding Box tool measurement	Width measurement.
Measurements\Length	BoundingBoxMeasurement	Length measurement
Measurements\Height	Bounding Box tool measurement	Height measurement.
Measurements\ZAngle	Bounding Box tool measurement	ZAngle measurement.
Measurements\GlobalX	Bounding Box tool measurement	Global X measurement.
Measurements\GlobalY	Bounding Box tool measurement	Global Y measurement.
Measurements\GlobalZAngle	Bounding Box tool measurement	Global Z Angle measurement.
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature
Features\AxisLine	GeometricFeature	AxisLine LineFeature

Bounding Box Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

SurfaceCsHole

A SurfaceCsHole element defines settings for a surface countersunk hole tool and one or more of its measurements.

SurfaceCsHole Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *SurfaceCsHole* above.

Element	Type	Description
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
NominalBevelAngle	64f	Nominal bevel angle (mm).
NominalOuterRadius	64f	Nominal outer radius (mm).
NominalInnerRadius	64f	Nominal inner radius (mm).
BevelRadiusOffset	64f	Bevel radus offset (mm).
Shape	32s	The shape of the countersunk hole: 0 – Cone 1 – Counterbore
PartialDetectionEnabled	Boolean	Setting to enable/disable partial detection: 0 – Disable 1 – Enable
RegionEnabled	Boolean	Setting to enable/disable region: 0 – Disable 1 – Enable
Region	Region3D	Measurement region.
RefRegionsEnabled	Boolean	Setting to enable/disable reference regions: 0 – Disable 1 – Enable
RefRegionCount	32s	Count of the reference regions which are to be used
RefRegions	(Collection)	Reference regions. Contains 2 SurfaceRegion2D elements.

Element	Type	Description
AutoTiltEnabled	Boolean	Setting to enable/disable tilt correction: 0 – Disable 1 – Enable
TiltXAngle	64f	Setting for manual tilt correction angle X.
TiltYAngle	64f	Setting for manual tilt correction angle Y.
CurveFitEnabled	Boolean	Setting to enable/disable curve fitting: 0 – Disable 1 – Enable
CurveOrientation	64f	The orientation of the curvature, in degrees.
PlaneFitRangeEnabled	Boolean	Setting to enable/disable the use of the plane fit range
PlaneFitRange	64f	Setting for the tolerance to use when doing the plane fit
Measurements\X	Countersunk Hole tool measurement	X measurement.
Measurements\Y	Countersunk Hole tool measurement	Y measurement.
Measurements\Z	Countersunk Hole tool measurement	Z measurement.
Measurements\OuterRadius	Countersunk Hole tool measurement	Outer Radius measurement.
Measurements\Depth	Countersunk Hole tool measurement	Depth measurement.
Measurements\BevelRadius	Countersunk Hole tool measurement	Bevel Radius measurement.
Measurements\BevelAngle	Countersunk Hole tool measurement	Bevel Angle measurement.
Measurements\XAngle	Countersunk Hole tool measurement	X Angle measurement.
Measurements\YAngle	Countersunk Hole tool measurement	Y Angle measurement.
Measurements\CounterboreDepth	Countersunk Hole tool measurement	CounterboreDepth measurement.
Measurements\AxisTilt	CsHoleMeasurement	Axis tilt measurement
Measurements\AxisOrientation	CsHoleMeasurement	Axis orientation measurement.
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature

Countersunk Hole Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).

Element	Type	Description
Name	String	Measurement name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *SurfaceCsHole* on page 611.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

SurfaceDimension

A SurfaceDimension element defines settings for a surface dimension tool and one or more of its measurements.

SurfaceDimension Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Surface source.

Element	Type	Description
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
Measurements\CenterX	Dimension tool measurement	Center X measurement
Measurements\CenterY	Dimension tool measurement	Center Y measurement
Measurements\CenterZ	Dimension tool measurement	Center Z measurement
Measurements\Distance	Dimension tool measurement	Distance measurement
Measurements\PlaneDistance	Dimension tool measurement	Plane Distance measurement
Measurements\Height	Dimension tool measurement	Height measurement
Measurements\Length	Dimension tool measurement	Length measurement
Measurements\Width	Dimension tool measurement	Width measurement

Dimension Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable

Element	Type	Description
		1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
Absolute *(Height, Length, and Width measurements only)*	Boolean	Setting for selecting absolute or signed result. 0 – Signed 1 – Absolute

Tool (type SurfaceEdge)

A Tool element of type SurfaceEdge defines settings for a surface edge tool and one or more of its measurements.

SurfaceEdge Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
@type	String	Type name of the tool.
@version	String	Version string for custom tool.
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for

Element	Type	Description
		anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Parameters\UseIntensity	GdkParamBool	Use intensity data.
Parameters\RegionCount	GdkParamInt	Count of regions.
Parameters\Region	GdkParamSurfaceRegion3d	Edge region parameters.
Parameters\Region1	GdkParamSurfaceRegion3d	Second edge region parameters.
Parameters\Region2	GdkParamSurfaceRegion3d	Third eddge region parameters.
Parameters\Region3	GdkParamSurfaceRegion3d	Fourth edge region parameter.
Parameters\SearchDirection	GdkParamInt	Direction of search.
Parameters\FixedAngleValue	GdkParamFloat	Fixed angle value
Parameters\FixedAngleValue.units	String	Units of fixed angle (e.g.: deg)
Parameters\UseFixedAngle	GdkParamBool	Use fixed angle boolean.
Parameters\PathSpacing	GdkParamFloat	Path spacing value
Parameters\PathSpacing.units	String	Units of path spacing (eg: mm)
Parameters\PathWidth	GdkParamFloat	Path width.
Parameters\PathWidth.units	String	Units of path width (e.g.: mm).
Parameters\SelectEdge	GdkParamInt	Edge selection type. Is either: 0 – Best 1 – First 2 – Last
Parameters\EdgeDirection	GdkParamInt	Edge direction type. Is either: 0 – Rising 1 – Falling 2 – Rising or Falling
Parameters\EdgeThreshold	GdkParamFloat	Edge threshold value.
Parameters\EdgeThreshold.units	String	Units of edge threshold (e.g.: mm).
Parameters\IntensityThreshold	GdkParamFloat	Intensity threshold value.
Parameters\UseRelativeThreshold	GdkParamBool	Use relative threshold boolean
Parameters\RelativeThreshold	GdkParamFloat	Relative threshold value.
Parameters\RelativeThreshold.units	String	Units of relative threshold (e.g.: %)

Element	Type	Description
Parameters\EdgeSmoothing	GdkParamFloat	Edge smoothing value.
Parameters\EdgeSmoothing.units	String	Units of edge smoothing (e.g.: mm).
Parameters\EdgeWidth	GdkParamFloat	The step width.
Parameters\EdgeWidth.units	String	Units of edge (e.g.: mm).
Parameters\EdgeMaxGap	GdkParamFloat	Edge max gap value.
Parameters\EdgeMaxGap.units	String	Units of edge max gap (eg: mm).
Parameters\FillBackground	GdkParamBool	Fill background boolean
Parameters\FillValue	GdkParamFloat	Fill value value.
Parameters\FillValue.units	String	Units of fill value (e.g.: mm).
Parameters\IntensityFillValue	GdkParamFloat	Intensity fill value value.
Parameters\IntensityFillValue.min	GdkParamFloat	Intensity fill value minimum value.
Parameters\IntensityFillValue.max	GdkParamFloat	Intensity fill value maximum value.
Parameters\RenderDetail	GdkParamBool	Render detail Boolean.
Measurements\Measurement @type=X	Edge Measurement	Base X measurement.
Measurements\Measurement @type=Y	Edge Measurement	Base Y measurement.
Measurements\Measurement @type=Z	Edge Measurement	Base Z measurement.
Measurements\Measurement @type=ZAngle	Edge Measurement	Base ZAngle measurement.
Measurements\Measurement @type=Height	Edge Measurement	Base Height measurement.
Features\Feature @type=EdgeLine	Gdk Feature	EdgeLine line feature.
Features\Feature @type=CenterPoint	Gdk Feature	CenterPoint point feature.

Edge Measurement Child Elements

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
@type	String	Type name of measurement.
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable

Element	Type	Description
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

SurfaceEllipse

A SurfaceEllipse element defines settings for a surface ellipse tool and one or more of its measurements.

SurfaceEllipse Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *SurfaceEllipse* above.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are:

Element	Type	Description
		1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
RegionEnabled	Boolean	Setting to enable/disable region.
Region	Region3D	Measurement region.
AsymmetryDetectionType	32s	Determine whether to use asymmetry detection and if enabled, which dimension would be the basis of detection. The possible values are: 0 – None 1 – Major 2 – Minor
Measurements\Major	Ellipse tool measurement	Major measurement.
Measurements\Minor	Ellipse tool measurement	Minor measurement.
Measurements\Ratio	Ellipse tool measurement	Ratio measurement.
Measurements\ZAngle	Ellipse tool measurement	ZAngle measurement.
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature
Features\MajorAxisLine	GeometricFeature	MajorAxisLine LineFeature
Features\MinroAxisLine	GeometricFeature	MinorAxisLine LineFeature

Ellipse Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable

Element	Type	Description
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

SurfaceHole

A SurfaceHole element defines settings for a surface hole tool and one or more of its measurements.

SurfaceHole Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *SurfaceHole* above.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface

Element	Type	Description
		4 – Section
Stream\Id	32u	The stream source ID.
NominalRadius	64f	Nominal radius (mm).
RadiusTolerance	64f	Radius tolerance (mm).
PartialDetectionEnabled	Boolean	Setting to enable/disable partial detection: 0 – Disable 1 – Enable
DepthLimitEnabled	Boolean	Setting to enable/disable depth limit: 0 – Disable 1 – Enable
DepthLimit	64f	The depth limit relative to the surface. Data below this limit is ignored.
RegionEnabled	Boolean	Setting to enable/disable region: 0 – Disable 1 – Enable
Region	Region3D	Measurement region.
RefRegionsEnabled	Boolean	Setting to enable/disable reference regions: 0 – Disable 1 – Enable
RefRegionCount	32s	Count of the reference regions that are to be used. (**Advanced** tab.)
RefRegions	(Collection)	Reference regions. Contains up to two RefRegion elements of type SurfaceRegion2D. (**Advanced** tab.)
AutoTiltEnabled	Boolean	Setting to enable/disable tilt correction: 0 – Auto Set 1 – Custom
TiltXAngle	64f	Setting for custom tilt correction angle X.
TiltYAngle	64f	Setting for custom tilt correction angle Y.
Measurements\X	Hole tool measurement	X measurement.
Measurements\Y	Hole tool measurement	Y measurement.
Measurements\Z	Hole tool measurement	Z measurement.
Measurements\Radius	Hole tool measurement	Radius measurement.
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature

Hole Tool Measurement

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).

Element	Type	Description
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

SurfaceOpening

A SurfaceOpening element defines settings for a surface opening tool and one or more of its measurements.

SurfaceOpening Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *SurfaceOpening* above.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.

Element	Type	Description
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
Type	32s	Type of the opening: 0 – Rounded 1 – Slot
NominalWidth	64f	Nominal width (mm).
NominalLength	64f	Nominal length (mm).
NominalAngle	64f	Nominal angle (degrees).
NominalRadius	64f	Nominal radius (mm).
WidthTolerance	64f	Radius tolerance (mm).
LengthTolerance	64f	Length tolerance (mm).
AngleTolerance	64f	Angle tolerance (degrees).
PartialDetectionEnabled	Boolean	Setting to enable/disable partial detection: 0 – Disable 1 – Enable
DepthLimitEnabled	Boolean	Setting to enable/disable depth limit: 0 – Disable 1 – Enable
DepthLimit	64f	The depth limit relative to the surface. Data below this limit is ignored.
RegionEnabled	Boolean	Setting to enable/disable region: 0 – Disable 1 – Enable
Region	Region3D	Measurement region.
RefRegionsEnabled	Boolean	Setting to enable/disable reference regions (**Advanced**

Element	Type	Description
		tab): 0 – Disable 1 – Enable
RefRegionCount	32s	Count of the reference regions that are to be used. (**Advanced** tab.)
RefRegions	(Collection)	Reference regions. Contains two RefRegion elements of type SurfaceRegion2D.
AutoTiltEnabled	Boolean	Setting to enable/disable tilt correction (**Advanced** tab): 0 – Disable 1 – Enable
TiltXAngle	64f	Setting for custom tilt correction angle X.
TiltYAngle	64f	Setting for custom tilt correction angle Y.
Measurements\X	Opening tool measurement	X measurement.
Measurements\Y	Opening tool measurement	Y measurement.
Measurements\Z	Opening tool measurement	Z measurement.
Measurements\Width	Opening tool measurement	Width measurement.
Measurements\Length	Opening tool measurement	Length measurement.
Measurements\Angle	Opening tool measurement	Angle measurement.
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature

Opening Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable

Element	Type	Description
		1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

SurfacePlane

A SurfacePlane element defines settings for a surface plane tool and one or more of its measurements.

SurfacePlane Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface

Element	Type	Description
		4 – Section
Stream\Id	32u	The stream source ID.
RegionsEnabled	Boolean	Setting to enable/disable regions: 0 – Disable 1 – Enable
RegionCount	32s	Count of the regions.
Regions	(Collection)	Measurement regions. Contains up to four Region elements of type Region3D.
Measurements\XAngle	Plane tool measurement	XAngle measurement.
Measurements\YAngle	Plane tool measurement	YAngle measurement.
Measurements\ZOffset	Plane tool measurement	ZOffset measurement.
Measurements\StdDev	Plane tool measurement	Standard deviation measurement
Measurements\MinError	Plane tool measurement	Minimum error measurement
Measurements\MaxError	Plane tool measurement	Maximum error measurement
Measurements\XNormal	PlaneMeasurement	XNormal measurement
Measurements\YNormal	PlaneMeasurement	YNormal measurement
Measurements\ZNormal	PlaneMeasurement	ZNormal measurement
Measurements\Distance	PlaneMeasurement	Distance from normal measurement
Features\Plane	GeometricFeature	Resulting plane PlaneFeature.

Plane Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state

Element	Type	Description
		0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

SurfacePosition

A SurfacePosition element defines settings for a surface position tool and one or more of its measurements.

SurfacePosition Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *SurfacePosition* above.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface

Element	Type	Description
		4 – Section
Stream\Id	32u	The stream source ID.
Feature	SurfaceFeature	Measurement feature.
Measurements\X	Position tool measurement	X measurement.
Measurements\Y	Position tool measurement	Y measurement.
Measurements\Z	Position tool measurement	Z measurement.
Features\Point	GeometricFeature	Point PointFeature

Position Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

SurfaceStud

A SurfaceStud element defines settings for a surface stud tool and one or more of its measurements.

SurfaceStud Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See *Feature Child Elements* on page 632.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
StudRadius	64f	Radius of stud (mm).
StudHeight	64f	Height of stud (mm).
BaseHeight	64f	Height of stud’s base.
TipHeight	64f	Height of stud’s tip.
RegionEnabled	Boolean	Setting to enable/disable region.
Region	Region3D	Measurement region.
RefRegionsEnabled	Boolean	Setting to enable/disable reference regions: 0 – Disable 1 – Enable
RefRegionCount	32s	Count of the reference regions that are to be used. (**Advanced** tab.)

Element	Type	Description
RefRegions	(Collection)	Reference regions. Contains up to four RefRegion elements of type SurfaceRegion2D. (**Advanced** tab.)
AutoTiltEnabled	Boolean	Setting to enable/disable tilt correction (**Advanced** tab): 0 – Auto Set 1 – Custom
TiltXAngle	64f	Setting for custom tilt correction angle X.
TiltYAngle	64f	Setting for custom tilt correction angle Y.
Measurements\BaseX	Stud tool measurement	BaseX measurement.
Measurements\BaseY	Stud tool measurement	BaseY measurement.
Measurements\BaseZ	Stud tool measurement	BaseZ measurement.
Measurements\TipX	Stud tool measurement	TipX measurement.
Measurements\TipY	Stud tool measurement	TipY measurement.
Measurements\TipZ	Stud tool measurement	TipZ measurement.
Measurements\Radius	Stud tool measurement	Radius measurement.
Features\TipPoint	GeometricFeature	TipPoint PointFeature
Features\BasePoint	GeometricFeature	BasePoint PointFeature

Stud Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.

Element	Type	Description
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
RadiusOffset *(Radius measurement only)*	64f	Radius offset of the stud.

Feature Child Elements

Element	Type	Description
@id	32s	The identifier of the geometric feature. -1 if unassigned.
@dataType	String	The data type of the feature. One of: – PointFeature – LineFeature
Name	String	The display name of the feature.
Enabled	Bool	Whether the given feature output is enabled.

SurfaceVolume

A SurfaceVolume element defines settings for a surface volume tool and one or more of its measurements.

SurfaceVolume Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Anchor\ZAngle	String (CSV)	The Z Angle measurements (IDs) used for anchoring.
Anchor\ZAngle.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of StreamOptions elements.

Element	Type	Description
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\Id	32u	The stream source ID.
RegionEnabled	Boolean	Setting to enable/disable region.
Region	Region3D	Measurement region.
Measurements\Volume	Volume tool measurement	Volume measurement.
Measurements\Area	Volume tool measurement	Area measurement.
Measurements\Thickness	Volume tool measurement	Thickness measurement.

Volume Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.

Element	Type	Description
Location *(Thickness measurement only)*	32s	Measurement type: 0 – Maximum 1 – Minimum 2 – 2D Centroid 3 – 3D Centroid 4 – Average 5 – Median

Tool (type FeatureDimension)

A Tool element of type FeatureDimension defines settings for a feature dimension tool and one or more of its measurements.

Tool Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
@type	String	Type name of the tool.
@version	String	Version string for custom tool.
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Parameters\RefPoint	GdkParamGeometricFeature	Reference point feature.
Parameters\Feature	GdkParamGeometricFeature	Reference feature.
Measurements\Measurement @type=Width	Dimension Measurement	Width measurement.

Element	Type	Description
Measurements\Measurement @type=Length	Dimension Measurement	Length measurement.
Measurements\Measurement @type=Height	Dimension Measurement	Width measurement.
Measurements\Measurement @type=Distance	Dimension Measurement	Distance measurement.
Measurements\Measurement @type=PlaneDistance	Dimension Measurement	Plane distance measurement.

Dimension Measurement Child Elements

@id	32s	Measurement ID. Optional (measurement disabled if not set).
@type	String	Type name of measurement.
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
Parameters\WidthAbsolute *(Width measurement only)*	GdkParamBool	Absolute width enabled boolean.
Parameters\LengthAbsolute *(Length measurement only)*	GdkParamBool	Absolute length enabled boolean.
Parameters\HeightAbsolute *(Height measurement only)*	GdkParamBool	Absolute height enabled boolean.

Tool (type FeatureIntersect)

A Tool element of type FeatureIntersect defines settings for a feature intersection tool and one or more of its measurements.

Tool Child Elements

Element	Type	Description
@isCustom	Bool	Reserved for future use.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
@id	32s	The tool's ID.
@type	String	Type name of the tool.
@version	String	Version string for custom tool.
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Parameters\Line	GdkParamGeometricFeature	Line feature input.
Parameters\RefLine	GdkParamGeometricFeature	Reference line feature input.
Measurements\Measurement @type=\X	Intersect Measurement	X measurement.
Measurements\Measurement @type=Y	Intersect Measurement	Y measurement.
Measurements\Measurement @type=Z	Intersect Measurement	Z measurement.
Measurements\Measurement @type=Angle	Intersect Measurement	Angle measurement.
Features\IntersectPoint	GDK Feature	Intersect point feature.

Intersect Measurement Child Elements

@id	32s	Measurement ID. Optional (measurement disabled if not

		set).
@type	String	Type name of measurement.
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state: 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Pinned	Boolean	Whether the measurement is pinned to main renderer.
Parameters\AngleRange	GdkParamInt	Angle range option choice. Is one of: 0 – -180 To 180 1 – 0 To 360

Custom

A Custom element defines settings for a user-created GDK-based tool and one or more of its measurements.

Custom Child Elements

Element	Type	Description
@type	String	Type name of the tool.
@version	String	Version string for custom tool.
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.

Element	Type	Description
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Parameters	GDK Parameter	Collection of parameters. The element name in the job file is the name of the parameter.
Measurements	GDK Measurement	Collection of measurements.
Features	GDK Feature	Collection of features.

Output

The Output element contains the following sub-elements: Ethernet, Serial, Analog, Digital0, and Digital1. Each of these sub-elements defines the output settings for a different type of output.

For all sub-elements, the source identifiers used for measurement outputs correspond to the measurement identifiers defined in each tool's Measurements element. For example, in the following XML, in the options attribute of the Measurements element, 2 and 3 are the identifiers of measurements that are enabled and available for output. The value of the Measurements element (that is, 2) means that only the measurement with id 2 (Surface Hole X) will be sent to output.

```
<SurfaceHole>   ...
  <Measurements>
    <X id="2">   ...
    <Y id="3">   ...


<Output>
  <Ethernet>     ...
    <Measurements options="2,3">2</Measurements>
```

Ethernet

The Ethernet element defines settings for Ethernet output.

In the Ethernet element, the source identifiers used for video, range, profile, and surface output, as well as range, profile, and surface intensity outputs, correspond to the *sensor* that provides the data. For example, in the XML below, the *options* attribute of the Surfaces element shows that only two sources are available (see the table below for the meanings of these values). The value in this element—0—indicates that only data from that source will be sent to output.

```
<Output>
  <Ethernet>
     ...
    <Ranges options=""/>
    <Profiles options=""/>
    <Surfaces options="0,1">0</Surfaces>
    ...
```

Ethernet Child Elements

Element	Type	Description
Ethernet.used	Boolean	Indicates if the output is available on the sensor.
Protocol	32s	The selected Ethernet protocol: 0 – Gocator 1 – Modbus 2 – EtherNet/IP 3 – ASCII 4 – PROFINET The Gocator protocol is always on and its output is always available, regardless of the output you choose. This allows simultaneous connections via an SDK application and a PLC, letting you for example archive or display scan data on a PC while controlling equipment with a PLC.
Protocol.options	32s (CSV)	List of available protocol options.
TimeoutEnabled	Boolean	Enable or disable auto-disconnection timeout. Applies only to the Gocator protocol.
Timeout	64f	Disconnection timeout (seconds). Used when TimeoutEnabled is true and the Gocator protocol is selected.
Ascii	Section	See *Ascii* on page 641.
EIP	Section	See *EIP* on page 642.
Modbus	Section	See *Modbus* on page 642.
Profinet	Section	See *Profinet* on page 642.
Ptp	Boolean	Enable or disable Precision Time Protocol support.
Videos	32s (CSV)	Selected video sources: 0 – Top 1 – Bottom 2 – Top left 3 – Top right 100 to 131 – G2 buddy sensor device indices for configurations with 2 to 31 buddy G2 sensors to identify a particular sensor's scan data. Main sensor is 100. First buddied sensor is 101. Second buddied sensor is 102 and so on.
Videos.options	32s (CSV)	List of available video sources (see above).
Ranges	32s (CSV)	Selected range sources: 0 – Top 1 – Bottom

Element	Type	Description
		2 – Top left 3 – Top right
Ranges.options	32s (CSV)	List of available range sources (see above).
Profiles	32s (CSV)	Selected profile sources: 0 – Top 1 – Bottom 2 – Top left 3 – Top right Selected video sources: 0 – Top 1 – Bottom 2 – Top left 3 – Top right 100 to 131 – G2 buddy sensor device indices for configurations with 2 to 31 buddy G2 sensors to identify a particular sensor's scan data. Main sensor is 100. First buddied sensor is 101. Second buddied sensor is 102 and so on.
Profiles.options	32s (CSV)	List of available profile sources (see above).
Surfaces	32s (CSV)	Selected surface sources: 0 – Top 1 – Bottom 2 – Top left 3 – Top right
Surfaces.options	32s (CSV)	List of available surface sources (see above).
SurfaceSections	32s (CSV)	Selected surface section sources.
SurfaceSections.options	32s (CSV)	List of available surface section sources.
RangeIntensities	32s (CSV)	Selected range intensity sources. 0 – Top 1 – Bottom 2 – Top left 3 – Top right
RangeIntensities.options	32s (CSV)	List of available range intensity sources (see above).
ProfileIntensities	32s (CSV)	Selected profile intensity sources. 0 – Top 1 – Bottom 2 – Top left 3 – Top right

Element	Type	Description
ProfileIntensities.options	32s (CSV)	List of available profile intensity sources (see above).
SurfaceIntensities	32s (CSV)	Selected surface intensity sources.
SurfaceIntensities.options	32s (CSV)	List of available surface intensity sources (see above).
SurfaceSectionIntensities	32s (CSV)	Selected surface section intensity sources
SurfaceSectionIntensities.options	32s (CSV)	List of available surface section intensity sources.
Tracheids	32s (CSV)	Selected tracheid sources.
Tracheids.options	32s (CSV)	List of available tracheid sources.
Measurements	32u (CSV)	Selected measurement sources.
Measurements.options	32u (CSV)	List of available measurement sources.
Events	32u (CSV)	Selected events
Events.Options	32u (CSV)	CSV list of possible event options: 0 – Exposure Begins 1 – Exposure Ends
Features	32u (CSV)	Selected feature sources.
Features.options	32u (CSV)	List of available feature sources.
ToolData	32u (CSV)	Selected tool data sources.
ToolData.options	32u (CSV)	List of available tool data sources.

Ascii

Ascii Child Elements

Element	Type	Description
Operation	32s	Operation mode: 0 – Asynchronous 1 – Polled
ControlPort	32u	Control service port number.
HealthPort	32u	Health service port number.
DataPort	32u	Data service port number.
Delimiter	String	Field delimiter.
Terminator	String	Line terminator.
InvalidValue	String	String for invalid output.
CustomDataFormat	String	Custom data format.
CustomFormatEnabled	Bool	Enables custom data format.
StandardFormatMode	32u	The formatting mode used if not a custom format: 0 – Standard 1 – Standard with Stamp

EIP

EIP Child Elements

Element	Type	Description
BufferEnabled	Bool	Enables EtherNet/IP output buffering.
EndianOutputType	32s	Endian output type: 0 – Big endian 1 – Little endian
ImplicitOutputEnabled	Bool	Enables Implict (I/O) Messaging.
ImplicitTriggerOverride	32s	Override requested trigger type by client: 0 – No override 1 – Cyclic 2 – Change of State

Modbus

Modbus Child Elements

Element	Type	Description
BufferEnabled	Bool	Enables Modbus output buffering.

Profinet

Profinet Child Elements

Element	Type	Description
IpAddress	String	Address in dotted notation (e.g. 1.1.1.1).
PrefixLength	32u	Length of prefix for the subnet.
SubnetMask	String	Address in dotted notation (e.g. 1.1.1.1).
Gateway	String	Address in dotted notation (e.g. 1.1.1.1).
DeviceName	String	Profinet name for the device.

Digital0 and Digital1

The Digital0 and Digital1 elements define settings for a sensor's two digital outputs.

Digital0 and Digital1 Child Elements

Element	Type	Description
Digital0.used	Boolean	Indicates if the output is available on the sensor.
Event	32s	Triggering event: 0 – None (disabled) 1 – Measurements 2 – Software 3 – Alignment state 4 – Acquisition start

Element	Type	Description
		5 – Acquisition end
SignalType	32s	Signal type: 0 – Pulse 1 – Continuous
ScheduleEnabled	Bool	Enables scheduling.
PulseWidth	64f	Pulse width (µs).
PulseWidth.min	64f	Minimum pulse width (µs).
PulseWidth.max	64f	Maximum pulse width (µs).
PassMode	32s	Measurement pass condition: 0 – AND of measurements is true 1 – AND of measurements is false 2 – Always assert
Delay	64f	Output delay (µs or mm, depending on delay domain defined below).
DelayDomain	32s	Output delay domain: 0 – Time (µs) 1 – Encoder (mm)
Inverted	Bool	Whether the sent bits are flipped.
Measurements	32u (CSV)	Selected measurement sources.
Measurements.options	32u (CSV)	List of available measurement sources.

Analog

The Analog element defines settings for analog output.

The range of valid measurement values [DataScaleMin, DataScaleMax] is scaled linearly to the specified current range [CurrentMin, CurrentMax].

Only one Value or Decision source can be selected at a time.

Analog Child Elements

Element	Type	Description
Analog.used	Boolean	Indicates if the output is available on the sensor.
Event	32s	Triggering event: 0 – None (disabled) 1 – Measurements 2 – Software
ScheduleEnabled	Bool	Enables scheduling.
CurrentMin	64f	Minimum current (mA).
CurrentMin.min	64f	Minimum value of minimum current (mA).
CurrentMin.max	64f	Maximum value of minimum current (mA).

Element	Type	Description
CurrentMax	64f	Maximum current (mA).
CurrentMax.min	64f	Minimum value of maximum current (mA).
CurrentMax.max	64f	Maximum value of maximum current (mA).
CurrentInvalidEnabled	Bool	Enables special current value for invalid measurement value.
CurrentInvalid	64f	Current value for invalid measurement value (mA).
CurrentInvalid.min	64f	Minimum value for invalid current (mA).
CurrentInvalid.max	64f	Maximum value for invalid current (mA).
DataScaleMax	64f	Measurement value corresponding to maximum current.
DataScaleMin	64f	Measurement value corresponding to minimum current.
Delay	64f	Output delay (µs or mm, depending on delay domain defined below).
DelayDomain	32s	Output delay domain: 0 – Time (µs) 1 – Encoder (mm)
Measurement	32u	Selected measurement source.
Measurement.options	32u (CSV)	List of available measurement sources.

The delay specifies the time or position at which the analog output activates. Upon activation, there is an additional delay before the analog output settles at the correct value.

Serial

The Serial element defines settings for Serial output.

Serial Child Elements

Element	Type	Description
Serial.used	Boolean	Indicates if the output is available on the sensor.
Protocol	32s	Serial protocol: 0 – ASCII 1 – Selcom
Protocol.options	32s (CSV)	List of available protocols.
Selcom	Section	See *Selcom* below.
Ascii	Section	See *Ascii* on the next page.
Measurements	32u (CSV)	Selected measurement sources.
Measurements.options	32u (CSV)	List of available measurement sources.

Selcom

Selcom Child Elements

Element	Type	Description
Rate	32u	Output bit rate.

Element	Type	Description
Rate.options	32u (CSV)	List of available rates.
Format	32s	Output format: 0 – 12-bit 1 – 12-bit with search 2 – 14-bit 3 – 14-bit with search
Format.options	32s (CSV)	List of available formats.
DataScaleMin	64f	Measurement value corresponding to minimum word value.
DataScaleMax	64f	Measurement value corresponding to maximum word value.
Delay	64u	Output delay in µs.

Ascii

Ascii Child Elements

Element	Type	Description
Rate	32u	Output bit rate.
Rate.options	32u (CSV)	List of available rates.
Delimiter	String	Field delimiter.
Terminator	String	Line terminator.
InvalidValue	String	String for invalid output.
CustomDataFormat	String	Custom data format.
CustomFormatEnabled	Bool	Enables custom data format.
StandardFormatMode	32u	The formatting mode used if not a custom format: 0 – Standard 1 – Standard with Stamp

Transform

The transformation component contains information about the physical system setup that is used to:

- Transform data from sensor coordinate system to another coordinate system (e.g., world)
- Define encoder resolution for encoder-based triggering
- Define the travel offset (Y offset) between sensors for staggered operation

You can access the Transform component of the active job as an XML file, either using path notation, via "_live.job/transform.xml", or directly via "_live.tfm".

You can access the Transform component in user-created job files in non-volatile storage, for example, "productionRun01.job/transform.xml". You can only access transformations in user-created job files using path notation.

See the following sections for the elements contained in this component.

Transformation Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Transform version="100">
  <EncoderResolution>1</EncoderResolution>
  <Speed>100</Speed>
  <Devices>
    <Device role="0">
      <X>-2.3650924829</X>
      <Y>0.0</Y>
      <Z>123.4966803469</Z>
      <XAngle>5.7478302588</XAngle>
      <YAngle>3.7078302555</XAngle>
      <ZAngle>2.7078302556</XAngle>
    </Device>
    <Device id="1">
      <X>0</X>
      <Y>0.0</Y>
      <Z>123.4966803469</Z>
      <XAngle>5.7478302588</XAngle>
      <YAngle>3.7078302555</XAngle>
      <ZAngle>2.7078302556</XAngle>
    </Device>
  </Devices>
</Transform>
```

The Transform element contains the alignment record for the sensor.

Transform Child Elements

Element	Type	Description
@version	32u	Major transform version (100).
@versionMinor	32u	Minor transform version (0).
EncoderResolution	64f	Encoder Resolution (mm/tick).
Speed	64f	Travel Speed (mm/s).
Devices	(Collection)	Contains two Device elements.

Device

A Device element defines the transformation for a sensor. There is one entry element per sensor, identified by a unique role attribute (0 for main and 1 for buddy):

Device Child Elements

Element	Type	Description
@role	32s	Role of device described by this section: 0 – Main

Element	Type	Description
		1 – Buddy
X	64f	Translation on the X axis (mm).
Y	64f	Translation on the Y axis (mm).
Z	64f	Translation on the Z axis (mm).
XAngle	64f	Rotation around the X axis (degrees).
YAngle	64f	Rotation around the Y axis (degrees).
ZAngle	64f	Rotation around the Z axis (degrees).

The rotation (counter-clockwise in the X-Z plane) is performed before the translation.

Part Models

Part models represent models created using the part matching feature.

You can access a model in the active job using path notation. For example, to access a model called scan.mdl, use "_live.job/scan.mdl".

You can access part models in user-created job files in non-volatile storage, for example, "productionRun01.job/model1.mdl". You can only access part models in user-created job files using path notation.

See the following sections for the elements contained in a model.

Part models contain the following subcomponents. You can access the subcomponents using path notation, for example, "productionRun01.job/myModel.mdl/config.xml".

Part Model Child Elements

Element	Type	Description
Configuration	config.xml	Model configuration XML. It is always present. (See *Configuration* on the next page.)
Edge Points	edge-height-top	Edge points for the top heightmap. (See *Edge Points* on the next page.)
Edge Points	edge-height-bottom	Edge points for the bottom heightmap.
Edge Points	edge-intensity-top	Edge points for the top intensity map.
Edge Points	edge-intensity-bottom	Edge points for the bottom intensity map.

The edge points file exists only when the model contains the source data for the edge points.

Edge Points

Edge Points Data

Field	Type	Offset	Description
id	16s	0	Sender ID -1 – Part matching
source	8s	2	Source 0 – Model 1 – Target
imageType	8s	3	Image type 0 – Height map 1 – Intensity map
imageSource	8s	4	Image source 0 – Top 1 – Bottom
width	32u	5	Width of model space, in units of xScale
length	32u	9	Length of model space, un units of yScale
xScale	32u	13	X scale (nm)
yScale	32u	17	Y scale (nm)
xOffset	32s	21	X offset (µm)
yOffset	32s	25	Y offset µm
zAngle	32s	29	Z rotation (microdegrees)
pointCount	32u	33	Number of edge points
points[pointCount]	(32u, 32u)	37	Edge points collection. Each point is a tuple of x and y values, in units of xScale and yScale, respectively.

Configuration

Configuration Child Elements

Element	Type	Description
@version	32u	Major version (1).
@versionMinor	32u	Minor version (0).
Edges	Collection	Collection of Edge items (described below).
EdgeSensitivity	64f	Sensitivity recorded during model edges generation (read-only).
TransformedDataRegion	Region3d	Data region of the model.
ZAngle	64f	Additional rotation applied to the model (degrees).
TargetEdgeSensitivity	64f	Sensitivity used to generate target edges.
ImageType	32s	Selects type of image used to generate edges: 0 – Height map 1 – Intensity map

Element	Type	Description
ImageType.options	32s (CSV)	List of available image types.

Integrations

Several integration tools are provided in the Utilities package available from the Downloads center, in the Software subsection for your sensor model and Gocator software release.

- Adaptive Vision AVParser: A set of Python scripts that allow creating Gocator GDK measurement tools from an Adaptive Vision project.
- GenICam GenTL driver (see below)
- GoRobot: A library providing support for robot-sensor integration
- LabVIEW (for more information, see the LabVIEW application guide at https://downloads.lmi3d.com/interfacing-gocator-labview-4x-guide): A set of Virtual Instruments (VIs) for interfacing LabVIEW with Gocator 2x00 sensors.
- MountainsMap transfer tool (see below)
- Rockwell EtherNet/IP files
- Universal Robots integration (see below)

Protocols

Gocator supports protocols for communicating with sensors over Ethernet (TCP/IP) and serial output. For a protocol to output data, it must be enabled and configured in the active job.

> The Gocator protocol is always on and its output is always available, regardless of the output you choose. This allows simultaneous connections via an SDK application and a PLC, letting you for example archive or display scan data on a PC while controlling equipment with a PLC.

> The Gocator emulator and accelerator (software and GoMax) do not support the PROFINET protocol.

> If you switch jobs or make changes to a job using the SDK or a protocol (from a PLC), the switch or changes are not automatically displayed in the web interface: you must refresh the browser to see these.

Protocols available over Ethernet

- Gocator
- Modbus
- EtherNet/IP
- PROFINET
- ASCII

For an overview of the Ethernet ports used by sensors, see *Required Ports* on page 39.

Protocols available over serial

- ASCII

Gocator Protocol

This section describes the TCP and UDP commands and data formats used by a client computer to communicate with Gocator sensors using the Gocator protocol. It also describes the connection types (Discovery, Control, Upgrade, Data, and Health), and data types. The protocol enables the client to:

- Send commands to run sensors, provide software triggers, read/write files, etc.
- Receive data, health, and diagnostic messages.
- Upgrade firmware.

> The Gocator 4.x/5.x firmware uses mm, mm^2, mm^3, and degrees as standard units. In all protocols, values are scaled by 1000, as values in the protocols are represented as integers. This results in effective units of mm/1000, mm^2/1000, mm^3/1000, and deg/1000 in the protocols.

To use the protocol, it must be enabled and configured in the active job.

> Sensors send UDP broadcasts over the network over the Internal Discovery channel (port 2016) at regular intervals during operation to perform peer discovery.

> The Gocator SDK provides open source C language libraries that implement the network commands and data formats defined in this section. For more information, see *GoSDK* on page 836.

For information on configuring the protocol using the web interface, see *Ethernet Output* on page 512.

For information on job file structures (for example, if you wish to create job files programmatically), see *Job File Structure* on page 552.

Data Types

The table below defines the data types and associated type identifiers used in this section.

All values except for IP addresses are transmitted in little endian format (least significant byte first) unless stated otherwise. The bytes in an IP address "a.b.c.d" will always be transmitted in the order a, b, c, d (big endian).

Data Types

Type	Description	Null Value
char	Character (8-bit, ASCII encoding)	-
byte	Byte.	-
8s	8-bit signed integer.	-128
8u	8-bit unsigned integer.	255U
16s	16-bit signed integer.	-32768 (0x8000)

Type	Description	Null Value
16u	16-bit unsigned integer.	65535 (0xFFFF)
32s	32-bit signed integer.	-2147483648 (0x80000000)
32u	32-bit unsigned integer.	4294967295 (0xFFFFFFFF)
32f	32-bit floating point.	-3.402823466e+38F
64s	64-bit signed integer.	-9223372036854775808 (0x8000000000000000)
64u	64-bit unsigned integer.	18446744073709551615 (0xFFFFFFFFFFFFFFFF)
64f	64-bit floating point	-1.7976931348623157e+308
Point16s	Two 16-bit signed integers	-
Point64f	Two 64-bit floating point values	-
Point3d64f	Three 64-bit floating point values	-
Point3d32f	Three 32-bit floating point values	-
Rect64f	Four 64-bit floating point values	-
Rect3d64f	Eight 64-bit floating point values	-
Facet3d32u	Three 32-bit unsigned integers	-
Transform3d64f	Twelve 64-bit floating point values ie. { xx, xy, xz, xt, yx, yy, yz, yt, zx, zy, zz, zt }	-

Commands

The following sections describe the commands available on the Discovery (page 653), Control (page 656), and Upgrade (page 692) channels.

When a client sends a command over the Control or Upgrade channel, the sensor sends a reply whose identifier is the same as the command's identifier. The identifiers are listed in the tables of each of the commands.

Status Codes

Each reply on the Discovery, Control, and Upgrade channels contains a *status* field containing a status code indicating the result of the command. The following status codes are defined:

Status Codes

Label	Value	Description
OK	1	Command succeeded.
Failed	0	Command failed.
Invalid State	-1000	Command is not valid in the current state.
Item Not Found	-999	A required item (e.g., file) was not found.
Invalid Command	-998	Command is not recognized.
Invalid Parameter	-997	One or more command parameters are incorrect.

Label	Value	Description
Not Supported	-996	The operation is not supported.
Simulation Buffer Empty	-992	The simulation buffer is empty.

Discovery Commands

Sensors ship with the following default network configuration:

Setting	Default
DHCP	0 (disabled)
IP Address	192.168.1.10
Subnet Mask	255.255.255.0
Gateway	0.0.0.0 (disabled)

Use the Get Address and Set Address commands to modify a sensor's network configuration. These commands are UDP broadcast messages:

Destination Address	Destination Port
255.255.255.255	3220

When a sensor accepts a discovery command, it will send a UDP broadcast response:

Destination Address	Destination Port
255.255.255.255	Port of command sender.

The use of UDP broadcasts for discovery enables a client computer to locate a sensor when the senor and client are configured for different subnets. All you need to know is the serial number of the sensor in order to locate it on an IP network.

Get Address

The Get Address command is used to discover sensors across subnets.

Command

Field	Type	Offset	Description
length	64s	0	Command length.
type	64s	8	Command type (0x1).
signature	64s	16	Message signature (0x0000504455494D4C)
deviceId	64s	24	Serial number of the device whose address information is queried. 0 selects all devices.

Reply

Field	Type	Offset	Description
length	64s	0	Reply length.
type	64s	8	Reply type (0x1001).

Field	Type	Offset	Description
status	64s	16	Operation status.
signature	64s	24	Message signature (0x0000504455494D4C)
deviceId	64s	32	Serial number.
dhcpEnabled	64s	40	0 – Disabled 1 – Enabled
reserved[4]	byte	48	Reserved.
address[4]	byte	52	The IP address in left to right order.
reserved[4]	byte	56	Reserved.
subnetMask[4]	byte	60	The subnet mask in left to right order.
reserved[4]	byte	64	Reserved.
gateway[4]	byte	68	The gateway address in left to right order.
reserved[4]	byte	72	Reserved.
reserved[4]	byte	76	Reserved.

Set Address

The Set Address command modifies the network configuration of a sensor. On receiving the command, the sensor will perform a reset. You should wait 30 seconds before re-connecting to the sensor.

Command

Field	Type	Offset	Description
length	64s	0	Command length.
type	64s	8	Command type (0x2).
signature	64s	16	Message signature (0x0000504455494D4C)
deviceId	64s	24	Serial number of the device whose address information is queried. 0 selects all devices.
dhcpEnabled	64s	32	0 – Disabled 1 – Enabled
reserved[4]	byte	40	Reserved.
address[4]	byte	44	The IP address in left to right order.
reserved[4]	byte	48	Reserved.
subnetMask[4]	byte	52	The subnet mask in left to right order.
reserved[4]	byte	56	Reserved.
gateway[4]	byte	60	The gateway address in left to right order.
reserved[4]	byte	64	Reserved.
reserved[4]	byte	68	Reserved.

Reply

Field	Type	Offset	Description
length	64s	0	Reply length.
type	64s	8	Reply type (0x1002).
status	64s	16	Operation status. For a list of status codes, see *Commands* on

Field	Type	Offset	Description
			page 652.
signature	64s	24	Message signature (0x0000504455494D4C).
deviceId	64s	32	Serial number.

Get Info

The Get Info command is used to retrieve sensor information.

Command

Field	Type	Offset	Description
length	64s	0	Command length.
type	64s	8	Command type (0x5).
signature	64s	16	Message signature (0x0000504455494D4C).
deviceId	64s	24	Serial number of the device whose address information is queried. 0 selects all devices.

Reply

Field	Type	Offset	Description
length	64s	0	Reply length.
type	64s	8	Reply type (0x1005).
status	64s	16	Operation status. For a list of status codes, see *Commands* on page 652.
signature	64s	24	Message signature (0x0000504455494D4C).
attrCount	16u	32	Byte count of the attributes (begins after this field and ends before propertyCount).
id	32u	34	Serial number.
version	32u	38	Version as a 4-byte integer (encoded in little-endian).
uptime	64u	42	Sensor uptime (microseconds).
ipNegotiation	byte	50	IP negotiation type: 0 – Static 1 – DHCP
addressVersion	byte	51	IP address version (always 4).
address[4]	byte	52	IP address.
reserved[12]	byte	56	Reserved.
prefixLength	32u	68	Subnet prefix length (in number of bits).
gatewayVersion	byte	72	Gateway address version (always 4).
gatewayAddress[4]	byte	73	Gateway address.
reserved[12]	byte	77	Reserved.
controlPort	16u	89	Control channel port.

Field	Type	Offset	Description
upgradePort	16u	91	Upgrade channel port.
healthPort	16u	93	Health channel port.
dataPort	16u	95	Data channel port.
webPort	16u	97	Web server port.
propertyCount	8u	99	Number of sensor ID properties.
properties[propertyCount]	Property	100	List of sensor ID properties.

Property

Field	Type	Description
nameLength	8u	Length of the name.
name[nameLength]	char	Name string.
valueLength	8u	Length of the value.
value[valueLength]	char	Value string.

Control Commands

A client sends control commands for most operations over the Control TCP channel (port 3190).

The Control channel and the Upgrade channel (port 3192) can be connected simultaneously. For more information on Upgrade commands, see *Upgrade Commands* on page 692.

States

A sensor system can be in one of two states: Ready or Running. The client sends the Start and Stop control commands to change the system's current state to Running and Ready, respectively. The sensor can also be configured to boot in either the Ready or Running state, by enabling or disabling autostart, respectively, using the Set Auto Start Enabled command.

In the Ready state, a sensor can be configured. In the Running state, a sensor responds to input signals, performs measurements, drives its outputs, and sends data messages to the client.

The state of the sensor can be retrieved using the Get States or Get System Info command.

Progressive Reply

Some commands send replies progressively, as multiple messages. This allows the sensor to stream data without buffering it first, and allows the client to obtain progress information on the stream.

A progressive reply begins with an initial, standard reply message. If the *status* field of the reply indicates success, the reply is followed by a series of “continue” reply messages.

A continue reply message contains a block of data of variable size, as well as status and progress information. The series of continue messages is ended by either an error, or a continue message containing 0 bytes of data.

Protocol Version

The Protocol Version command returns the protocol version of the connected sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4511)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4511).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
majorVersion	8u	10	Major version.
minorVersion	8u	11	Minor version.

Get Address

The Get Address command is used to get a sensor address.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x3012)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x3012).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
dhcpEnabled	byte	10	0 – DHCP not used 1 – DHCP used
address[4]	byte	11	IP address (most significant byte first).
subnetMask[4]	byte	15	Subnet mask.
gateway[4]	byte	19	Gateway address.

Set Address

The Set Address command modifies the network configuration of a sensor. On receiving the command, the sensor will perform a reset. You should wait 30 seconds before re-connecting to the sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.

Field	Type	Offset	Description
id	16u	4	Command identifier (0x3013)
dhcpEnabled	byte	6	0 – DHCP not used 1 – DHCP used
address[4]	byte	7	IP address (most significant byte first).
subnetMask[4]	byte	11	Subnet mask.
gateway[4]	byte	15	Gateway address.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x3013).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Get System Info V2

The Get System Info command reports information about the local node, remote nodes and assigned buddies.

Firmware version refers to the version of the sensor's firmware installed on each individual sensor. The client can upgrade the sensor's firmware by sending the Start Upgrade command (see *Start Upgrade* on page 692). Firmware upgrade files are available from the downloads section under the support tab on the LMI web site. For more information on getting the latest firmware, see *Firmware Upgrade* on page 83.

Every sensor contains factory backup firmware. If a firmware upgrade command fails (e.g., power is interrupted), the factory backup firmware will be loaded when the sensor is reset or power cycled. In this case, the sensors will fall back to the factory default IP address. To avoid IP address conflicts in a multi-sensor system, connect to one sensor at a time and re-attempt the firmware upgrade.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4010)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4010).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
localInfoSize	16u	10	Size of localInfo structure. Current: 116.

Field	Type	Offset	Description
localInfo	Local Info	12	Info for this device.
remoteCount	32u	-	Number of discovered sensors.
remoteInfoSize	16u	-	Size of remoteInfo structure. Current 124.
remoteInfo[remoteCount]	Remote Info	-	List of info for discovered sensors.
buddyInfoCount	32u	-	Number of buddies assigned (can be 0).
buddyInfoSize	16u	-	Size of buddyInfo structure. Current: 8.
Buddies[buddyCount]	Buddy Info	-	List of info for the assigned buddies.

Local Info

Field	Type	Offset	Description
deviceId	32u	0	Serial number of the device.
address[4]	byte	4	IP address (most significant byte first).
modelName[32]	char	8	Model name; "part number" starting with GoSdk 5.3.17.23. Should not be parsed.
firmwareVersion[4]	byte	40	Firmware version (most significant byte first).
state	32s	44	Sensor state 0 – Ready 1 – Running For more information on states, see *Control Commands* on page 656.
role	32s	48	Sensor role 0 – Main
modelNumber[32]	char	52	Model number that can be parsed.
modelDisplayName[32]	char	56	User-friendly model display name that can be used to rename sensors more appropriately for custom-branding naming.

Remote Info

Field	Type	Offset	Description
deviceId	32u	0	Serial number of the remote device.
address[4]	byte	4	IP address (most significant byte first).
modelName[32]	char	8	Remote model name; "remote part number" starting with GoSdk 5.3.17.23.
firmwareVersion[4]	byte	40	Remote firmware version (most significant byte first).
state	32s	44	Remote sensor state 0 – Ready 1 – Running For more information on states, see *Control Commands* on page 656.

Field	Type	Offset	Description
role	32s	48	Sensor role 0 – Main
mainId	32u	52	Serial number of the main device, or zero.
buddyableStatus	32s	56	Whether or not the device can be buddied: 1 – Can be buddied Errors: 0 – Unbuddiable (General Error) -100 – Already buddied -99 – Invalid State (e.g. running) -98 – Version Mismatch -97 – Model Mismatch
modelNumber[32]	char	60	Model number that can be parsed.
modelDisplayName[32]	char	92	Remote user-friendly model display name that can be used to rename sensors more appropriately for custom-branding naming.

Buddy Info

Field	Type	Offset	Description
deviceId	32u	2	Serial number of the device.
state	k32s	6	Buddy state 2 - Connecting 1 – Connected Errors: 0 – Unbuddiable (General Error) -100 – Already buddied -99 – Invalid State (e.g. running) -98 – Version Mismatch -97 – Model Mismatch -95 – Device Missing -92 – Standalone Sensor -91 – Restricted Sensor Mismatch

Get System Info

> This version of the Get System Info command is deprecated. Use Get System Info (v2) instead.

The Get System Info command reports information for sensors that are visible in the system.

Firmware version refers to the version of the sensor's firmware installed on each individual sensor. The client can upgrade the sensor's firmware by sending the Start Upgrade command (see *Start Upgrade* on page 692). Firmware upgrade files are available from the downloads section under the support tab on the LMI web site. For more information on getting the latest firmware, see *Firmware Upgrade* on page 83.

Every sensor contains factory backup firmware. If a firmware upgrade command fails (e.g., power is interrupted), the factory backup firmware will be loaded when the sensor is reset or power cycled. In this case, the sensors will fall back to the factory default IP address. To avoid IP address conflicts in a multi-sensor system, connect to one sensor at a time and re-attempt the firmware upgrade.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4002)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4002).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
localInfo	Sensor Info	10	Info for this device.
remoteCount	32u	66	Number of discovered sensors.
remoteInfo[remoteCount]	Sensor Info	70	List of info for discovered sensors.

Sensor Info

Field	Type	Offset	Description
deviceId	32u	0	Serial number of the device.
address[4]	byte	4	IP address (most significant byte first).
modelName[32]	char	8	Model name.
firmwareVersion[4]	byte	40	Firmware version (most significant byte first).
state	32s	44	Sensor state 0 – Ready 1 – Running For more information on states, see *Control Commands* on page 656.
role	32s	48	Sensor role 0 – Main

Get States

The Get States command returns various system states.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4525)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4525).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
count	32u	10	Number of state variables.
sensorState	32s	14	Sensor state 0 – Ready 1 – Running For more information on states, see *Control Commands* on page 656.
loginState	32s	18	Device login state 0 – No user 1 – Administrator 2 – Technician
alignmentReference	32s	22	Alignment reference 0 – Fixed 1 – Dynamic
alignmentState	32s	26	Alignment state 0 – Unaligned 1 – Aligned
recordingEnabled	32s	30	Whether or not recording is enabled 0 – Disabled 1 – Enabled
playbackSource	32s	34	Playback source 0 – Live data 1 – Recorded data
uptimeSec	32su	38	Uptime (whole seconds component)
uptimeMicrosec	32u	42	Uptime (remaining microseconds component)
playbackPos	32u	46	Playback position
playbackCount	32u	50	Playback frame count
autoStartEnabled	32u	54	Auto-start enable (boolean)

Field	Type	Offset	Description
isAccelerator	32u	58	Is the device an accelerator instance?
voltage	32u	62	Voltage setting 0 – 48V 1 – 24V
cableLength	32u	66	Cable length (maximum Is 60.0 meters, default is 5.0 meters)
quickEditEnabled	32u	70	Quick Edit state
securityLevel	32s	74	Security Level 0 – No security, any user type can access system. 1 – Basic security level, only authorized user types can access system.
brandingType	32s	78	Branding Type 0 – None/Gocator (default) 1 – White Label 2 – Custom

Log In/Out

The Log In/Out command is used to log in or out of a sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4003).
userType	32s	6	Defines the user type 0 – None (log out) 1 – Administrator 2 – Technician
password[64]	char	10	Password (required for log-in only).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4003).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Change Password

The Change Password command is used to change log-in credentials for a user.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4004).
user type	32s	6	Defines the user type 0 – None (log out) 1 – Administrator 2 – Technician
password[64]	char	10	New password.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4004).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Passwords can only be changed if a user is logged in as an administrator.

Set Buddy

The Set Buddy command is used to assign or unassign a Buddy sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4005).
buddyId	32u	6	Id of the sensor to acquire as buddy. Set to 0 to remove buddy.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4005).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

List Files

The List Files command returns a list of the files in the sensor's file system.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.

Field	Type	Offset	Description
id	16u	4	Command identifier (0x101A).
extension[64]	char	6	Specifies the extension used to filter the list of files (does not include the "."). If an empty string is used, then no filtering is performed.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101A).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
count	32u	10	Number of file names.
fileNames[count][64]	char	14	File names.

Copy File

The Copy File command copies a file from a source to a destination within the connected sensor (a .job file, a component of a job file, or another type of file; for more information, see *Job File Structure* on page 552).

To make a job active (to load it), copy a saved job to "_live.job".

To "save" the active job, copy from "_live.job" to another file.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101B).
source[64]	char	6	Source file name.
destination[64]	char	70	Destination file name.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101B).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Read File

Downloads a file from the connected sensor (a .job file, a component of a job file, or another type of file; for more information, see *Job File Structure* on page 552).

To download the live configuration, pass "_live.job" in the *name* field.

To read the configuration of the live configuration only, pass "_live.job/config.xml" in the *name* field.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1007).
name[64]	char	6	Source file name.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1007).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
length	32u	10	File length.
data[length]	byte	14	File contents.

Write File

The Write File command uploads a file to the connected sensor (a .job file, a component of a job file, or another type of file; for more information, see *Job File Structure* on page 552).

To make a job file live, write to "_live.job". Except for writing to the live file, the file is permanently stored on the sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1006).
name[64]	char	6	Source file name.
length	32u	70	File length.
data[length]	byte	74	File contents.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1006).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Delete File

The Delete File command removes a file from the connected sensor (a .job file, a component of a job file, or another type of file; for more information, see *Job File Structure* on page 552).

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1008).
name[64]	char	6	Source file name.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1008).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

User Storage Used

The User Storage Used command returns the amount of user storage that is used.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1021).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1021).
status	32s	6	Reply status.
spaceUsed	64u	10	The used storage space in bytes.

User Storage Free

The User Storage Free command returns the amount of user storage that is free.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1022).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1022).
status	32s	6	Reply status.
spaceFree	64u	10	The free storage space in bytes.

Get Default Job

The Get Default Job command gets the name of the job the sensor loads when it powers up.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4100).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4100).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
name[64]	char	10	The file name (null-terminated) of the job the sensor loads when it powers up.

Set Default Job

The Set Default Job command sets the job the sensor loads when it powers up.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4101).
fileName[64]	char	6	File name (null-terminated) of the job the sensor loads when it powers up.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4101).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Get Loaded Job

The Get Loaded Job command returns the name and modified status of the currently loaded file.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4512).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4512).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
fileName[64]	char	10	Name of the currently loaded job.
changed	8u	74	Whether or not the currently loaded job has been changed (1: yes; 0: no).

Get Alignment Reference

The Get Alignment Reference command is used to get the sensor's alignment reference.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4104).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4104).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
reference	32s	10	Alignment reference 0 – Fixed 1 – Dynamic

Set Alignment Reference

The Set Alignment Reference command is used to set the sensor's alignment reference.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4103).
reference	32s	6	Alignment reference 0 – Fixed 1 – Dynamic

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4103).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Clear Alignment

The Clear Alignment command clears sensor alignment.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4102).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4102).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Get Timestamp

The Get Timestamp command retrieves the sensor's timestamp, in clock ticks. All devices in a system are synchronized with the system clock; this value can be used for diagnostic purposes, or used to synchronize the start time of the system.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x100A).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x100A).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
timestamp	64u	10	Timestamp, in clock ticks.

Get Encoder

This command retrieves the current system encoder value.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101C).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101C).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
encoder	64s	10	Current encoder position, in ticks.

Reset Encoder

The Reset Encoder command is used to reset the current encoder value.

> The encoder value can be reset only when the encoder is connected directly to a sensor. When the encoder is connected to the master, the value cannot be reset via this command.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101E).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101E).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Start

The Start command starts the sensor system (system enters the Running state). For more information on states, see *Control Commands* on page 656.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x100D).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.

Field	Type	Offset	Description
id	16u	4	Reply identifier (0x100D).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Scheduled Start

The scheduled start command starts the sensor system (system enters the Running state) at target time or encoder value (depending on the trigger mode). For more information on states, see *Control Commands* on page 656.

Command

Field	Type	Offset	Description
length	32u	0	Command size – in bytes.
id	16u	4	Command identifier (0x100F).
target	64s	6	Target scheduled start value (in ticks or μs, depending on the trigger type).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size – in bytes.
id	16u	4	Reply identifier (0x100F).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Stop

The Stop command stops the sensor system (system enters the Ready state). For more information on states, see *Control Commands* on page 656.

Command

Field	Type	Type	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1001).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1001).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Get Auto Start Enabled

The Get Auto Start Enabled command returns whether the system automatically starts after booting.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x452C).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x452C).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
enable	8u	10	0: disabled 1: enabled

Set Auto Start Enabled

The Set Auto Start Enabled command sets whether the system automatically starts after booting (enters Running state; for more information on states, see *Control Commands* on page 656).

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x452B).
enable	8u	6	0: disabled 1: enabled

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x452B).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Get Voltage Settings

The Get Voltage Settings command returns the sensor's voltage and cable length settings.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4539).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4539).
Voltage	16u	10	0: 48 Volts; 1: 24 Volts.
Cable Length	32u	12	0 – 100: Meters

Set Voltage Settings

The Set Voltage Settings command sets the sensor's voltage and cable length settings.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4538).
Voltage	16u	6	0: 48 Volts; 1: 24 Volts.
Cable Length	32u	8	0 – 100: Meters

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4538).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Get Quick Edit Enabled

The Get Quick Edit Enabled command returns whether Quick Edit mode is enabled on the sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4541).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4541).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
Enable	8u	10	0: disabled; 1: enabled.

Set Quick Edit Enabled

The Set Quick Edit Enabled command enables or disables Quick Edit mode on the sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4540).
enable	8u	6	0: disabled; 1: enabled.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4540).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Start Alignment

The Start Alignment command is used to start the alignment procedure on a sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4600).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4600).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
opId	32u	10	Operation ID. Use this ID to correlate the command/reply on the Command channel with the correct Alignment Result message on the Data channel. A unique ID is returned each time the client uses this command.

Start Exposure Auto-set

The Start Exposure Auto-set command is used to start the exposure auto-set procedure on a sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4601).
index	32s	6	Device index of sensor to auto-set. 0 – Main 1-31 – Buddy device

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4601).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
opId	32u	10	Operation ID. Use this ID to correlate the command/reply on the Command channel with the correct Exposure Calibration Result message on the Data channel. A unique ID is returned each time the client uses this command.

Software Trigger

The Software Trigger command causes the sensor to take a snapshot while in software mode and in the Running state.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4510).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4510).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Schedule Digital Output

The Schedule Digital Output command schedules a digital output event. The digital output must be configured to accept software-scheduled commands and be in the Running state.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4518).
index	16u	6	Index of the output (starts from 0).
target	64s	8	Specifies the time (clock ticks) when or position (µm) at which the digital output event should happen. The target value is ignored if ScheduleEnabled is set to false. (**Scheduled** is unchecked in **Digital** in the **Output** panel.) The output will be triggered immediately.
value	8u	16	Specifies the target state: 0 – Set to low (continuous)

Field	Type	Offset	Description
			1 – Set to high (continuous) Ignored if output type is pulsed.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4518).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Schedule Analog Output

The Schedule Analog Output command schedules an analog output event. The analog output must be configured to accept software-scheduled commands and be in the Running state.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4519).
index	16u	6	Index of the output. Must be 0.
target	64s	8	Specifies the time (clock ticks) or position (encoder ticks) of when the event should happen. The target value is ignored if ScheduleEnabled is set to false. (**Scheduled** is unchecked in **Analog** in the **Output** panel.) The output will be triggered immediately.
value	32s	16	Output current (microamperes).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4519).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

> The analog output takes about 75 us to reach 90% of the target value for a maximum change, then roughly another 40 us to settle completely.

Ping

The Ping command can be used to test the control connection. This command has no effect on sensors.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x100E).
timeout	64u	6	Timeout value (microseconds).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x100E).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

> If a non-zero value is specified for timeout, the client must send another ping command before the timeout elapses; otherwise the server would close the connection. The timer is reset and updated with every command.

Reset

The Reset command reboots the Main sensor and any Buddy sensors. All sensors will automatically reset 3 seconds after the reply to this command is transmitted.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4300).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4300).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Backup

The Backup command creates a backup of all files stored on the connected sensor and downloads the backup to the client.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1013).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1013).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
length	32u	10	Data length.
data[length]	byte	14	Data content.

Restore

The Restore command uploads a backup file to the connected sensor and then restores all sensor files from the backup.

> The sensor must be reset or power-cycled before the restore operation can be completed.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1014).
length	32u	6	Data length.
data[length]	byte	10	Data content.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1014).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Restore Factory

The Restore Factory command restores the connected sensor to factory default settings.

> The command erases the non-volatile memory of the main device.

This command has no effect on connected Buddy sensors.

Note that the sensor must be reset or power-cycled before the factory restore operation can be completed.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4301).

Field	Type	Offset	Description
resetIp	8u	6	Specifies whether IP address should be restored to default: 0 – Do not reset IP 1 – Reset IP

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4301).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Get Recording Enabled

The Get Recording Enabled command retrieves whether recording is enabled.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4517).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4517).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
enable	8u	10	0: disabled; 1: enabled.

Set Recording Enabled

The Set Recording Enabled command enables recording for replay later.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4516).
enable	8u	6	0: disabled; 1: enabled.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4516).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Clear Replay Data

The Clear Replay Data command clears the sensors replay data..

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4513).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4513).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Get Playback Source

The Get Playback Source command gets the data source for data playback.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4524).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4524).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
source	32s	10	Source 0 – Live 1 – Replay buffer

Set Playback Source

The Set Playback Source command sets the data source for data playback.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4523).
source	32s	6	Source 0 – Live 1 – Replay buffer

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4523).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Simulate

The Simulate command simulates the last frame if playback source is live, or the current frame if playback source is the replay buffer.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4522).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4522).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
bufferValid	8u	10	Whether or not the buffer is valid.

A reply status of -996 means that the current configuration (mode, sensor type, etc.) does not support simulation.

A reply status of -992 means that the simulation buffer is empty. Note that the buffer can be valid even if the simulation buffer is actually empty due to optimization choices. This scenario means that the simulation buffer would be valid if data were recorded.

Seek Playback

The Seek Playback command seeks to any position in the current playback dataset. The frame is then sent.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4503).
frame	32u	6	Frame index.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.

Field	Type	Offset	Description
id	16u	4	Reply identifier (0x4503).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Step Playback

The Step Playback command advances playback by one frame.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4501).
direction	32s	6	Define step direction 0 – Forward 1 – Reverse

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4501).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

> When the system is running in the Replay mode, this command advances replay data (playback) by one frame. This command returns an error if no live playback data set is loaded. You can use the Copy File command to load a replay data set to _live.rec.

Playback Position

The Playback Position command retrieves the current playback position.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4502).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4502).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
Frame Index	32u	10	Current frame index (starts from 0).
Frame Count	32u	14	Total number of available frames/objects.

Clear Measurement Stats

The Clear Measurement Stats command clears the sensor's measurement statistics.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4526).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4526).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Read Live Log

The Read Live Log command returns an XML file containing the log messages between the passed start and end indexes.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101F).
Start	32u	6	First log to read
End	32u	10	Last log to read

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101F).
status	32s	6	Reply status.
length	32u	10	File length
data[length]	byte	14	XML Log File

Clear Log

The Clear Log command clears the sensor's log.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101D).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101D).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Simulate Unaligned

The Simulate Unaligned command simulates data before alignment transformation.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x452A).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x452A).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Acquire

The Acquire command acquires a new scan.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4528).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4528).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

> The command returns after the scan has been captured and transmitted.

Acquire Unaligned

The Acquire Unaligned command acquires a new scan without performing alignment transformation.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4527).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4527).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

> The command returns after the scan has been captured and transmitted.

Create Model

The Create Model command creates a new part model from the active simulation scan.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4602).
modelName[64]	char	6	Name of the new model (without .mdl extension)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4602).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Detect Edges

The Detect Edges command detects and updates the edge points of a part model.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4604).
modelName[64]	char	6	Name of the model (without .mdl extension)
sensitivity	16u	70	Sensitivity (in thousandths).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4604).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Add Tool

The Add Tool command adds a tool to the live job.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4530).
typeName[64]	char	6	Type name of the tool (e.g., ProfilePosition)
name[64]	char	70	User-specified name for tool instance

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4530).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Add Measurement

The Add Measurement command adds a measurement to a tool instance.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4531).
toolIndex	32u	6	Index of the tool instance the new measurement is added to.
typeName[64]	char	10	Type name of the measurement (for example, X).
name[64]	char	74	User-specified name of the measurement instance.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4531).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

	This command can only be used with dynamic tools (tools with a dynamic list of measurements). The maximum number of instances for a given measurement type can be found in the ToolOptions node. For dynamic tools, the maximum count is greater than one, while for static tools it is one.

Read File (Progressive)

The progressive Read File command reads the content of a file as a stream.

This command returns an initial reply, followed by a series of "continue" replies if the initial reply's *status* field indicates success. The continue replies contain the actual data, and have 0x5000 as their identifier.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4529).
name[64]	char	6	Source file name.

Initial Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4529).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.

Continue Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x5000).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.
size	32u	18	Size of the chunk in bytes.
data[size]	byte	22	Chunk data.

Export CSV (Progressive)

The progressive Export CSV command exports replay data as a CSV stream.

This command returns an initial reply, followed by a series of "continue" replies if the initial reply's *status* field indicates success. The continue replies contain the actual data, and have 0x5000 as their identifier.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4507).

Initial Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4507).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.

Continue Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x5000).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.
size	32u	18	Size of the chunk in bytes.
data[size]	byte	22	Chunk data.

> Only the current surface scan, as determined by the playback position, is exported to the CSV stream.

Export Bitmap (Progressive)

The progressive Export Bitmap command exports replay data as a bitmap stream.

This command returns an initial reply, followed by a series of "continue" replies if the initial reply's *status* field indicates success. The continue replies contain the actual data, and have 0x5000 as their identifier.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4508).
type	32s	6	Data type: 0 – Range or video 1 – Intensity
source	32s	10	Data source to export.

Initial Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4508).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.

Continue Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x5000).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.
size	32u	18	Size of the chunk in bytes.
data[size]	byte	22	Chunk data.

Get Flag

The Get Flag command returns the given flag value as a string.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4533).
name[256]	Char	6	A string representing the flag name whose value is to be retrieved.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4533).
valueLength	32u	10	The length of the string representing the flag's value.
value[valueLength]	Char	14	The value of the flag.

Set Flag

The Set Flag command sets the string value for the given flag name.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4534).
Variablename[256]	Char	6	A string representing the flag name whose value is to be retrieved.
valueLength	32u	262	The length of the flag's value string.
value[valueLength]	Char	266	The string representing the flag's value.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4534).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 652.

Get Runtime Variable Count

The Get Runtime Variable Count command gets the number of runtime variables that can be accessed.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4537).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4537).
status	32s	6	Reply status.
valueLength	32u	10	The count of runtime variables.

Set Runtime Variables

The Set Runtime Variables command sets the runtime variables at the given index for the given length.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4536).
index	32u	6	The starting index of the variables to set.
length	32u	10	The number of values to set from the starting index.
values[length]	32s	14	The runtime variable values to set.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4536).
status	32s	6	Reply status.

Get Runtime Variables

The Get Runtime Variables command gets the runtime variables for the given index and length.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4535).
index	32u	6	The starting index of the variables to retrieve.
length	32u	10	The number of values to retrieve from the starting index.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4535).
status	32s	6	Reply status.
index	32u	10	The starting index of the variables being returned.
length	32u	14	The number of values being returned.
values[length]	32s	18	The runtime variable values.

Upgrade Commands

A client sends firmware upgrade commands over the Upgrade TCP channel (port 3192).

The Control channel (port 3190) and the Upgrade channel can be connected simultaneously. For more information on Control commands, see *Control Commands* on page 656.

After connecting to a sensor, you can use the Protocol Version command to retrieve the protocol version. Protocol version refers to the version of the Gocator Protocol supported by the *connected sensor* (the sensor to which a command connection is established), and consists of major and minor parts. The minor part is updated when backward-compatible additions are made to the protocol. The major part is updated when breaking changes are made to the protocol.

Start Upgrade

The Start Upgrade command begins a firmware upgrade for the sensors in a system. All sensors automatically reset 3 seconds after the upgrade process is complete.

Command

Field	Type	Offset	Description
length	64s	0	Command size including this field, in bytes.
id	64s	8	Command identifier (0x0000).
length	64s	16	Length of the upgrade package (bytes).
data[length]	byte	24	Upgrade package data.

Reply

Field	Type	Offset	Description
length	64s	0	Reply size including this field, in bytes.
id	64s	8	Reply identifier (0x0000).
status	64s	16	Reply status. For a list of status codes, see *Commands* on page 652.

Start Upgrade Extended

The Start Upgrade Extended command begins a firmware upgrade for the sensors in a system. All sensors automatically reset 3 seconds after the upgrade process is complete.

Command

Field	Type	Offset	Description
length	64s	0	Command size including this field, in bytes.
id	64s	8	Command identifier (0x0003).
skipValidation	64s	16	Whether or not to skip validation (0 – do not skip, 1 – skip).
length	64s	24	Length of the upgrade package (bytes).
data[length]	byte	32	Upgrade package data.

Reply

Field	Type	Offset	Description
length	64s	0	Reply size including this field, in bytes.
id	64s	8	Reply identifier (0x0003).
status	64s	16	Reply status. For a list of status codes, see *Commands* on page 652.

Get Upgrade Status

The Get Upgrade Status command determines the progress of a firmware upgrade.

Command

Field	Type	Offset	Description
length	64s	0	Command size including this field, in bytes.
id	64s	8	Command identifier (0x1)

Reply

Field	Type	Offset	Description
length	64s	0	Reply size including this field, in bytes.
id	64s	8	Reply identifier (0x1).
status	64s	16	Reply status. For a list of status codes, see *Commands* on page 652.
state	64s	24	Upgrade state: -1 – Failed 0 – Completed 1 – Running 2 – Completed, but should run again
progress	64s	32	Upgrade progress (valid when in the Running state)

Get Upgrade Log

The Get Upgrade Log command can retrieve an upgrade log in the event of upgrade problems.

Command

Field	Type	Offset	Description
length	64s	0	Command size including this field, in bytes.
id	64s	8	Command identifier (0x2)

Reply

Field	Type	Offset	Description
length	64s	0	Reply size including this field, in bytes.
id	64s	8	Reply identifier (0x2).
status	64s	16	Reply status. For a list of status codes, see *Commands* on page 652.
length	64s	24	Length of the log (bytes).
log[length]	char	32	Log content.

Results

The following sections describe the results (data and health) that a sensor sends.

Data Results

A client can receive data messages from a sensor by connecting to the Data TCP channel (port 3196).

The Data channel and the Health channel (port 3194) can be connected at the same time. The sensor accepts multiple connections on each port. For more information on the Health channel, see *Health Results* on page 708.

Messages that are received on the Data and Health channels use a common structure, called Gocator Data Protocol (GDP). Each message consists of a 6-byte header, containing *size* and *control* fields, followed by a variable-length, message-specific content section. The structure of the GDP message is defined below.

Gocator Data Protocol

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last Message flag Bits 0-14: Message type identifier. (See individual data result sections.)

Messages are always sent in groups. The Last Message flag in the *control* field is used to indicate the final message in a group. If there is only one message per group, this bit will be set in each message.

Stamp

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 1.
count (C)	32u	6	Count of stamps in this message.
size	16u	10	Stamp size, in bytes (min: 56, current: 56).
source	8u	12	Source (0 – Main).
reserved	8u	13	Reserved.
stamps[C]	Stamp	14	Array of stamps (see below).

Stamp

Field	Type	Offset	Description
frameIndex	64u	0	Frame index (counts up from zero).
timestamp	64u	8	Timestamp (µs).
encoder	64s	16	Current encoder value (ticks).
encoderAtZ	64s	24	Encoder value latched at z/index mark (ticks).
status	64u	32	Bit field containing various frame information: Bit 0: sensor digital input state Bit 4: master digital input state Bit 8-9: inter-frame digital pulse trigger. (Master digital input if master is connected, otherwise sensor digital input. Value is cleared after each frame and clamped at 3 if more than 3 pulses are received).
serialNumber	32u	40	Sensor serial number. (In a dual-sensor system, the serial number of the main sensor.)
reserved[3]	32u	44	Reserved.
ptpTimestamp	64u	56	PTP Timestamp (µs).

Video

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 2.
attributesSize	16u	6	Size of attributes, in bytes (min: 20, current: 20).
height (H)	32u	8	Image height, in pixels.
width (W)	32u	12	Image width, in pixels.
pixelSize	8u	16	Pixel size, in bytes.
pixelFormat	8u	17	Pixel format: 1 – 8-bit greyscale 2 – 8-bit color filter 3 – 8-bits-per-channel color (B, G, R, X)
colorFilter	8u	18	Color filter array alignment: 0 – None 1 – Bayer BG/GR 2 – Bayer GB/RG 3 – Bayer RG/GB 4 – Bayer GR/BG
source	8u	19	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right 100 to 131 – G2 buddy sensor device indices for configurations with 2 to 31 buddy G2 sensors to identify a particular sensor’s scan data. Main sensor is 100. First buddied sensor is 101. Second buddied sensor is 102 and so on.
cameraIndex	8u	20	Camera index.
exposureIndex	8u	21	Exposure index.
exposure	32u	22	Exposure (ns).
flippedX	8u	26	Indicates whether the video data must be flipped horizontally to match up with profile data.
flippedY	8u	27	Indicates whether the video data must be flipped vertically to match up with profile data.
streamStep	32s	28	Data stream step number. For video, values are: 0 – video stream step

Field	Type	Offset	Description
			8 – tool data stream step
streamStepId	32s	32	Data stream step identifier within the stream step.
transposed	8u	36	Indicates whether the video data must be transposed to match up with profile data.
pixels[H][W]	(Variable)	37	Image pixels. (Depends on pixelSize above.)

Uniform Surface

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 8.
attributeSize	16u	6	Size of attributes, in bytes (min: 44, current: 68).
length (L)	32u	8	Surface length (rows).
length (W)	32u	12	Surface width (columns).
xScale	32u	16	X scale (nm).
yScale	32u	20	Y scale (nm).
zScale	32u	24	Z scale (nm).
xOffset	32s	28	X offset (µm).
yOffset	32s	32	Y offset (µm).
zOffset	32s	36	Z offset (µm).
source	8u	40	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
exposure	32u	41	Exposure (ns).
reserved[7]	8u	45	Reserved.
streamStep	32s	52	Data stream step number. For a surface, values are: 3 – surface stream step 8 – tool data stream step
streamStepId	32s	56	Data stream step identifier within the stream step.
Reserved	32s	60	Reserved
Reserved	32s	64	Reserved
ranges[L][W]	16s	68	Surface ranges.

Surface Point Cloud

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 28.
attributeSize	16u	6	Size of attributes, in bytes (min: 44, current: 60).
length (L)	32u	8	Surface length (rows).
length (W)	32u	12	Surface width (columns).
xScale	32u	16	X scale (nm).
yScale	32u	20	Y scale (nm).
zScale	32u	24	Z scale (nm).
xOffset	32s	28	X offset (µm).
yOffset	32s	32	Y offset (µm).
zOffset	32s	36	Z offset (µm).
source	8u	40	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
exposure	32u	41	Exposure (ns).
isAdjacent	Bool	45	Is the data Adjacant/Sorted? (That is, graphable?)
streamStep	32s	46	Data stream step number. For a surface, values are: 3 – surface stream step 8 – tool data stream step
streamStepId	32s	50	Data stream step identifier within the stream step.
Reserved	32s	54	Reserved
Reserved	32s	56	Reserved
ranges[L][W]	Point3d16s	60	Surface ranges. Tuple (x, y, z) 16s

Surface Intensity

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 9.
attributeSize	16u	6	Size of attributes, in bytes (min: 32, current: 56).
length (L)	32u	8	Surface length (rows).

Field	Type	Offset	Description
width (W)	32u	12	Surface width (columns).
xScale	32u	16	X scale (nm).
yScale	32u	20	Y scale (nm).
xOffset	32s	24	X offset (µm).
yOffset	32s	28	Y offset (µm).
source	8u	32	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
exposure	32u	33	Exposure (ns).
reserved[3]	8u	37	
streamStep	32s	40	Data stream step number. For surface, values are: 3 – surface stream step 8 – tool data stream step
streamStepId	32s	44	Data stream step identifier within the stream step.
Reserved	32s	48	Reserved
Reserved	32s	52	Reserved.
intensities[H][W]	8u	56	Surface intensities.

Surface Section

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 20.
attributeSize	16u	6	Size of attributes, in bytes (min: 45, current: 61).
count (C)	32u	8	Number of profile arrays.
width (W)	32u	12	Number of points per profile array.
xScale	32u	16	X scale (nm).
zScale	32u	20	Z scale (nm).
xOffset	32s	24	X offset (µm).
zOffset	32s	28	Z offset (µm).
source	8u	32	Source 0 – Top 1 – Bottom 2 – Top Left

Field	Type	Offset	Description
			3 – Top Right
sectionId	32u	33	Section Id
exposure	32u	37	Exposure (ns).
poseAngle	32s	41	Z angle of the pose (microdegrees).
poseX	32s	45	X offset of the pose (µm)
poseY	32s	49	Y offset of the pose (µm)
streamStep	32s	53	Stream step.
streamStepId	32s	57	Stream step ID.
ranges[C][W]	16s	61	Profile ranges.

The pose can be used to transform the section data into the surface frame of reference, via a rotation and then a translation.

Surface Section Intensity

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 21.
attributesSize	16u	6	Size of attributes, in bytes (min: 37, current: 53).
count (C)	32u	8	Number of profile intensity arrays
width (W)	32u	12	Number of points per profile intensity array
xScale	32u	16	X scale (nm).
xOffset	32s	20	X offset (µm).
source	8u	24	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
sectionId	32u	25	Section Id.
exposure	32u	29	Exposure (ns).
poseAngle	32s	33	Z angle of the pose (microdegrees).
poseX	32s	37	X offset of the pose (µm).
poseY	32s	41	Y offset of the pose (µm).
streamStep	32s	45	Stream step.
streamStepId	32s	49	Stream step ID.
points[C][W]	8u	53	Intensity arrays.

Measurement

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 10.
count (C)	32u	6	Count of measurements in this message.
reserved[2]	8u	10	Reserved.
id	16u	12	Measurement identifier.
measurements[C]	Measurement	14	Array of measurements (see below).

Measurement

Field	Type	Offset	Description
value	32s	0	Measurement value.
decision	8u	4	Measurement decision bitmask. Bit 0: 1 – Pass 0 – Fail Bits 1-7: 0 – Measurement value OK 1 – Invalid value 2 – Invalid anchor
reserved[3]	8u	5	Reserved.

Alignment Result

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 11.
attributesSize	16u	6	Size of attributes, in bytes (min: 8, current: 16).
opId	32u	8	Operation ID.
status	32s	12	Operation status. 1 – OK 0 – General failure -1 – No data in the field of view for stationary alignment -2 – No profiles with sufficient data for line fitting for travel alignment -3 – Invalid target detected. Examples include:

Field	Type	Offset	Description
			- Calibration disk diameter too small. - Calibration disk touches both sides of the field of view. - Too few valid data points after outlier rejection. -4 – Target detected in an unexpected position. -5 – No reference hole detected in bar alignment. -6 – No change in encoder value during travel calibration -988 – User aborted -993 – Timed out -997 – Invalid parameter

Exposure Calibration Result

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 12.
attributesSize	16u	6	Size of attributes, in bytes (min: 8, current: 16).
opId	32u	8	Operation ID.
status	32s	12	Operation status.
exposure	32u	16	Exposure result (ns).

Edge Match Result

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 16.
decision	8u	6	Overall match decision.
xOffset	32s	7	Target x offset in model space (μm).
yOffset	32s	11	Target y offset in model space (μm).
zAngle	32s	15	Target z rotation in model space (microdegrees).
quality	32s	19	Match quality (thousandth).
qualityDecision	8u	23	Quality match decision.
reserved[2]	8u	24	Reserved.

Bounding Box Match Result

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 17.
decision	8u	6	Overall match decision.
xOffset	32s	7	Target x offset in model space (µm).
yOffset	32s	11	Target y offset in model space (µm).
zAngle	32s	15	Target z rotation in model space (microdegrees).
width	32s	19	Width axis length (µm)
widthDecision	8u	23	Width axis decision.
length	32s	24	Length axis length (µm)
lengthDecision	8u	28	Length axis decision.

Ellipse Match Result

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 18.
decision	8u	6	Overall match decision.
xOffset	32s	7	Target x offset in model space (µm).
yOffset	32s	11	Target y offset in model space (µm).
zAngle	32s	15	Target z rotation in model space (microdegrees).
minor	32s	19	Minor axis length (µm)
minorDecision	8u	23	Minor axis decision.
major	32s	24	Major axis length (µm)
majorDecision	8u	28	Major axis decision.

Event

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 22.
attributesSize	16u	6	Size of attributes, in bytes (min: 8, current: 16).
eventType	32u	8	The type of event: 0 – Exposure Begin 1 – Exposure End

Field	Type	Offset	Description
length	32u	12	The number of bytes containing additional data.
data[length]	8u	16	Additional data.

Feature Point

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 24.
id	16u	6	Feature Id
Point.x	64s	8	X Coordinate of Point (Scaled by 10^6)
Point.y	64s	16	Y Coordinate of Point (Scaled by 10^6)
Point.z	64s	24	Z Coordinate of Point (Scaled by 10^6)

Feature Line

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 25.
id	16u	6	Feature Id
Point.x	64s	8	X Coordinate of Point (Scaled by 10^6)
Point.y	64s	16	Y Coordinate of Point (Scaled by 10^6)
Point.z	64s	24	Z Coordinate of Point (Scaled by 10^6)
Direction.x	64s	32	X Component of Direction Vector (Scaled by 10^6)
Direction.y	64s	40	Y Component of Direction Vector (Scaled by 10^6)
Direction.z	64s	48	Z Component of Direction Vector (Scaled by 10^6)

Feature Plane

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 26.
id	16u	6	Feature Id
Normal.x	64s	8	X Component of Normal Vector (Scaled by 10^6)
Normal.y	64s	16	Y Component of Normal Vector (Scaled by 10^6)
Normal.z	64s	24	Z Component of Normal Vector (Scaled by 10^6)
originDistance	64s	32	Distance to Origin (Scaled by 10^6)

Feature Circle

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 27.
id	16u	6	Feature Id
Point.x	64s	8	X Coordinate of Point (Scaled by 10^6)
Point.y	64s	16	Y Coordinate of Point (Scaled by 10^6)
Point.z	64s	24	Z Coordinate of Point (Scaled by 10^6)
Normal.x	64s	32	X Component of Normal Vector (Scaled by 10^6)
Normal.y	64s	40	Y Component of Normal Vector (Scaled by 10^6)
Normal.z	64s	48	Z Component of Normal Vector (Scaled by 10^6)
radius	64s	56	Radius of Circle (Scaled by 10^6)

Generic Message

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 29.
attributeSize	16u	6	Size of attributes, in bytes (min: 32, current: 40).
streamStep	32s	8	Data stream step.
streamStepId	32s	12	Data stream step ID.
userType	32u	16	User-define data type ID
isObject	8u	20	0 – Content is raw byte buffer 1 – Content is an kObject
contentLength	32u	21	Length of content array, in bytes
Content[contentLength]	byte	25	Content array. If isObject is true, the byte buffer should be deserialized using kDat6Serializer.

Null Message

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 30.
attributeSize	16u	6	Size of attributes, in bytes (min: 20, current: 20).
errorStatus	32s	8	Error status related to the message

Field	Type	Offset	Description
reserved	32s	12	Reserved
reserved	32s	16	Reserved

Mesh

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. For this message, set to 36.
attributesSize	16u	6	Size of attributes, in bytes (min: 178 current: 178).
source	8u	8	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right 100 to 131 – G2 buddy sensor device indices for configurations with 2 to 31 buddy G2 sensors to identify a particular sensor's scan data. Main sensor is 100. First buddied sensor is 101. Second buddied sensor is 102 and so on.
streamStep	32s	9	Stream step.
streamStepId	32s	13	Stream ID.
hasData	8u	17	Indicator whether any data are allocated.
systemChannelCount	32s	18	Number of system channels (currently 6).
maxUserChannelCount	32s	22	Maximum number of user channel supported (currently 5).
userChannelCount	32s	26	Number of active user channels currently.
channelCount	32s	30	Number of all channels, including all system channels and active user channels.
offset	Point3d64f	34	Offset of mesh data.
range	Point3d64f	58	Range of mesh data.
transformation	Transform3d64f	82	Transformation of mesh data.
channels[channelCount]	Private mesh channel data	178	Array of private mesh channel data (see below).

Mesh Channel Data

Field	Type	Offset	Description
attributesSize	16u	0	Size of attributes, in bytes (min: 26 current: 26).

Field	Type	Offset	Description
id	32s	2	Channel ID. 0 – Vertex channel 1 – Facet channel 2 – Facet normal channel 3 – Vertex normal channel 4 – Vertex texture channel 5 – Vertex curvature channel
type	32u	6	Channel type. 0 – Invalid type 1 – Vertex type 2 – Facet type 3 – Facet normal type 4 – Vertex normal type 5 – Vertex texture type 6 – Vertex curvature type
state	32s	10	Channel state. -1 – Error state 0 – Unallocated state 1 – Allocated state 2 – Empty state 3 – Partial used state 4 – Full state
flag	32u	14	Channel flag is an user specified field. It can be used to send any optional data to determine how to deserialize user channel data.
allocateCount	32s	18	Number of allocated channel data items. System channels – Equal to number of items (ie, allocateCount for vertex channel is number of Point3d32f allocated). User channels – Equal to number of bytes (8u) of entire buffer.
usedCount	32s	22	Number of used channel data items.
buffer[allocateCount]	various	26	Data buffer of various types depending on channel ID. Channel ID. 0 – Point3d32f 1 – Facet3d32u 2 – Point3d32f

Field	Type	Offset	Description
			3 – Point3d32f 4 – 8u 5 – 32f 6 or higher – User is responsible to use channel type and/or channel flag to properly deserialize buffer into appropriate buffer item type.

Health Results

A client can receive health messages from a sensor by connecting to the Health TCP channel (port 3194).

The Data channel (port 3196) and the Health channel can be connected at the same time. The sensor accepts multiple connections on each port. For more information on the Data channel, see *Data Results* on page 694.

Messages that are received on the Data and Health channels use a common structure, called Gocator Data Protocol (GDP). Each message consists of a 6-byte header, containing *size* and *control* fields, followed by a variable-length, message-specific content section. The structure of the GDP message is defined below.

Gocator Data Protocol

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last Message flag Bits 0-14: Message type identifier. (See individual data result sections.)

Messages are always sent in groups. The Last Message flag in the *control* field is used to indicate the final message in a group. If there is only one message per group, this bit will be set in each message.

A Health Result contains a single data block for health *indicators*. Each indicator reports the current status of some aspect of the sensor system, such as CPU usage or network throughput.

Health Result

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier. Always 0.
count (C)	32u	6	Count of indicators in this message.
source	8u	10	Source (0 – Main, 1 – Buddy).
reserved[3]	8u	11	Reserved
indicators[C]	Indicator	14	Array of indicators (see format below).

The indicators block contains a 2-dimensional array of indicator data. Each row in the array has the following format:

Indicator Format

Field	Type	Offset	Description
id	32u	0	Unique indicator identifier (see *Indicator identifiers* below table below).
instance	32u	4	Indicator instance.
value	64s	8	Value (identifier-specific meaning).

The following health indicators are defined for sensor systems.

When a sensor is accelerated, some health indicators report values from the *PC* that is accelerating the sensor, or a combination of both. In the table below, values are reported from the sensor unless otherwise indicated.

Undocumented indicators may be included in addition to the indicators defined below.

Indicator identifiers

Indicator	ID	Instance	Value
Encoder Value	1003	-	Current system encoder tick.
Encoder Frequency	1005	-	Current system encoder frequency (ticks/s).
App Version	2000	-	Firmware application version.
Internal Temperature	2002	-	Internal temperature (centidegrees Celsius).
Uptime	2017	-	Time elapsed since node boot-up or reset (seconds).
Projector Temperature	2404	-	Projector module temperature (centidegrees Celsius). Only available on projector based devices.
Control Temperature	2028	-	Control module temperature (centidegrees Celsius). Available only on 3B-class devices.
Memory Usage	2003	-	Amount of memory currently used (bytes).
Memory Capacity	2004	-	Total amount of memory available (bytes).
Storage Usage	2005	-	Amount of non-volatile storage used (bytes).
Storage Capacity	2006	-	Total amount of non-volatile storage available (bytes).
Alignment State	20008	-	Alignment state: 0 – not aligned 1 - aligned
CPU Usage	2007	-	CPU usage (percentage of maximum).

Indicator	ID	Instance	Value
Net Out Capacity	2009	-	Total available outbound network throughput (bytes/s).
Net Out Link Status	2034	-	Current Ethernet link status.
Sync Source*	2043	-	Synchronization source. 1 - Master device 2 - Sensor
Digital Inputs*	2024	-	Current digital input status (one bit per input).
Event Count	2102	-	Total number of events triggered.
Camera Trigger Drops	2201	-	Number of dropped triggers.
Analog Output Drops	21014 (previously 2501)	Output Index	Number of dropped outputs.
Digital Output Drops	21015 (previously 2601)	Output Index	Number of dropped outputs.
Serial Output Drops	21016 (previously 2701)	Output Index	Number of dropped outputs.
Platform CUDA Status	3007	-	Status of CUDA/GPU support on the sensor (accelerated and non-accelerated) platform. 0 = CUDA/GPU execution supported in current platform environment.
Sensor State*	20000	-	Sensor state. -1 – Conflict 0 – Ready 1 – Running
Current Sensor Speed*	20001	-	Current sensor speed. (Hz)
Maximum Speed*	20002	-	The sensor’s maximum speed.
Spot Count*	20003	-	Number of spots found in the last unresampled profile/surface.
Max Spot Count*	20004	-	Maximum number of spots that can be found.
Scan Count*	20005	-	Number of surfaces detected from a top device.
Master Status*	20006	0 for main 1 for buddy	Master connection status: 0 – Not connected 1 – Connected The indicator with instance = buddy does not exist if the buddy is not connected.
Cast Start State*	20007		The state of the second digital input. (NOTE: Only

Indicator	ID	Instance	Value
			available on XLine capable licensed devices)
Point Count	20015	-	Number of points found in last resampled Profile/Surface.
Max Point Count	20016	-	Maximum number of points that can be found.
Laser Overheat*	20020	-	Indicates whether laser overheat has occurred. 0 – Has not overheated 1 – Has overheated Only available on certain 3B laser devices.
Laser Overheat Duration*	20021	-	The length of time in which the laser overheating state occurred. Only available on certain 3B laser devices.
Playback Position*	20023	-	The current replay playback position.
Playback Count*	20024	-	The number of frames present in the replay.
FireSync Version	20600	-	The FireSync version used by the Gocator build. The low-level firmware version used by the sensor.
Processing Drops**	21000	-	Number of dropped frames. The sum of various processing drop related indicators.
Last Processing Latency	21001	-	Last delay from camera exposure to availability of all results.
Max Processing Latency	21002	-	Maximum value of processing latency.
Ethernet Output	21003	-	Number of bytes transmitted.
Ethernet Rate	21004	-	The average number of bytes per second being transmitted.
Ethernet Drops	21005	-	Number of dropped Ethernet packets.
Digital Output Pass	21006	Output Index	Number of pass digital output pulse.
Digital Output Fail	21007	Output Index	Number of fail digital output pulse.
Trigger Drops**	21010		Number of dropped triggers. The sum of various triggering-related drop indicators.
Output Drops**	21011		Number of dropped output data. The sum of all output drops (analog, digital, serial, host server, and ASCII server).
Controlled Trigger Drops	21017		Trigger drops from the Controlled Triggering System (Grouped with “Trigger Drops” indicator)
Surface Processing Time	21018		Processing time of frame on 35xx/32xx (microseconds)
Max Frame Rate	21019		32xx/35xx max configurable frame rate given above in Surface Processing Time (scaled by 1x10-

Indicator	ID	Instance	Value
			6)
Range Valid Count**	21100	-	Number of valid ranges.
Range Invalid Count**	21101	-	Number of invalid ranges.
Anchor Invalid Count**	21200	-	Number of frames with anchoring invalid.
Light Operational Time	21201	-	Total running time of G2 laser or G3 projector light (on Gocator firmware 5.3 or later), in minutes.
First Log Id	21301		ID of the first available log entry.
Last Log Id	21300		ID of the last available log entry. It is inclusive: for example, if first = 3 and last = 5, the available log IDs are 3, 4, 5. If no log is available, the last ID is less than the first ID.
Z-Index Drop Count	22000	-	The number of dropped surfaces due to a lack of z-encoder pulse during rotational part detection.
Tool Run Time	22004	Tool Index	The most recent time taken to execute the tool.
Part Total Emitted	22006	-	Total number of parts emitted by profile part detection.
Part Length Limit	22007	-	Number of parts emitted due to reaching the length limit.
Part Min Area Drops	22008	-	Number of parts dropped due to being smaller than the minimum area.
Part Backtrack Drops	22009	-	Number of parts dropped due to backtracking.
Parts Currently Active	22010	-	Number of parts currently being tracked.
Part Length	22011	-	Length of largest active part.
Part Start Y	22012	-	Start Y position of the largest active part.
Part Tracking State	22013	-	Tracking state of the largest active part.
Part Capacity Exceeded	22014	-	Part detection part or run capacity has been exceeded.
Part X Position	22015	-	Center X position of the largest active part.
Tool Runtime Minimum	22016	-	Minimum time spent for tool to process a sample
Tool Runtime Maximum	22017	-	Maximum time spent for tool to process a sample
Tool Runtime Average	22018	-	Average time for tool to process a sample
Tool Runtime Percent Average	22019	-	Average percentage of total time spent running this tool
Bar Alignment Status	22020	-	Status of the buffered bar alignment when aligning: 1 – buffer leveling in progress

Indicator	ID	Instance	Value
			2 – buffer searching in progress 3 – buffer scanning in progress 4 – buffer padding in progress 5 – buffering complete; processing alignment on buffered data 11 – alignment leveling in progress 12 – alignment searching in progress 13 – alignment fitting in progress 14 – alignment complete 15 – alignment completed but failed 16 – alignment cancelled
Value	30000	Measurement ID	Measurement Value.
Pass	30001	Measurement ID	Number of pass decision.
Fail	30002	Measurement ID	Number of fail decision.
Min	30003	Measurement ID	Minimum measurement value.
Max	30004	Measurement ID	Maximum measurement value.
Average	30005	Measurement ID	Average measurement value.
Std. Dev.	30006	Measurement ID	Measurement value standard deviation.
Invalid Count	30007	Measurement ID	Number of invalid values.
Overflow	30008	Measurement ID	Number of times this measurement has overflown on any output. Multiple simultaneous overflows result in only a single increment to this counter. Overflow conditions include: -Value exceeds bit representation available for given protocol -Analog output (mA) falls outside of acceptable range (0-20 mA) When a measurement value overflow occurs, the value is set to the null value appropriate for the given protocol's measurement value output type. The Overflow health indicator increments.

* When the sensor is accelerated, the indicator's value is reported from the accelerating PC.

** When the sensor is accelerated, the indicator's value is the sum of the values reported from the sensor and the accelerating PC.

Modbus Protocol

Modbus is designed to allow industrial equipment such as Programmable Logic Controllers (PLCs), sensors, and physical input/output devices to communicate over an Ethernet network.

Modbus embeds a Modbus frame into a TCP frame in a simple manner. This is a connection-oriented transaction, and every query expects a response.

This section describes the Modbus TCP commands and data formats. Modbus TCP communication lets the client:

- Switch jobs.
- Align and run sensors.
- Receive measurement results, sensor states, and stamps.

To use the Modbus protocol, it must be enabled and configured in the active job. For information on configuring the protocol using the Web interface, see *Ethernet Output* on page 512.

> The Gocator 4.x/5.x firmware uses mm, mm^2, mm^3, and degrees as standard units. In all protocols, values are scaled by 1000, as values in the protocols are represented as integers. This results in effective units of mm/1000, mm^2/1000, mm^3/1000, and deg/1000 in the protocols.

If buffering is enabled with the Modbus protocol, the PLC must read the Buffer Advance output register (see *State* on page 717) to advance the queue before reading the measurement results.

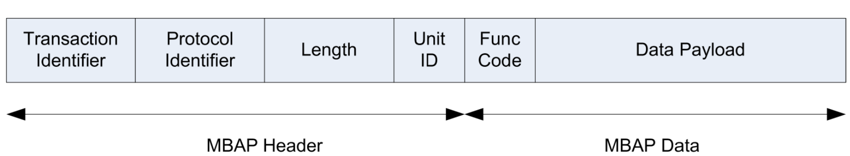
Concepts

A PLC sends a command to start each sensor. The PLC then periodically queries each sensor for its latest measurement results. In Modbus terminology, the PLC is a Modbus Client. Each sensor is a Modbus Server which serves the results to the PLC.

The Modbus protocol uses TCP for connection and messaging. The PLC makes a TCP connection to the sensor on port 502. Control and data messages are communicated on this TCP connection. Up to eight clients can be connected to the sensor simultaneously. A connection closes after 10 minutes of inactivity.

Messages

All Modbus TCP messages consist of an MBAP header (Modbus Application Protocol), a function code, and a data payload.



The MBAP header contains the following fields:

Modbus Application Protocol Header

Field	Length (Bytes)	Description
Transaction ID	2	Used for transaction pairing. The Modbus Client sets the value and the Server (the sensor) copies the value into its responses.
Protocol ID	2	Always set to 0.
Length	2	Byte count of the rest of the message, including the Unit identifier and data fields.
Unit ID	1	Used for intra-system routing purpose. The Modbus Client sets the value and the Server (the sensor) copies the value into its responses.

Modbus Application Protocol Specification describes the standard function codes in detail. Gocator supports the following function codes:

Modbus Function Code

Function Code	Name	Data Size (bits)	Description
3	Read Holding Registers	16	Read multiple data values from the sensor.
4	Read Input Registers	16	Read multiple data values from the sensor.
6	Write Single Register	16	Send a command or parameter to the sensor.
16	Write Multiple Registers	16	Send a command and parameters to the sensor.

The data payload contains the registers that can be accessed by Modbus TCP messages. If a message accesses registers that are invalid, a reply with an exception is returned. Modbus Application Protocol Specification defines the exceptions and describes the data payload format for each function code.

The sensor data includes 16-bit, 32-bit, and 64-bit data. All data are sent in big endian format, with the 32-bit and 64-bit data spread out into two and four consecutive registers.

32-bit Data Format

Register	Name	Bit Position
0	32-bit Word 1	31 .. 16
1	32-bit Word 0	15 .. 0

64-bit Data Format

Register	Name	Bit Position
0	64-bit Word 3	63 .. 48
1	64-bit Word 2	47 .. 32
2	64-bit Word 1	31 .. 16
3	64-bit Word 0	15 .. 0

Registers

Modbus registers are 16 bits wide and are either control registers or output registers.

Control registers are used to control the sensor states (e.g., start, stop, or calibrate a sensor).

The output registers report the sensor states, stamps, and measurement values and decisions. You can read multiple output registers using a single Read Holding Registers or a single Read Input Registers command. Likewise, you can control the state of the sensor using a single Write Multiple Register command.

Control registers are write-only, and output registers are read-only.

Register Map Overview

Register Address	Name	Read/Write	Description
0 - 124	Control Registers	WO	Registers for Modbus commands. See *Control Registers* below for detailed descriptions.
300 - 899	Sensor States	RO	Report sensor states. See *State* on the next page for detailed descriptions.
900 - 999	Stamps	RO	Return stamps associated with each surface. See *State* on the next page for detailed descriptions.
1000 - 1998	Measurements & Decisions	RO	333 measurement and decision pairs. See *Measurement Registers* on page 719 for detailed descriptions.

Control Registers

Control registers are used to operate the sensor. Register 0 stores the command to be executed. Subsequent registers contain parameters for the commands if applicable. The sensor executes a command when the value in register 0 is changed. To set the parameters before a command is executed, you should set up the parameters and the command using a single Multiple Write register command.

Control Register Map

Register Address	Name	Read/Write	Description
0	Command Register	WO	Takes a 16-bit command. For a list of the available commands, see table below.
1 – 64	Command Parameters	WO	For **Load Job** (5) command: Null-terminated filename. Each 16-bit register holds a single character. Specifies the filename. If the file extension ".job" is missing, it is automatically appended to the filename. For **Set Runtime Variables** (6) command: Registers 1-8 are used to set the values of the runtime variables.

The 16-bit values used for Command Register are described below.

Command Register Values

Value	Name	Description
0	Stop Running	Stops the sensor. No effect if sensor is already stopped.
1	Start Running	Starts the sensor. No effect if sensor is already started.
2	Align (stationary target)	Starts the stationary alignment process. State register 301 will be set to 1 (busy). When the alignment process is complete, the register is set back to zero.
3	Align (moving target)	Starts moving alignment process and also calibrate encoder resolution. State register 301 will be set to 1 (busy). When the alignment process is complete, the register is set back to zero.
4	Clear Alignment	Clears the alignment.
5	Load Job	Activates the specified job file. Set registers 1-64 to the null-terminated filename, one filename character per 16-bit register, including the null terminator character. The ".job" extension is optional; if it is missing, it is automatically appended to the file name.
6	Set Runtime Variables	Sets the runtime variables. Set registers 1 through 8 to the values of all four 32-bit runtime variables.
7	Software trigger	Software trigger the sensor to capture one frame. The sensor must already be running, in trigger mode “Software”. Otherwise, software trigger has no effect.

Output Registers

Output registers are used to output states, stamps, and measurement results. Each register address holds a 16-bit data value.

State

State registers report the current sensor state.

State Register Map

Register Address	Name	Type	Description
300	Sensor State	16u	Sensor State: 0 - Stopped 1 - Running
301	Modbus Command in Progress	16u	1 when the sensor is busy performing the last command, 0 when done. Registers 302 and 311-371 below are only valid when there is no command in progress.
302	Alignment State	16u	Current Alignment State: 0 - Not aligned 1- Aligned

Register Address	Name	Type	Description
			(Valid when register 301 = 0.)
303	Encoder Position High	64u	Current encoder position (64-bit value, requiring four 16-bit registers)
304	Encoder		
305	Encoder		
306	Encoder Low		
307	Time High	64s	Uptime timestamp (64-bit value, requiring four 16-bit registers)
308	Time		
309	Time		
310	Time Low		
311	Job File Name Length	16u	Number of characters in the current job file name. (Valid when register 301 = 0.)
312 – 371	Live Job Name	16u	Name of currently loaded job file. Does not include the extension. Each 16-bit register contains a single character. (Valid when register 301 = 0.)
375	Runtime Variable 0 High	32s	Runtime variable value stored in two register locations.
376	Runtime Variable 0 Low		
...	...	...	...
381	Runtime Variable 3 High	32s	Runtime variable value stored in two register locations.
382	Runtime Variable 3 Low		

Stamp

Stamps contain trigger timing information used for synchronizing a PLC's actions. A PLC can also use this information to match up data from multiple sensors.

In Surface mode, the stamps are updated after each surface has been processed.

Stamp Register Map

Register Address	Name	Type	Description
960-975	reserved		Not used.
976	Buffer Advance Register	16u	If buffering is enabled, this address must be read by the PLC Modbus client first to advance the buffer. After the buffer advance read operation, the Modbus client can read the updated Measurements & Decisions in addresses 1000-1059.
977	Buffer Count	16u	Number of buffered messages currently in the

Register Address	Name	Type	Description
			queue.
978	Buffer Overflow Flag	16u	Buffer Overflow Indicator: 0 - No overflow 1 - Overflow. (Indicates data is being lost.)
979	Inputs	16u	Digital input state of the last frame.
980	zPosition High	64u	Encoder position at time of last index pulse. 64-bit value, requiring four 16-bit registers.
981	zPosition		
982	zPosition		
983	zPosition Low		
984	Exposure High	32u	Laser exposure (µs) of the last frame. Stored in two register locations.
985	Exposure Low		
986	Temperature High	32u	Sensor temperature in degrees Celcius * 100 (centidegrees) of the last frame. Stored in two register locations.
987	Temperature Low		
988	Encoder Position High	64u	Encoder position of the last frame when the image data was scanned/taken. 64-bit value, requiring four 16-bit registers.
989	Encoder Position		
990	Encoder Position		
991	Encoder Position Low		
992	Time High	64u	Time stamp in microseconds of the last frame. 64-bit value, requiring four 16-bit registers.
993	Time		
994	Time		
995	Time Low		
996	Frame Index High	64u	The frame number of the last frame. 64-bit value, requiring four 16-bit registers.
997	Frame Index		
998	Frame Index		
999	Frame Index Low		

Measurement Registers

Measurement results are reported in pairs of values and decisions. Measurement values are 32 bits wide and decisions are 8 bits wide.

The measurement ID is used to find the register address of each pair. The register address of the first word can be calculated as (1000 + 3 * ID). For example, a measurement with ID set to 4 can be read from registers 1012 (high word) and 1013 (low word), and the decision at 1015.

In Surface mode, the measurement results are updated after each discrete part has been processed.

Measurement Register Map

Register Address	Name	Type	Description
1000	Measurement 0 High	32u	Measurement value in µm (0x80000000 if invalid)
1001	Measurement 0 Low		
1002	Decision 0	16u	Measurement decision. A bit mask, where: Bit 0: 1 - Pass 0 - Fail Bits 1-7: 0 - Measurement value OK 1 - Invalid value 2 - Invalid anchor
1003	Measurement 1 High		
1004	Measurement 1 Low		
1005	Decision 1		
1006	Measurement 2 High		
1007	Measurement 2 Low		
1008	Decision 2		
...	...	...	...
1996	Measurement 332 High		
1997	Measurement 332 Low		
1998	Decision 332		

EtherNet/IP Protocol

EtherNet/IP is an industrial protocol that allows bidirectional data transfer with PLCs. It encapsulates the object-oriented Common Industrial Protocol (CIP). EtherNet/IP communication enables the client to:

- Switch jobs.
- Align and run sensors.
- Receive sensor states, stamps, and measurement results.
- Set and retrieve runtime variables.

This section describes the EtherNet/IP messages and data formats.

Note that in firmware version 5.2, the identity information was updated as follows:

Attribute	Before Firmware 5.2	Firmware 5.2 and later
Product Code	Was 1000, 2000, or 3000 depending on the model.	Now 1.
Major Revision	Matched firmware major version.	Now 1.
Minor Revision	Matched firmware minor version.	Now 1.

This update may require a change on a device attempting to connect to a sensor via EtherNet/IP. A compatible EDS file can be downloaded from the web interface. If the existing EDS must be maintained, the device can be configured to disable electronic keying, ignoring the product code and version numbers.

To use the EtherNet/IP protocol, it must be enabled and configured in the active job. For information on configuring the protocol using the Web interface, see *Ethernet Output* on page 512.

> The Gocator 4.x/5.x firmware uses mm, mm^2, mm^3, and degrees as standard units. In all protocols, values are scaled by 1000, as values in the protocols are represented as integers. This results in effective units of mm/1000, mm^2/1000, mm^3/1000, and deg/1000 in the protocols.

Sensors support unconnected or connected explicit messaging (with TCP), as well as implicit (or I/O) messaging. For information on explicit messaging assemblies and objects, see *Explicit Messaging* below. For information on implicit messaging assemblies and objects, see *Implicit Messaging* on page 728.

Explicit Messaging

To EtherNet/IP-enabled devices on the network, the sensor information is seen as a collection of objects, which have attributes that can be queried.

Sensors support all required objects for explicit messaging, such as the Identity object, TCP/IP object, and Ethernet Link object. In addition, an Assembly object is used for sending sensor and sample data and receiving commands. The Assembly object contains four assemblies: the command assembly (32 bytes), the runtime variable configuration assembly (64 bytes), the sensor state assembly (100 bytes), and the sample state assembly object (380 bytes). The data attribute (0x03) of the assembly objects is a byte array containing information about the sensor. The data attribute can be accessed with the Get Attribute and Set Attribute commands.

The PLC sends a command to start a sensor. The PLC then periodically queries the attributes of the assembly objects for its latest measurement results. In EtherNet/IP terminology, the PLC is a scanner and the sensor is an adapter.

For detailed information on setting up explicit messaging using Allen-Bradley PLCs, see https://downloads.lmi3d.com/setting-ethernetip-explicit-messaging-allen-bradley-plcs.

The following sections describe the explicit messaging assemblies and objects.

Identity Object (Class 0x01)

Attribute	Name	Type	Value	Description	Access
1	Vendor ID	UINT	1256	ODVA-provided vendor ID	Get
2	Device Type	UINT	43	Device type	Get
3	Product Code	UINT	1	Product code	Get
4	Revision	USINT	1.1	Byte 0 - 1 Byte 1 - 1	Get
6	Serial number	UDINT	32-bit value	Sensor serial number	Get
7	Product Name	SHORT STRING 32	"Gocator"	Gocator product name	Get

TCP/IP Object (Class 0xF5)

The TCP/IP Object contains read-only network configuration attributes such as IP Address. TCP/IP configuration via Ethernet/IP is not supported. See Volume 2, Chapter 5-3 of the CIP Specification for a complete listing of TCP/IP object attributes.

Attribute	Name	Type	Value	Description	Access
1	Status	UDINT	0	TCP interface status	Get
2	Configuration Capability	UINT	0		Get
3	Configuration Control	UINT	0	Product code	Get
4	Physical Link Object	Structure (See description)		See 5.3.3.2.4 of CIP Specification Volume 2: Path size (UINT) Path (Padded EPATH)	Get
5	Interface Configuration	Structure (See description)		See 5.3.3.2.5 of CIP Specification Volume 2: IP address (UDINT) Network mask (UDINT)	Get

Attribute	Name	Type	Value	Description	Access
				Gateway address (UDINT) Name server (UDINT) Secondary name (UDINT) Domain name (UDINT)	

Ethernet Link Object (Class 0xF6)

The Ethernet Link Object contains read-only attributes such as MAC Address (Attribute 3). See Volume 2, Chapter 5-4 of the CIP Specification for a complete listing of Ethernet Link object attributes.

Attribute	Name	Type	Value	Description	Access
1	Interface Speed	UDINT	1000	Ethernet interface data rate (mbps)	Get
2	Interface Flags	UDINT		See 5.4.3.2.1 of CIP Specification Volume 2: Bit 0: Link Status 0 – Inactive 1 - Active Bit 1: Duplex 0 – Half Duplex 1 – Full Duplex	Get
3	Physical Address	Array of 6 USINTs		MAC address (for example: 00 16 20 00 2E 42)	Get

Assembly Object (Class 0x04)

For explicit messaging, the Ethernet/IP object model includes the following assemblies: command, runtime variable configuration, sensor state, and sample state.

All assembly object instances are static. Data in a data byte array in an assembly object are stored in the big endian format.

Command Assembly

The command assembly object is used to start, stop, and align the sensor, and also to switch jobs on the sensor.

Command Assembly

Information	Value
Class	0x4
Instance	0x310
Attribute Number	3
Length	32 bytes
Supported Service	0x10 (SetAttributeSingle)

Attributes 1 and 2 are not implemented, as they are not required for the static assembly object.

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array	See Below	Command parameters Byte 0 - Command. See table below for specification of the values.	Get, Set

Command Definitions

Value	Name	Description
0	Stop running	Stop the sensor. No action if the sensor is already stopped
1	Start Running	Start the sensor. No action if the sensor is already started.
2	Stationary Alignment	Start the stationary alignment process. Byte 1 of the sensor state assembly will be set to 1 (busy) until the alignment process is complete, then back to zero.
4	Clear Alignment	Clear the alignment.
5	Load Job	Load the job. Set bytes 1-31 to the file name (one character per byte. File name must be null-terminated. The job name and extension are case-sensitive. If the extension ".job" is missing, it is automatically appended to the file name.
6	Reserved	Do not use.
7	Software trigger	Sends a software trigger to the sensor to capture one frame. The sensor must already be running, and its trigger mode must be set to "Software". Otherwise, software trigger has no effect.

Runtime Variable Configuration Assembly

The runtime variable configuration assembly object contains the sensor's intended runtime variables.

Runtime Variable Configuration Assembly

Information	Value
Class	0x04
Instance	0x311
Attribute Number	3
Length	64 bytes
Supported Service	0x10 (SetAttributeSingle)

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array	See below	Runtime variable configuration information. See below for more details.	Get

Sensor State Information

Byte	Name	Type	Description
0-3	Runtime	32s	Stores the intended value of the Runtime Variable at index 0.

Byte	Name	Type	Description
	Variable 0		
4-7	Runtime Variable 1	32s	Stores the intended value of the Runtime Variable at index 1.
8-11	Runtime Variable 2	32s	Stores the intended value of the Runtime Variable at index 2.
12-15	Runtime Variable 3	32s	Stores the intended value of the Runtime Variable at index 3.
16-63	Reserved		

Sensor State Assembly

The sensor state assembly object contains the sensor's states, such as the current sensor temperature, frame count, and encoder values.

Sensor State Assembly

Information	Value
Class	0x04
Instance	0x320
Attribute Number	3
Length	100 bytes
Supported Service	0x0E (GetAttributeSingle)

Attributes 1 and 2 are not implemented, as they are not required for the static assembly object.

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array	See below	Sensor state information. See below for more details.	Get

Sensor State Information

Byte	Name	Type	Description
0	Sensor State		Sensor state: 0 - Stopped 1 - Running
1	EtherNet/IP Command in Progress		Command busy status: 0 - Not busy 1 - Busy performing the last command Bytes 2 and 19-83 below are only valid when there is no command in progress.
2	Alignment State		Alignment status: 0 - Not aligned 1 - Aligned

Byte	Name	Type	Description
			The value is only valid when byte1 is set to 0.
3-10	Encoder	64s	Current encoder position
11-18	Time	64s	Current timestamp
19	Current Job Filename Length	8u	Number of characters in the current job filename. (e.g., 11 for "current.job"). The length includes the .job extension. Valid when byte 1 = 0.
20-83	Current Job Filename		Name of currently loaded job, including the ".job" extension. Each byte contains a single character. Valid when byte 1 = 0.
84-87	Runtime Variable 0	32s	Runtime variable value at index 0
...	...		
96-99	Runtime Variable 3	32s	Runtime variable value at index 3

Sample State Assembly

The sample state object contains measurements and their associated stamp information.

Sample State Assembly

Information	Value
Class	0x04
Instance	0x321
Attribute Number	3
Length	380 bytes
Supported Service	0x0E (GetAttributeSingle)

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array	See below	Sample state information. See below for more details.	Get

Sample State Information

Byte	Name	Type	Description
0-1	Inputs	16u	Digital input state of the last frame.
2-9	Z Index Position	64u	Encoder position at time of last index pulse of the last frame.
14-17	Temperature	32u	Sensor temperature in degrees Celsius * 100 (centidegrees) of the last frame.
18-25	Encoder Position	64u	Encoder position of the last frame when the image data was scanned/taken.

Byte	Name	Type	Description
26-33	Time	64u	Time stamp in microseconds of the last frame.
34-41	Frame Counter	64u	The frame number of the last frame.
42	Buffer Count	8u	Represents the number of frames waiting to be output if buffering is enabled.
43	Buffer Overflowing	8u	Indicates whether the output buffer has overflowed: 0 - No overflow 1 - Overflow
44 - 79	Reserved		Reserved bytes.
80-83	Measurement 0	32s	Measurement value in µm (0x80000000 if invalid).
84	Decision 0	8u	Measurement decision. A bit mask, where: Bit 0: 1 - Pass 0 - Fail Bits 1-7: 0 - Measurement value OK 1 - Invalid value 2 - Invalid anchor
...	...		
375-378	Measurement 59	32s	Measurement value in µm (0x80000000 if invalid).
379	Decision 59	8u	Measurement decision. A bit mask, where: Bit 0: 1 - Pass 0 - Fail Bits 1-7: 0 - Measurement value OK 1 = Invalid value 2 = Invalid anchor

Measurement results are reported in pairs of values and decisions. Measurement values are 32 bits wide and decisions are 8 bits wide.

The measurement ID defines the byte position of each pair within the state information. The position of the first word can be calculated as (80 + 5 * ID). For example, a measurement with ID set to 4 can be read from byte 100 (high word) to 103 (low word) and the decision at 104.

In Surface mode, the measurement results are updated after each discrete part has been processed. If buffering is enabled in the Ethernet Output panel, reading the Extended Sample State Assembly Object automatically advances the buffer. See *Ethernet Output* on page 512 for information on the **Output** panel.

Implicit Messaging

Implicit messaging uses UDP and is faster than explicit messaging, and is ideal for time-critical applications. However, implicit messaging is layered on top of UDP. UDP is connectionless and data delivery is not guaranteed. For this reason, implicit messaging is only suitable for applications where occasional data loss is acceptable.

For detailed information on setting up implicit messaging using Allen-Bradley PLCs, see http://lmi3d.com/sites/default/files/APPNOTE_Implicit_Messaging_with_Allen-Bradley_PLCs.pdf.

The following sections describe the implicit messaging assemblies.

Assembly Object (Class 0x04)

For implicit messaging, the Ethernet/IP object model includes the following assemblies: implicit messaging command and implicit messaging output.

All assembly object instances are static. Data in a data byte array in an assembly object are stored in the big endian format.

Implicit Messaging Command Assembly

Implicit Messaging Command Assembly

Information	Value
Class	0x04
Instance	0x64
Attribute Number	3
Length	32 bytes

Implicit Messaging Command Assembly Information

Byte	Name	Type	Description
0	Command	8u	A bit mask where setting the following bits will only perform the action with highest priority*: 1 – Stop sensor 2 – Start sensor 4 – Perform stationary alignment 8 – Perform moving alignment 16 – Clear alignment 32 – Set runtime variables 64 – Load job file 128 – Software trigger

Byte	Name	Type	Description
			*The priority of commands is currently as follows: 1. Stop sensor 2. Start sensor 3. Perform stationary alignment 4. Perform moving alignment 5. Clear alignment 6. Set runtime variables 7. Load job file 8. Software trigger
1-31	Reserved (except for configuring runtime variables and loading job file)		If you are setting the runtime variables, use bytes 4-19 to define the values of each of the four runtime variables in little endian format. If you are loading job file, use bytes 1-31 for the filename, one character per byte. The job name and extension are case-sensitive. The filename must be null terminated and must end with ".job".

Implicit Messaging Output Assembly

Implicit Messaging Output Assembly

Information	Value
Class	0x04
Instance	0x322
Attribute Number	3
Length	376 bytes

Implicit Messaging Output Assembly Information

Byte	Name	Type	Description
0	Sensor State	8u	Sensor state is a bit mask where: Bit 0: 1 – Running 0 – Stopped Bit 1: 1 – Conflict due to unreachable buddy 0 – No conflict Bits [2-7]: Not used.
1	Alignment and Command state	8u	A bit mask where: Bit 0:

Byte	Name	Type	Description
			1 – Explicit or Implicit Command in progress 0 – No Explicit or Implicit command is in progress Bit 1 1 – Aligned 0 – Not aligned
2-3	Inputs	16u	Digital input state of the last frame.
4-11	Z Index Position	64u	Encoder position at time of last index pulse of the last frame.
12-15	Exposure	32u	Exposure in µs of the last frame.
16-19	Temperature	32u	Sensor temperature in degrees celsius * 100 (centidegrees) of the last frame.
20-27	Encoder Position	64s	Encoder position of the last frame when the image data was scanned/taken.
28-35	Time	64u	Time stamp in microseconds of the last frame.
36-43	Frame Index	64u	The frame number of the last frame.
44-55	Reserved		
56	Decision 0	8u	Measurement decision is a bit mask where: Bit 0: 1 – Pass 0 – Fail Bits [1-7]: 0 – Measurement value OK 1 – Invalid Value 2 – Invalid Anchor
...	...		
119	Decision 63	8u	Measurement decision is a bit mask where: Bit 0: 1 – Pass 0 – Fail Bits [1-7]: 0 – Measurement value OK 1 – Invalid Value 2 – Invalid Anchor
120-123	Measurement 0	32s	Measurement value in µm. (0x80000000 if invalid)

Byte	Name	Type	Description
...	...		
372-375	Measurement 63	32s	Measurement value in µm. (0x80000000 if invalid)

Rockwell Allen-Bradley Instructions

This section describes how to set up network communications over the EtherNet/IP industrial communication protocol with Allen-Bradley PLCs that are EtherNet/IP-capable. Gocator supports two EtherNet/IP messaging methods: implicit messaging via UDP and explicit messaging via TCP.

Implicit messaging has advantages and disadvantages. Implicit messaging uses UDP and is faster than explicit messaging and is ideal for time-critical applications. Since implicit messaging is layered on top of UDP, it is connectionless and data delivery is not guaranteed. For this reason, implicit messaging is only suitable for applications where occasional data loss is acceptable. Two connection types are available for implicit communication: a Monitor Data connection or a Monitor Data and Control Data connection.

Explicit messaging is more suitable for deterministic and verified communication transfer where no losses are acceptable. Add-On Profile (AOP) is not available for the Gocator, and it is not possible to use the EDS file for automatic configuration.

For these reasons, LMI recommends in most application using a closed ethernet subnet (i.e., network switch, PLC, Gocator(s), and setup PC only) to minimize losses and collisions and cyclical implicit messaging over the EtherNet/IP protocol unless a specific control command such as job loading and/or transfer verification is required.

Software and Hardware Setup

The following software and hardware were used during development.

Requirements	Details
Gocator Firmware	5.2 and higher
Gocator Series	G1, G2, and G3 sensors.
Required Files	GocatorEip.eds
	LMI.ico
	Gocator_EthernetIP.ACD
Other	Allen-Bradley L16ER-BB1B PLC
	Allen-Bradley Studio 5000 programming tool V21.11 or newer
	D-Link Unmanaged Industrial Gigabit Ethernet Switch DGS-108

Note: The Ethernet card to which the Gocator is connected should be added as a module to the Backplane. Verify that the IP Address is on the correct subnet. Note the IP address should be that of the PLC's Ethernet modules, not that of the Gocator's.

Byte Order Options

Gocator supports outputting in either Big Endian or Little Endian byte ordering options.

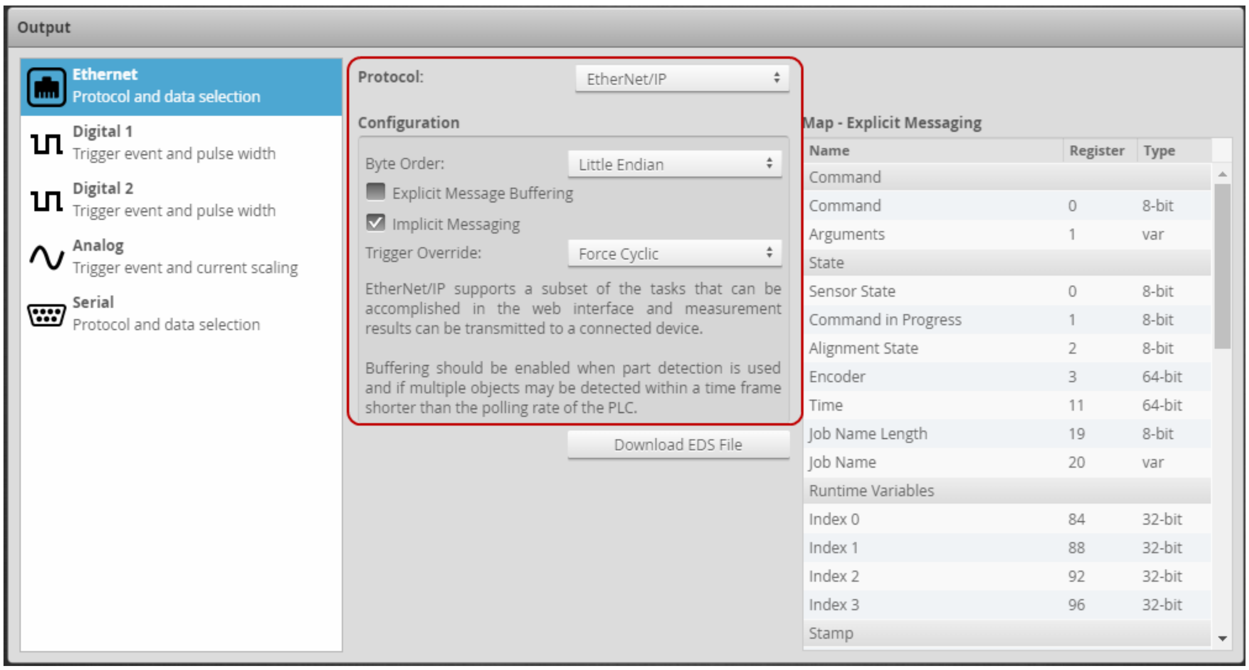
Big Endian Byte Order: The most significant byte (the "big end") of the data is placed at the byte with the lowest address. The rest of the data is placed in order of decreasing significance in the next three bytes of memory.

Little Endian Byte Order: The least significant byte (the "little end") of the data is placed at the byte with the lowest address. The rest of the data is placed in order of increasing significance in the next three bytes in memory.

Most Allen-Bradley PLCs default to Little Endian addressing formats, but you should verify this when configuring the PLC.

Setting Up Implicit Messaging on the Gocator

To output in EtherNet/IP implicit messaging mode on the sensor, you configure the sensor using the **Protocol** setting and the **Configuration** area on the **Output** page. Note that the *type* of implicit messaging (cyclic versus change of state) is determined by the **Trigger Override** setting.



To configure the sensor for EtherNet/IP implicit messaging mode:

1. On the **Output** page, in the **Ethernet** category, choose **EtherNet/IP** as the protocol.
2. Choose **Little Endian** from the **Byte Order** dropdown box.
3. Make sure that **Explicit Message Buffering** is unchecked.
4. Check the **Implicit Messaging** option.
5. Set the **Trigger Override** dropdown to the type of implicit messaging you are using.

   For cyclic messaging, set **Trigger Override** to **Force Cyclic**.

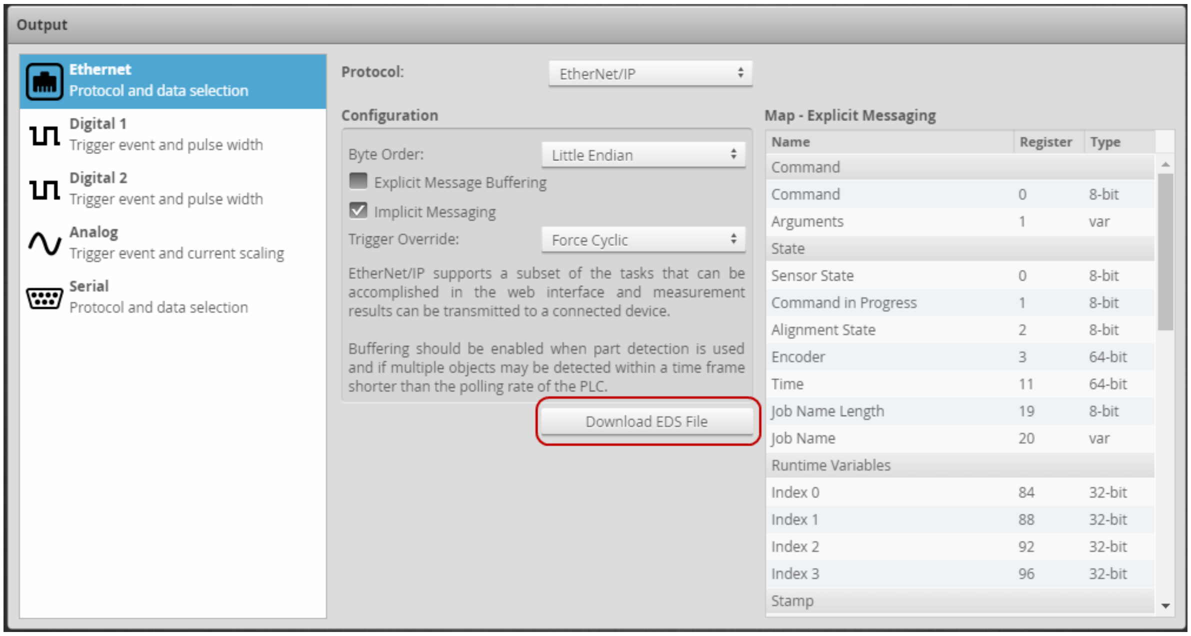
   For change of state messaging, set **Trigger Override** to **Force Change of State**.

   When you set up the PLC to communicate with a Gocator using change of state implicit messaging, an event task must be created on the PLC to rapidly check whether the sensor is running; if the frame count increases, data is copied to an array. The event task period must allow the event task to be executed at a higher rate than Gocator frame rate. For more information, see *Setting Up Implicit Messaging on the PLC* on the next page.

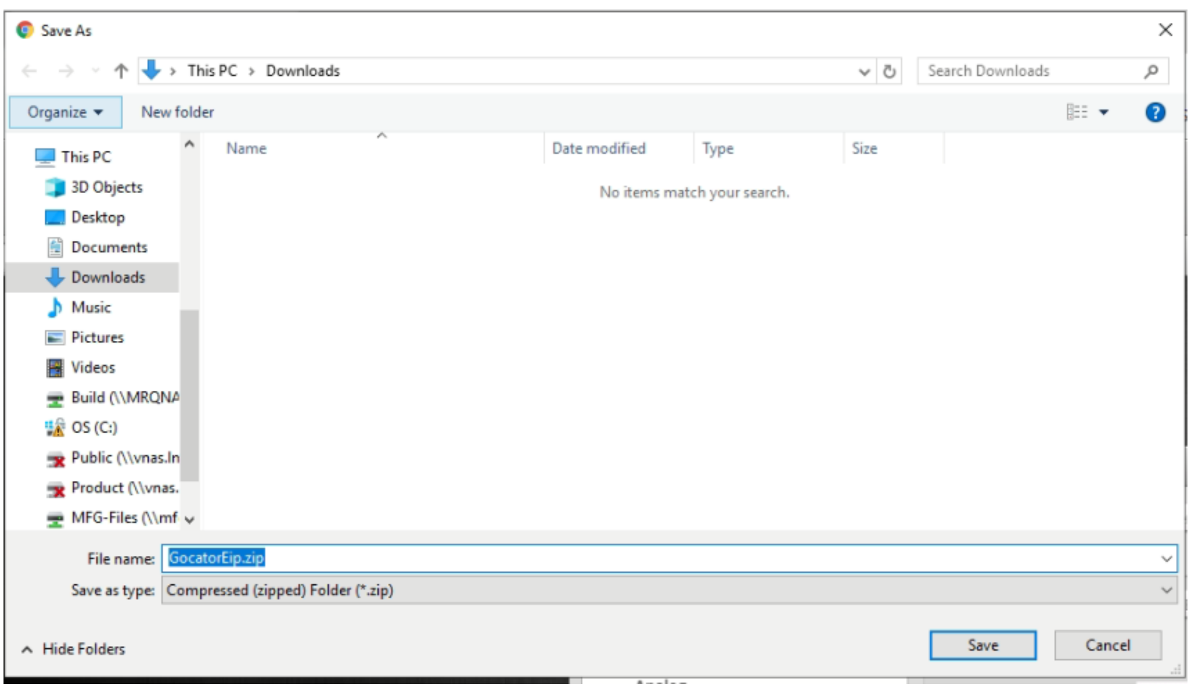
Before setting up implicit messaging on the PLC, you must download the EDS file from the Gocator sensor to the PC.

To download the EDS file:

1. Click Download EDS File to download the latest Gocator EDS file to the PC connected to the sensor.

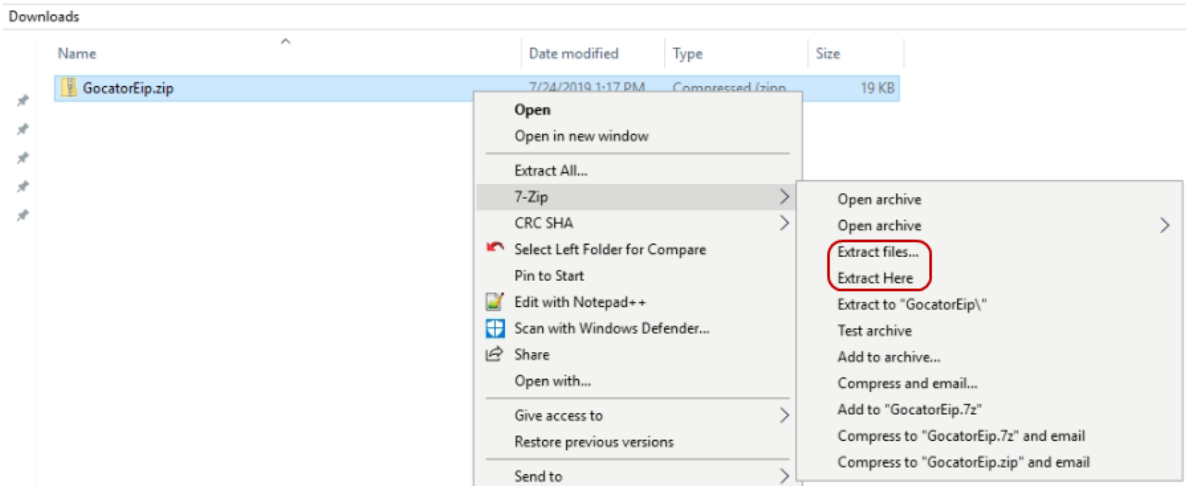


2. Click **Save** to save the zipped folder to a convenient location.



3. Extract the zipped folder.

   Your unzipped folder will contain two files: a .eds file and a .ico file.



4. Make note of where you have extracted the EDS file.

Setting Up Implicit Messaging on the PLC

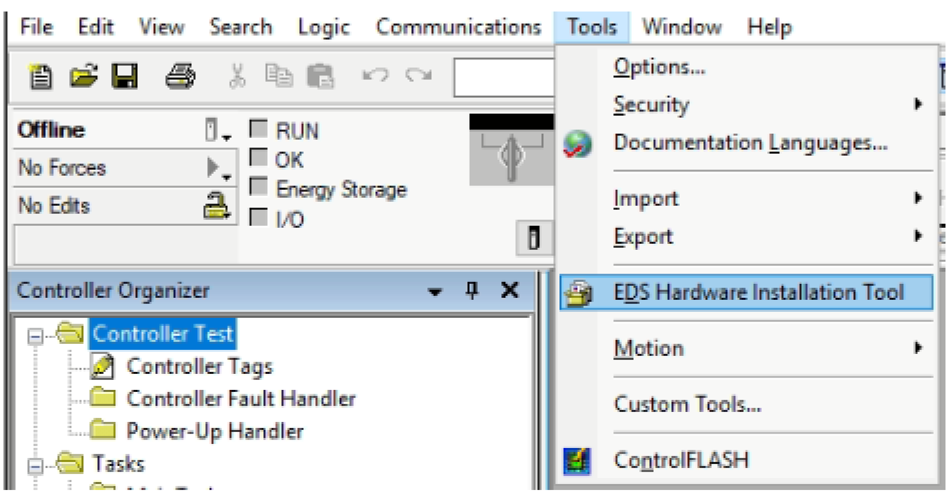
This section describes setting up implicit messaging on the PLC.

Install EDS File

1. If you haven't already done so, download the EDS file from the Gocator sensor.

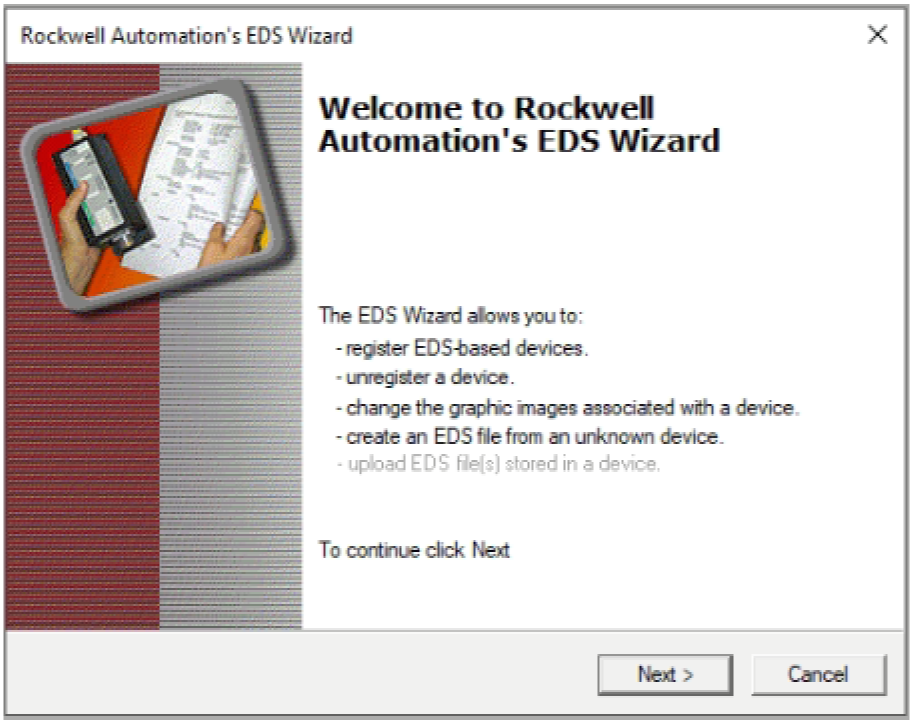
   For more information, see *To download the EDS file:* on the previous page.

2. In Studio 5000, under the **Tools** menu, click **EDS Hardware Installation Tool**.

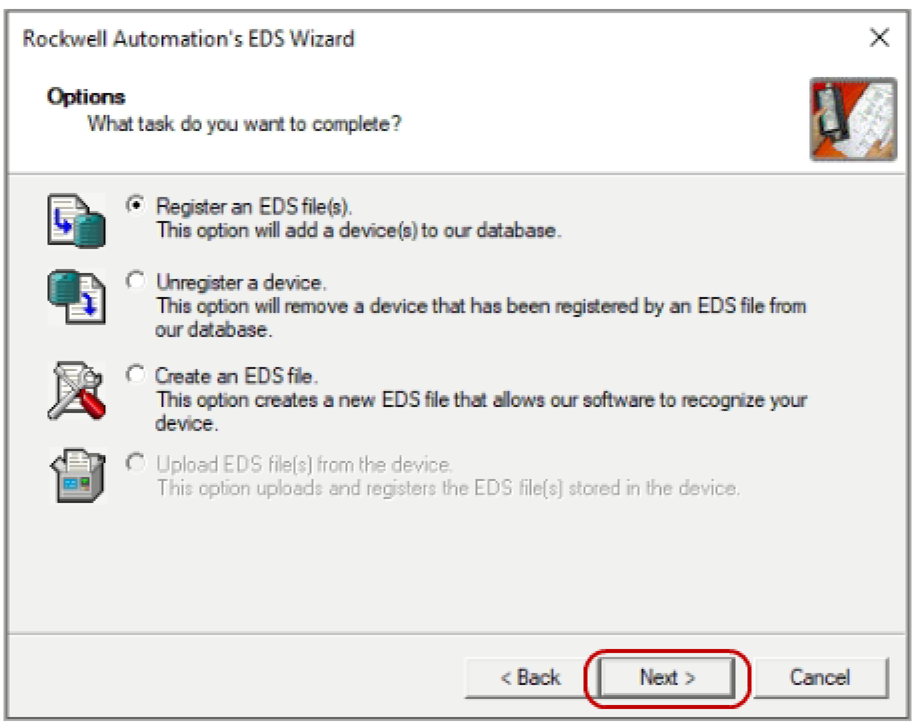


   The EDS setup tool (the Rockwell Automation EDS Wizard) launches.

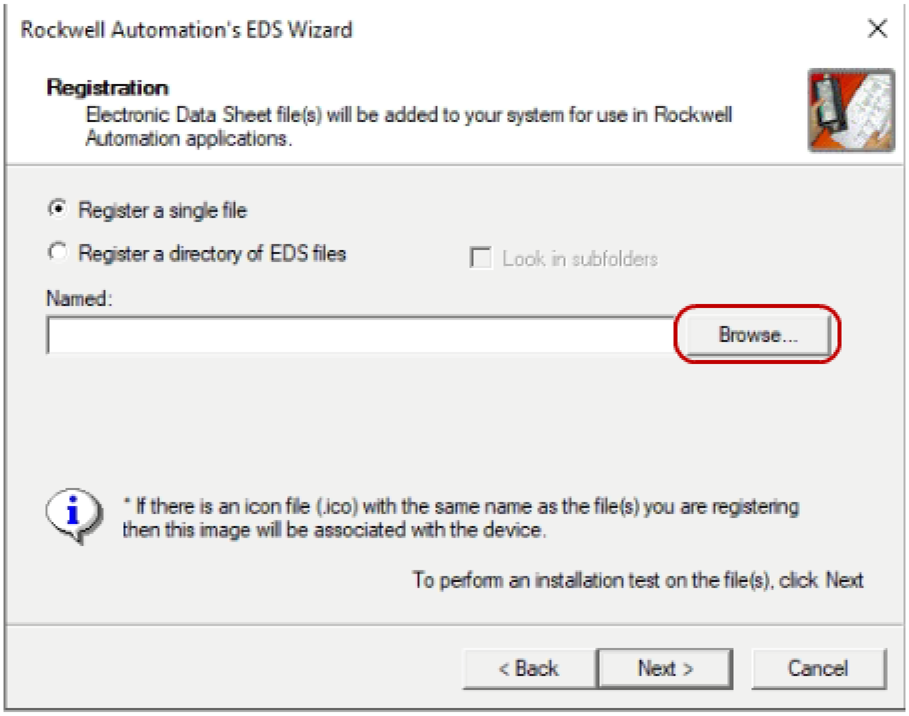
3. In the wizard, click **Next**.



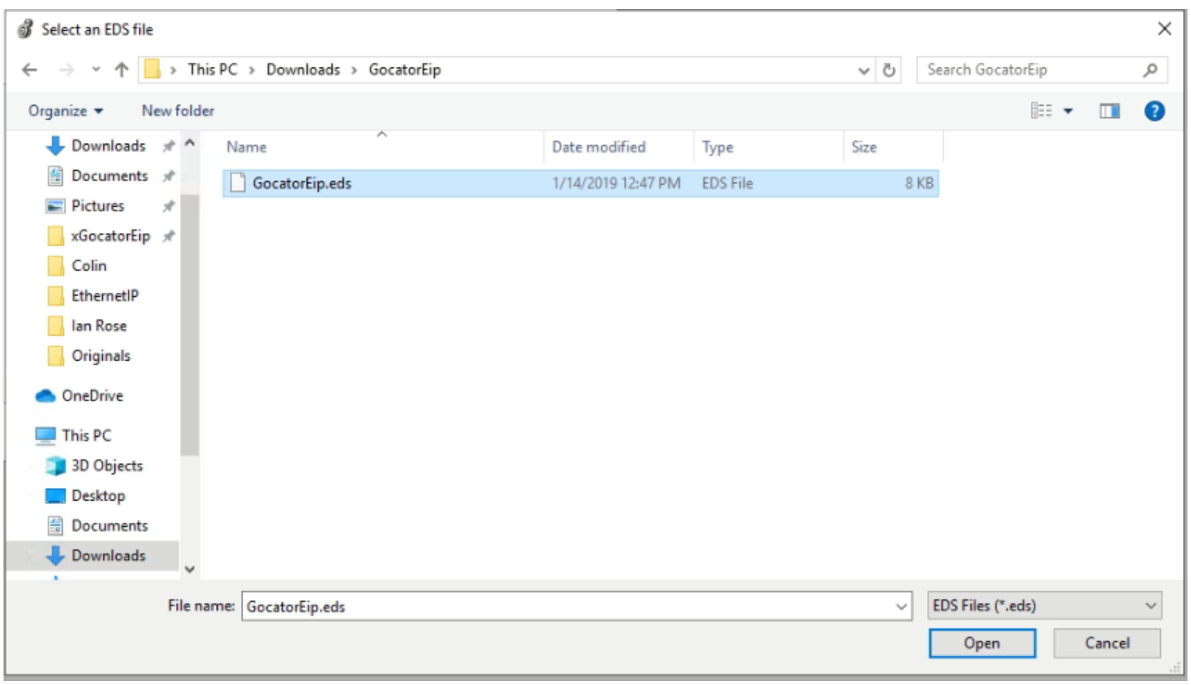
4. Choose **Register an EDS file(s)** and click **Next**



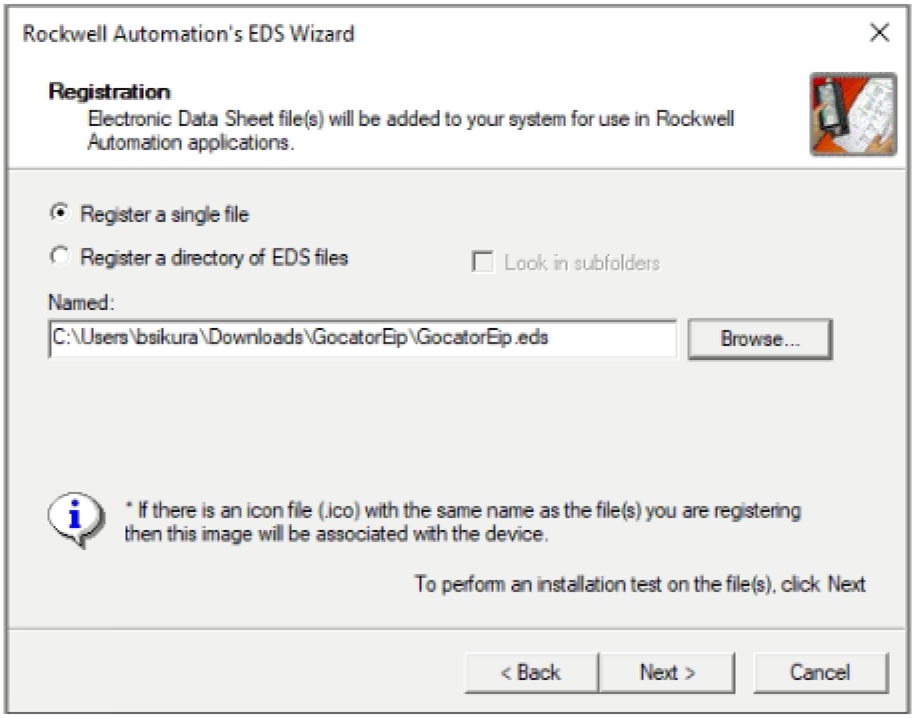
5. Choose **Register a single file** and then click **Browse**.



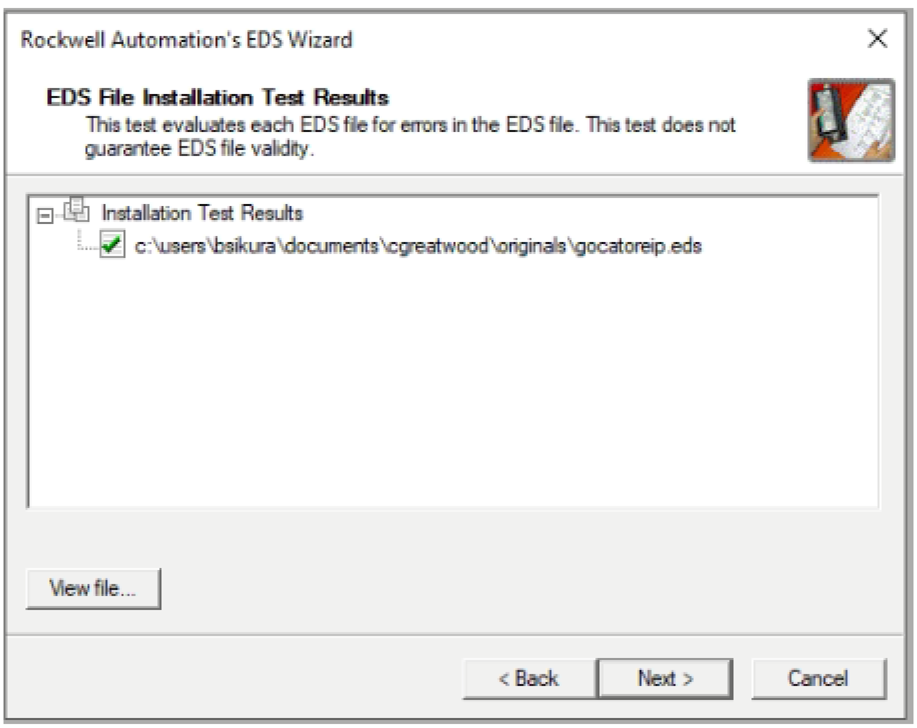
6. Navigate to the unzipped .eds file you downloaded and unzipped, select it, and click **Open**.



7. Click **Next**.



8. If your EDS file has no errors or conflicts (a green checkmark is displayed next to the .eds file), click **Next**.

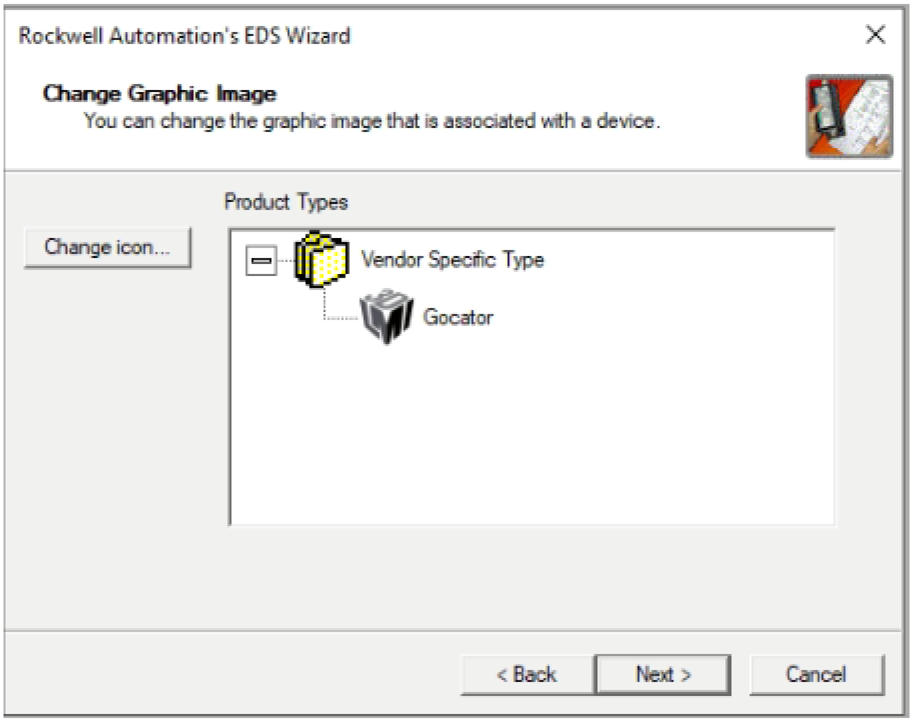


9. Verify that the tool automatically selects the LMI logo from the unzipped folder.

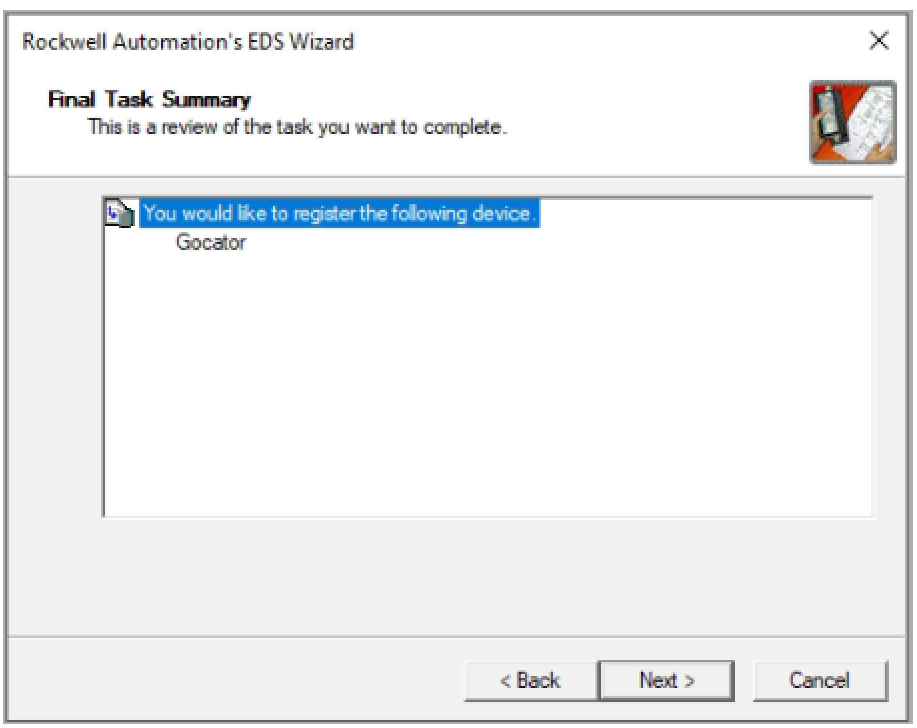
   The .ico file contained in the zip folder you downloaded previously contains the logo.

10. If the tool does not automatically select the LMI logo, navigate to the file, select it, and then click **Next**.

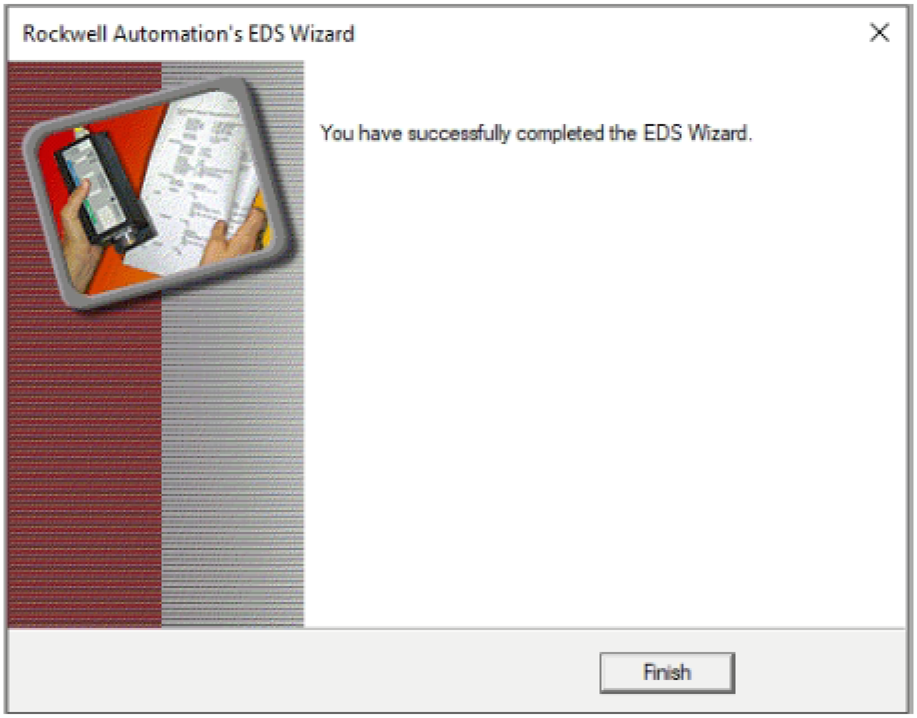
Proper icon selection is important, as this will make it easier for maintenance/future engineers to identify the sensor product from a long list of connected devices in a PLC program.



11. Click **Next**.

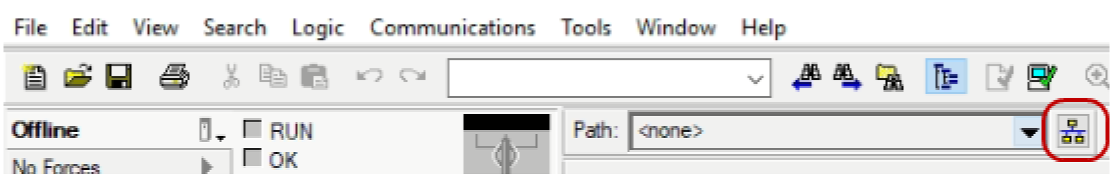


12. Click **Finish**.

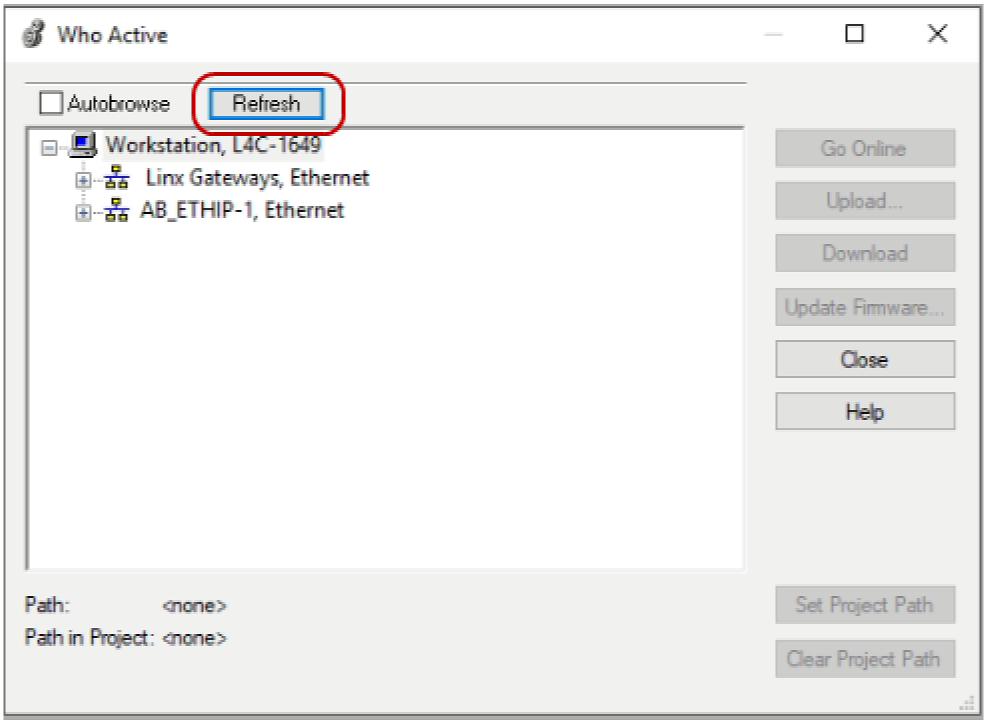


Add Gocator IO Device to PLC Program

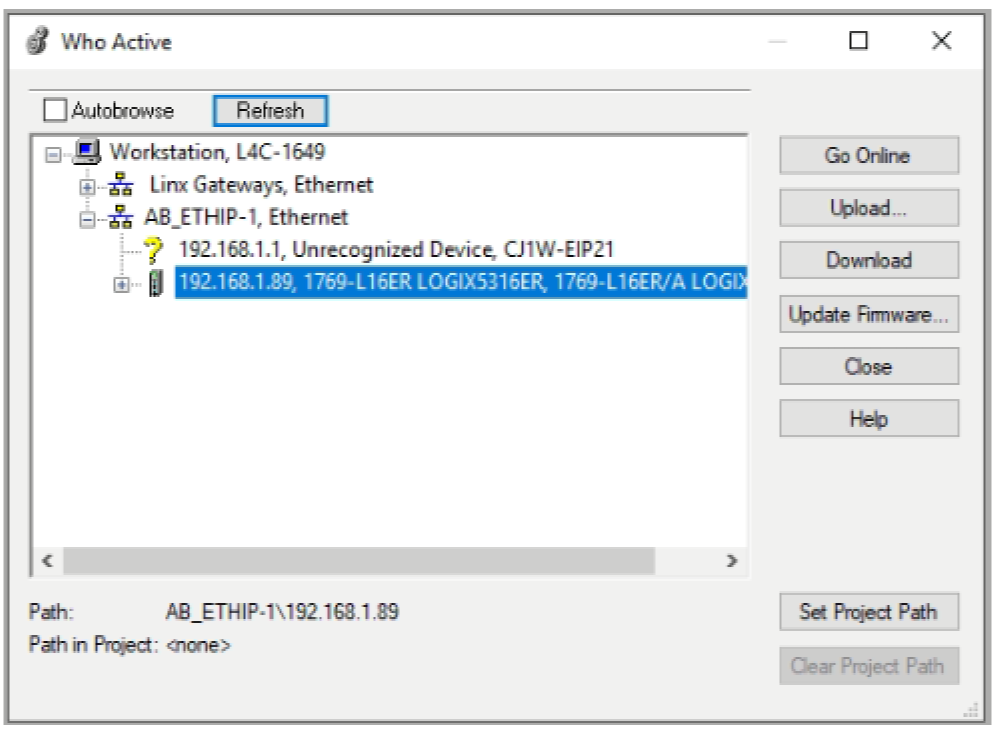
1. Click the **Who Active** button to the right of the Path field.



2. Click **Refresh** in RSLinx the **Who Active** dialog to update your available devices.



3. In the tree structure, navigate to and select your PLC controller ethernet node.

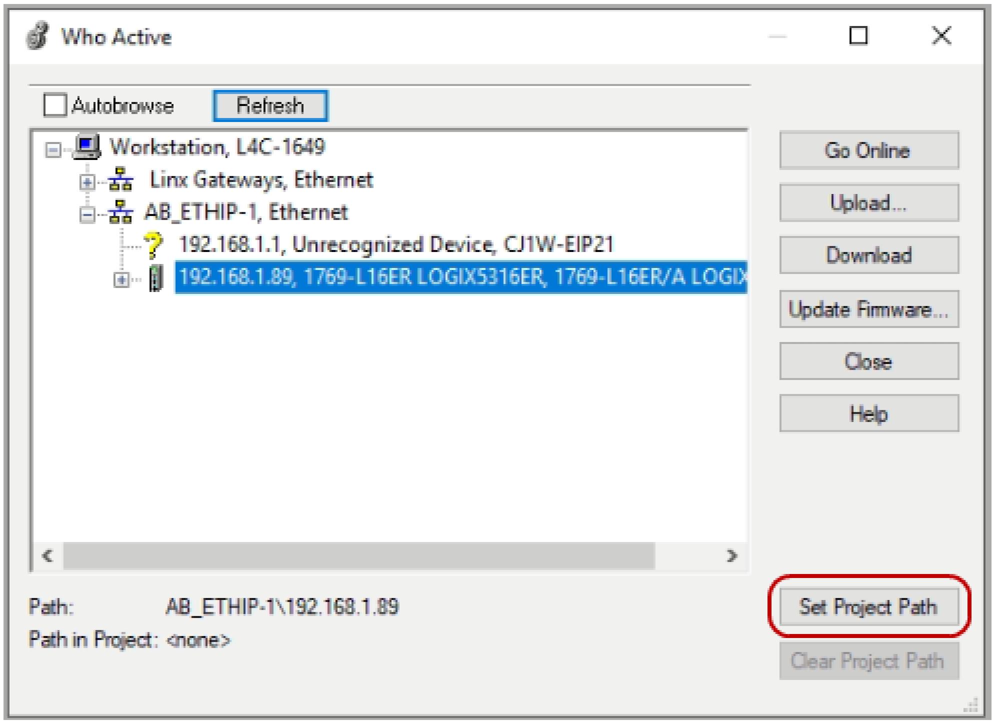


4. If you do not see the **Go Online** option at this point, make sure that RSLinx has been started and is running in the background on your setup PC.

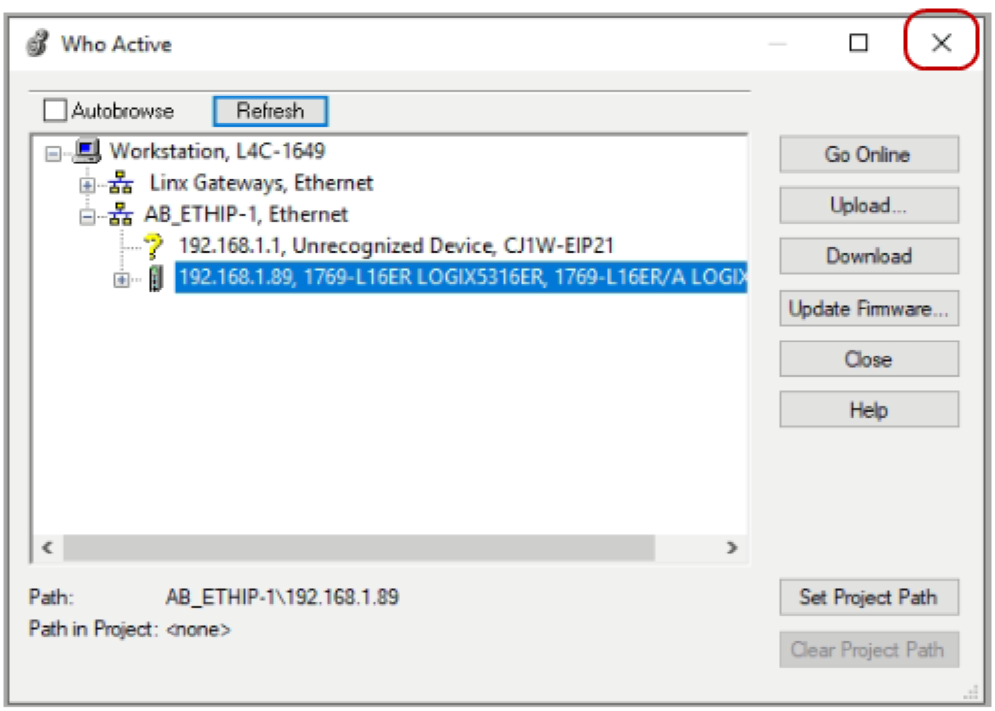
   Device discovery will not complete if RSLinx is not running.

5. Click **Set Project Path**.

This will set your project path when you attempt to download to the PLC later.

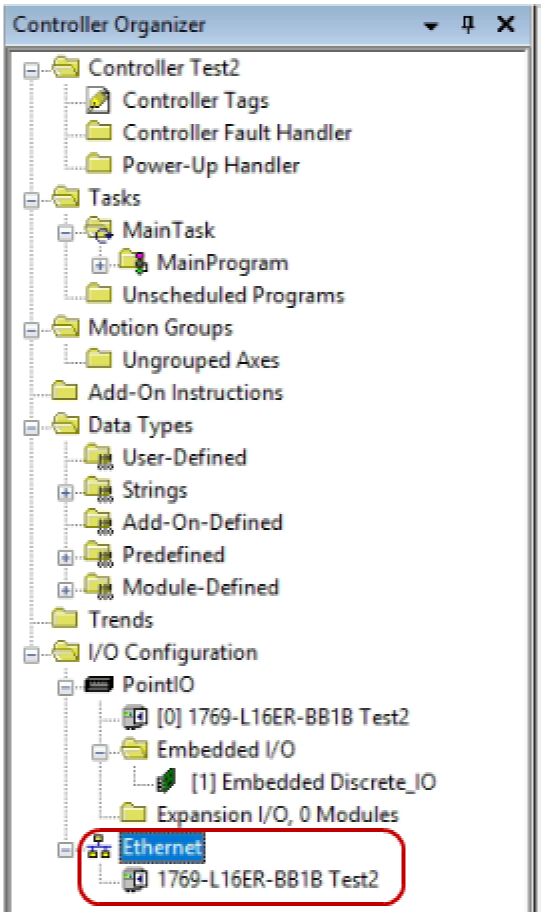


6. Click X to exit your node setup.

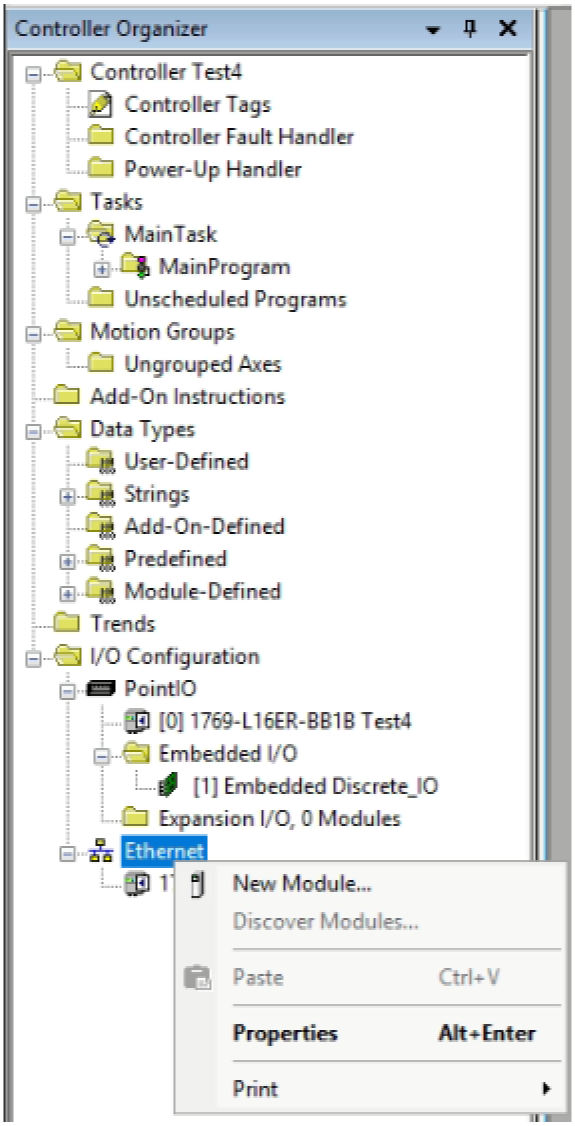


7. In Studio 5000, verify that the path is updated to the IP address of your controller.

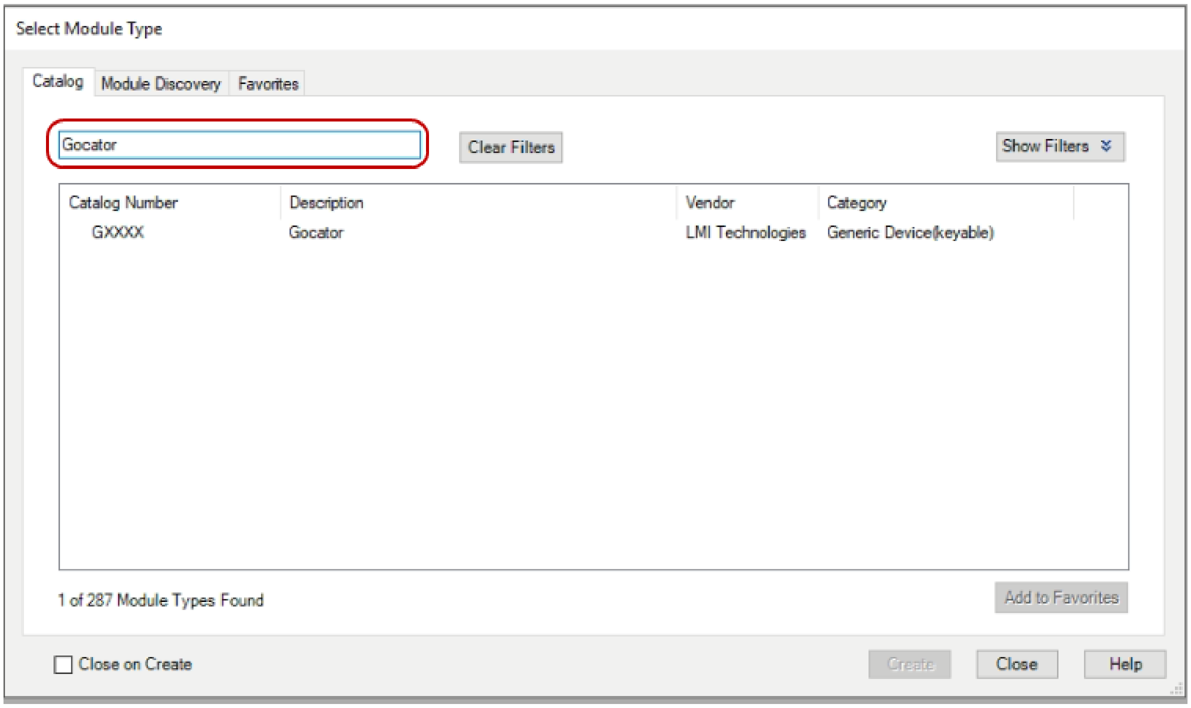
8. In the Controller Organizer, choose **Ethernet** under the **IO Configuration** node.



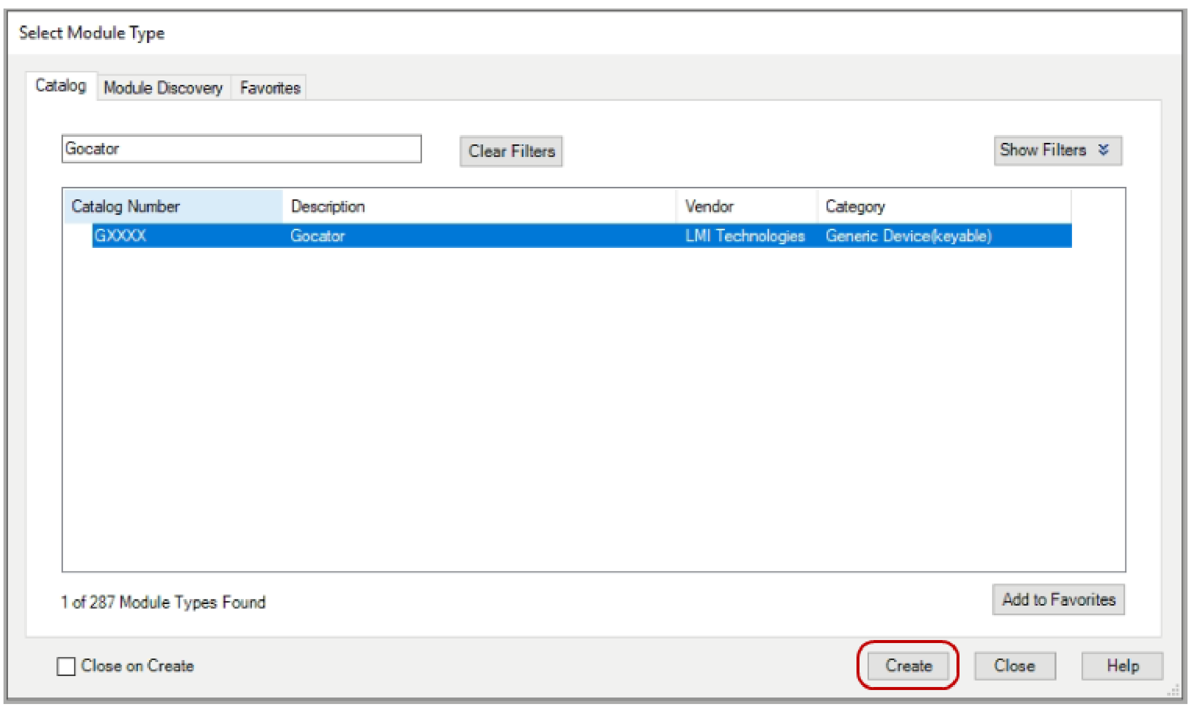
9. Right-click the Ethernet network node and click **New Module**.



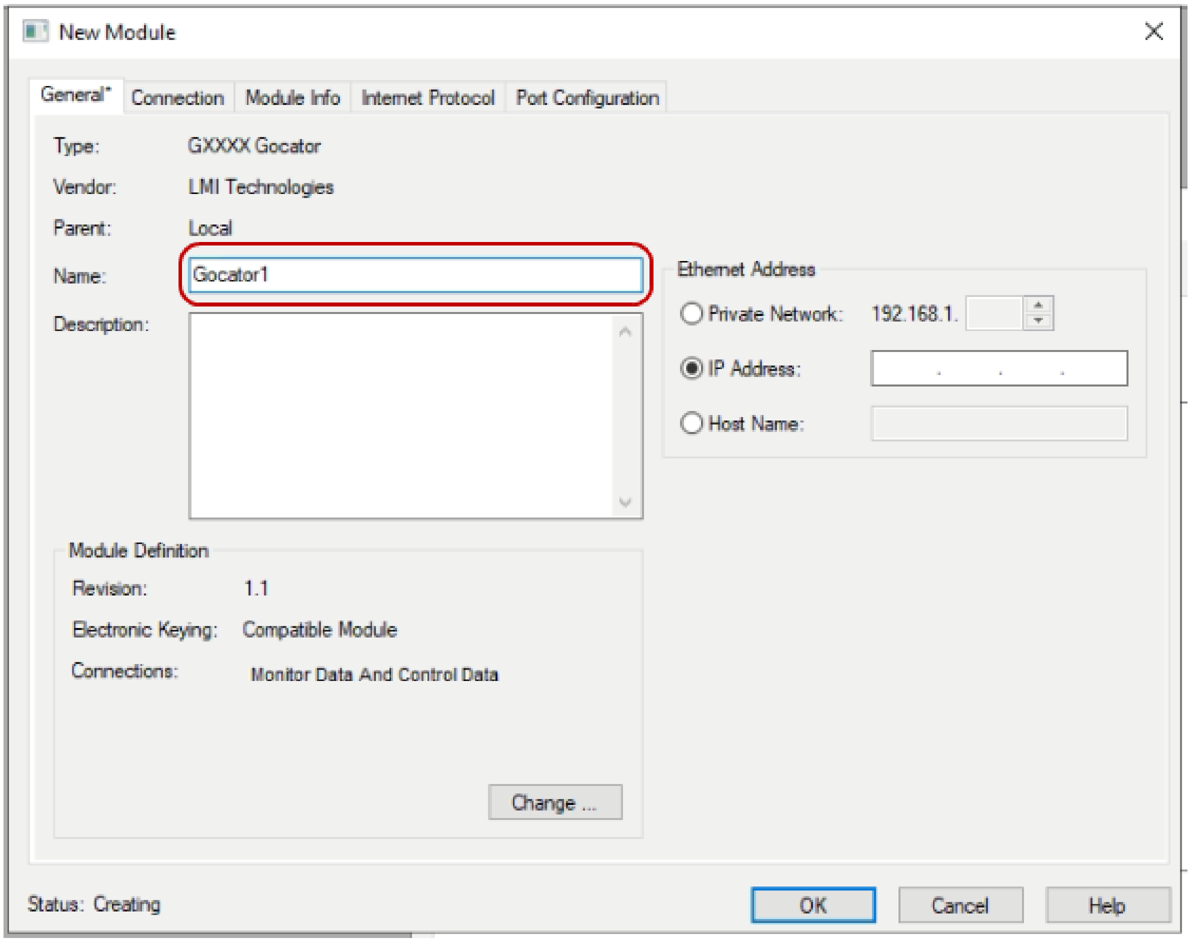
10. Type “Gocator” into the search bar of the dialog that appears.



11. In the list under the search bar, select the new Gocator device file and click **Create**.

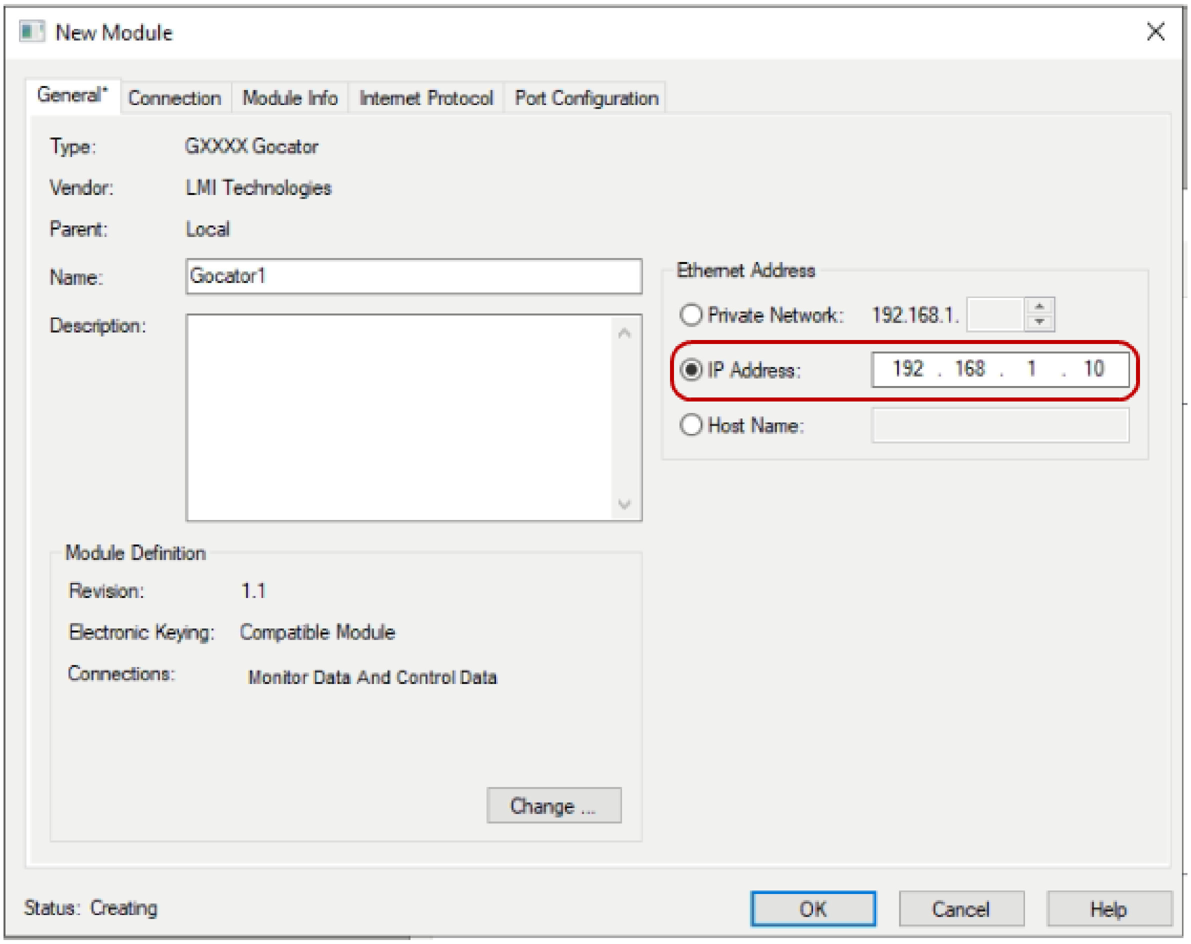


12. In the New Module dialog, in the **Name** field, give the new IO device a *unique* name.



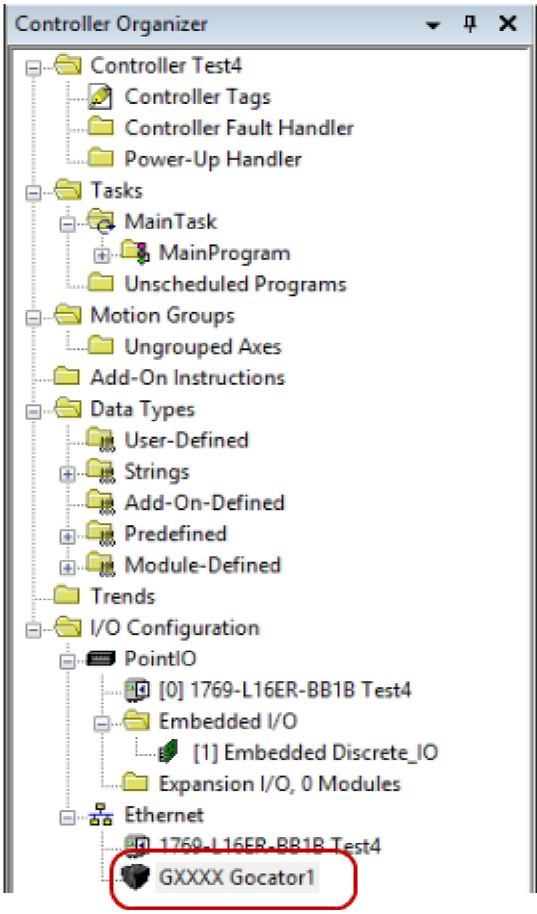
13. Type in the static IP address of the first sensor that you are trying to set up, and then click **OK**.

    The default IP address for all Gocator sensors from the factory is 192.168.1.10. You can verify the IP address of the sensor by logging into the web user interface in a browser or by using the kDiscovery utility available in the GoUtilities package available from LMI's Download Center.

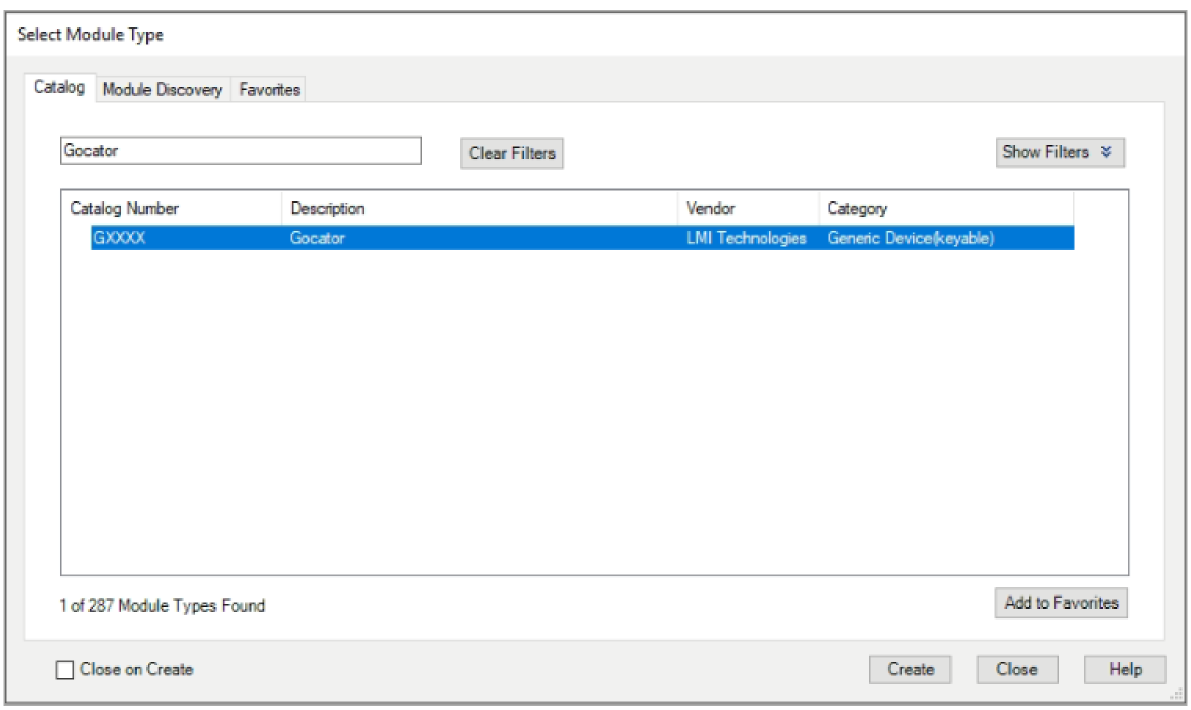


14. In your Program tree, verify that you now have a new IO device.

    The naming format shown should be device devicename (i.e., GXXX Gocator1)

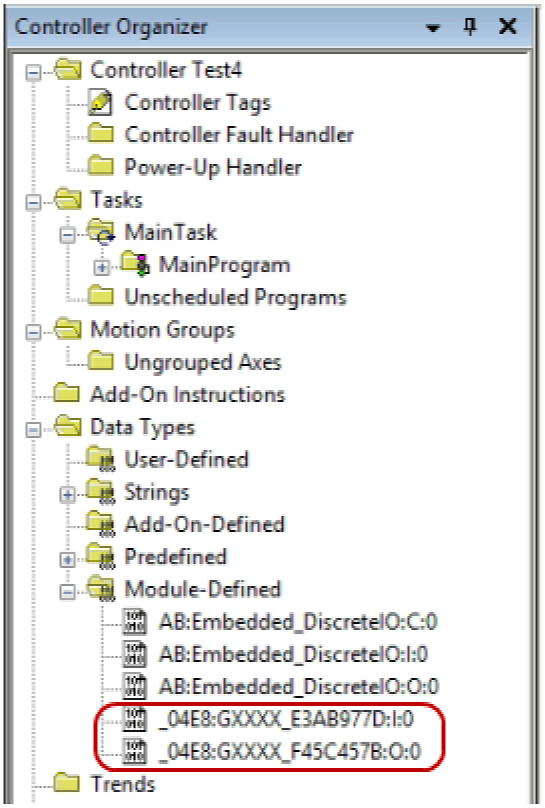


15. In the Select Module Type dialog, click **Close**.



16. In the Controller Organizer, under **Data-Types**, expand **Module-Defined** and verify that you have two new data blocks.

    These will correspond to the Input and Output data coming from and going to the Gocator, respectively, for a *Monitor Data and Control Data* connection type.

    When the Gocator is in Implicit Messaging mode, data will be streamed and stored in the Gocator1:I tag when both the PLC is in Run mode and the Gocator is started. The tag address header is formatted as devicename:I and/or devicename:O for inputs and outputs, respectively.

    For the data format, see *Implicit Messaging Output Assembly* on page 729.

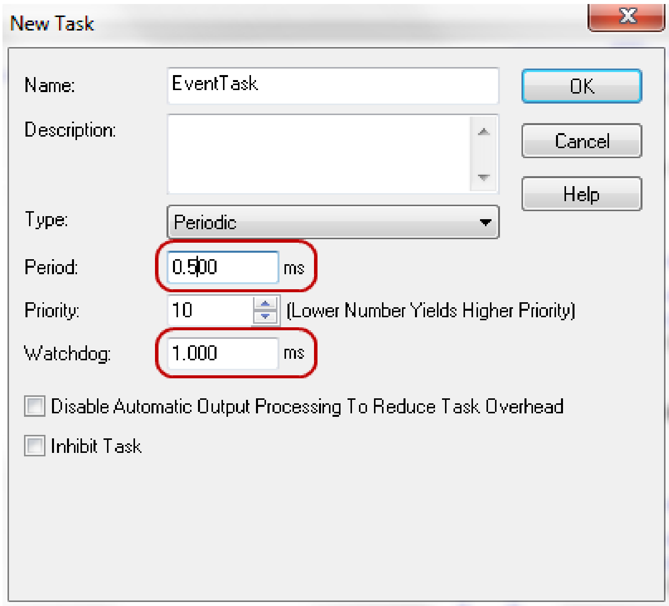
    The EDS file now contains detailed tag descriptions as shown below that can be used directly in the PLC program.

Name	Data Type
ConnectionFaulted	BOOL
Sensor_State	SINT
Run_State	BOOL
State_Issue1	BOOL
State_Issue2	BOOL
State_Issue3	BOOL
State_Issue4	BOOL
State_Issue5	BOOL
State_Issue6	BOOL
State_Issue7	BOOL
Alignment_and_Command_State	SINT
Command_in_Progress	BOOL
Aligned	BOOL
Inputs	INT
Z_Index_Position_0	DINT
Z_Index_Position_1	DINT
Exposure	DINT
Temperature	DINT
Encoder_Position_0	DINT
Encoder_Position_1	DINT
Time_0	DINT
Time_1	DINT
Frame_0	DINT
Frame_1	DINT
Decision0	SINT
Decision1	SINT
Decision2	SINT
Decision3	SINT
Decision4	SINT
Decision5	SINT
Decision6	SINT
Decision7	SINT
Decision8	SINT
Decision9	SINT
Decision10	SINT
Decision11	SINT
Decision12	SINT
Decision13	SINT
Decision14	SINT
Decision15	SINT
Decision16	SINT
Decision17	SINT
Decision18	SINT
Decision19	SINT
Decision20	SINT
Decision21	SINT
Decision22	SINT
Decision23	SINT
Decision24	SINT
Decision25	SINT
Decision26	SINT
Decision27	SINT
Decision28	SINT
Decision29	SINT
Decision30	SINT
Decision31	SINT
Decision32	SINT
Decision33	SINT
Decision34	SINT
Decision35	SINT
Decision36	SINT
Decision37	SINT
Decision38	SINT
Decision39	SINT
Decision40	SINT
Decision41	SINT
Decision42	SINT
Decision43	SINT
Decision44	SINT
Decision45	SINT
Decision46	SINT
Decision47	SINT
Decision48	SINT
Decision49	SINT
Decision50	SINT
Decision51	SINT
Decision52	SINT
Decision53	SINT
Decision54	SINT
Decision55	SINT
Decision56	SINT
Decision57	SINT
Decision58	SINT
Decision59	SINT
Decision60	SINT
Decision61	SINT
Decision62	SINT
Decision63	SINT
Measurement0	DINT
Measurement1	DINT
Measurement2	DINT
Measurement3	DINT
Measurement4	DINT
Measurement5	DINT
Measurement6	DINT
Measurement7	DINT
Measurement8	DINT
Measurement9	DINT
Measurement10	DINT
Measurement11	DINT
Measurement12	DINT
Measurement13	DINT
Measurement14	DINT
Measurement15	DINT
Measurement16	DINT
Measurement17	DINT
Measurement18	DINT
Measurement19	DINT
Measurement20	DINT
Measurement21	DINT
Measurement22	DINT
Measurement23	DINT
Measurement24	DINT
Measurement25	DINT
Measurement26	DINT
Measurement27	DINT
Measurement28	DINT
Measurement29	DINT
Measurement30	DINT
Measurement31	DINT
Measurement32	DINT
Measurement33	DINT
Measurement34	DINT
Measurement35	DINT
Measurement36	DINT
Measurement37	DINT
Measurement38	DINT
Measurement39	DINT
Measurement40	DINT
Measurement41	DINT
Measurement42	DINT
Measurement43	DINT
Measurement44	DINT
Measurement45	DINT
Measurement46	DINT
Measurement47	DINT
Measurement48	DINT
Measurement49	DINT
Measurement50	DINT
Measurement51	DINT
Measurement52	DINT
Measurement53	DINT
Measurement54	DINT
Measurement55	DINT
Measurement56	DINT
Measurement57	DINT
Measurement58	DINT
Measurement59	DINT
Measurement60	DINT
Measurement61	DINT
Measurement62	DINT
Measurement63	DINT

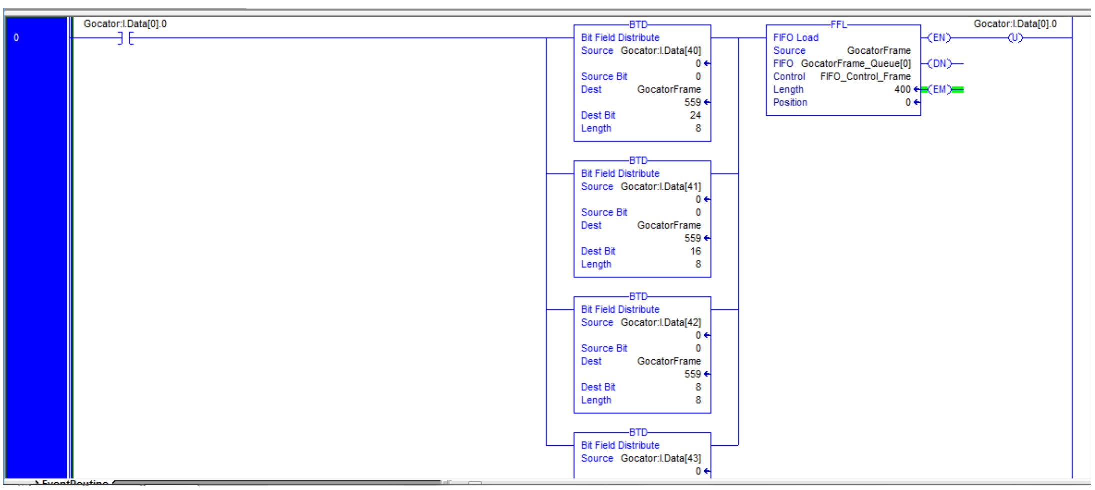
17. If you set the sensor to use change of state earlier (**Trigger Override** is set to **Force Change of State** in the Output panel), perform the following additional steps.

    a. In the RSLogix 5000 programming tool, create a new task with a 0.5 millisecond period and a 1.0 millisecond watchdog, and then click **OK** at the upper right.

       A major fault alarm is triggered if the task does not finish execution within the watchdog time limit.

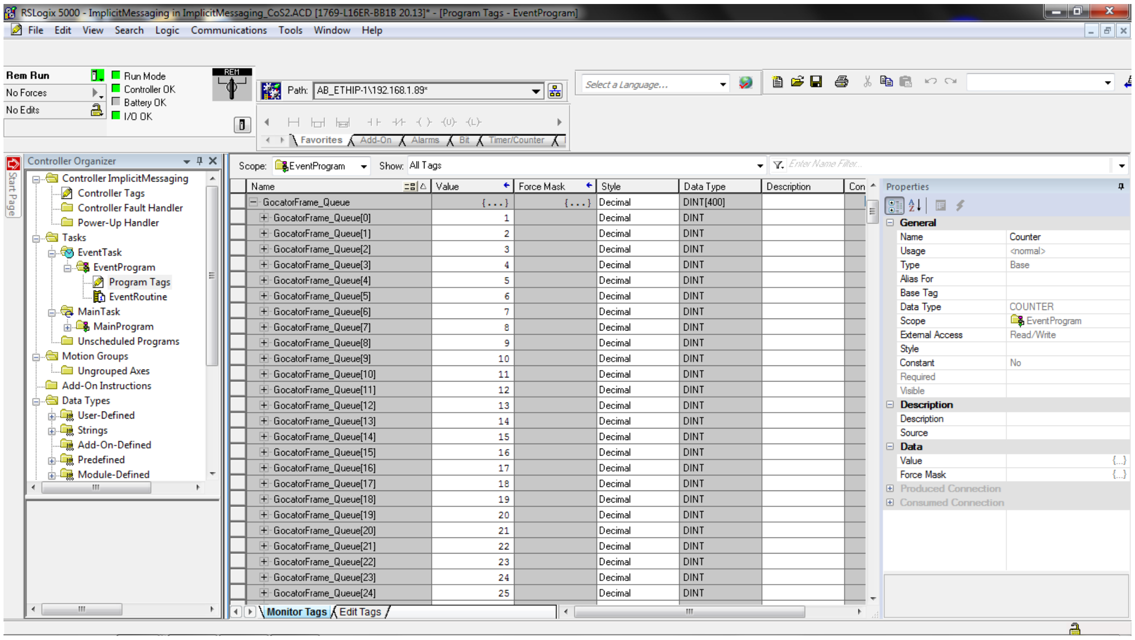


Ladder logic is written to monitor the Gocator's running state and store data into a FIFO (Ladder Element FFL) array of the same data type.



b. Confirm that frames are properly stored in the stored array, without any repetition or dropped frames.

In this case, the Gocator frame count is stored in a user-defined array.



Using the Implicit Messaging Gocator Command Assembly

The Output Message format (from PLC to Gocator) is used to control the sensor through implicit messaging. This message is sent continuously from the PLC to the Gocator at the user-requested Request Packet Interval (RPI) on the PLC side. The default Gocator RPI is 10ms.

In PLC programming, the standard practice is to use bits instead of sending a value representing that command, for example, start/stop bits. When using values, the PLC needs to add more code to convert it to bits and vice versa.

Since the Gocator does not allow parallel commands, a priority scheme is needed to handle multiple command bits being set at the same time. Only the bit with the highest priority will be accepted as the command.

The total command message size is 32 bytes.

For information on the command assembly structure, see *Implicit Messaging Command Assembly* on page 728.

It's important to understand that because the Gocator is driven internally by its own clock, and because users can configure the Gocator for any frame rate—independently of the RPI request configured on the PLC—Cyclic implicit messaging can cause unnecessary data loss if the two clocks are not synchronized. Using Change of State implicit messaging instead can overcome this issue. For instructions on how to set up Change of State implicit messaging, see [Setting Up Change of State Implicit Messaging](#).

The data block used to send control messages to the Gocator should have been set properly up in *Setting Up Implicit Messaging on the PLC* on page 734. It will appear in the Gocator Module-Defined data types as shown below:

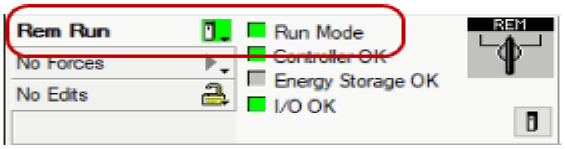
Name	Data Type
Command	SINT
Reserved_or_Job_File1	SINT
Reserved_or_Job_File2	SINT
Reserved_or_Job_File3	SINT
Reserved_or_Job_File4	SINT
Reserved_or_Job_File5	SINT
Reserved_or_Job_File6	SINT
Reserved_or_Job_File7	SINT
Reserved_or_Job_File8	SINT
Reserved_or_Job_File9	SINT
Reserved_or_Job_File10	SINT
Reserved_or_Job_File11	SINT
Reserved_or_Job_File12	SINT
Reserved_or_Job_File13	SINT
Reserved_or_Job_File14	SINT
Reserved_or_Job_File15	SINT
Reserved_or_Job_File16	SINT
Reserved_or_Job_File17	SINT
Reserved_or_Job_File18	SINT
Reserved_or_Job_File19	SINT
Reserved_or_Job_File20	SINT
Reserved_or_Job_File21	SINT
Reserved_or_Job_File22	SINT
Reserved_or_Job_File23	SINT
Reserved_or_Job_File24	SINT
Reserved_or_Job_File25	SINT
Reserved_or_Job_File26	SINT
Reserved_or_Job_File27	SINT
Reserved_or_Job_File28	SINT
Reserved_or_Job_File29	SINT
Reserved_or_Job_File30	SINT
Reserved_or_Job_File31	SINT

Starting a Sensor

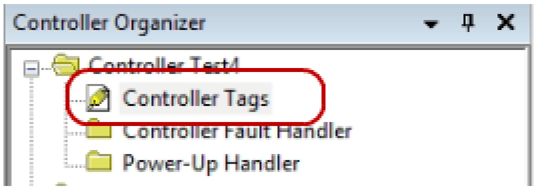
To start a sensor, do the following:

1. Make sure that you have downloaded your PLC program to the controller and that your controller is in Run mode.

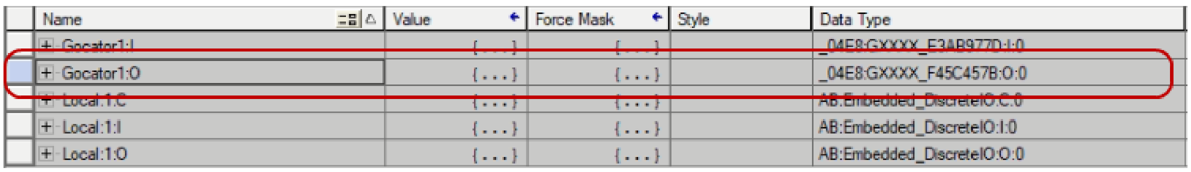
   For information on downloading the the PLC program to the controller, see *Install EDS File* on page 734.



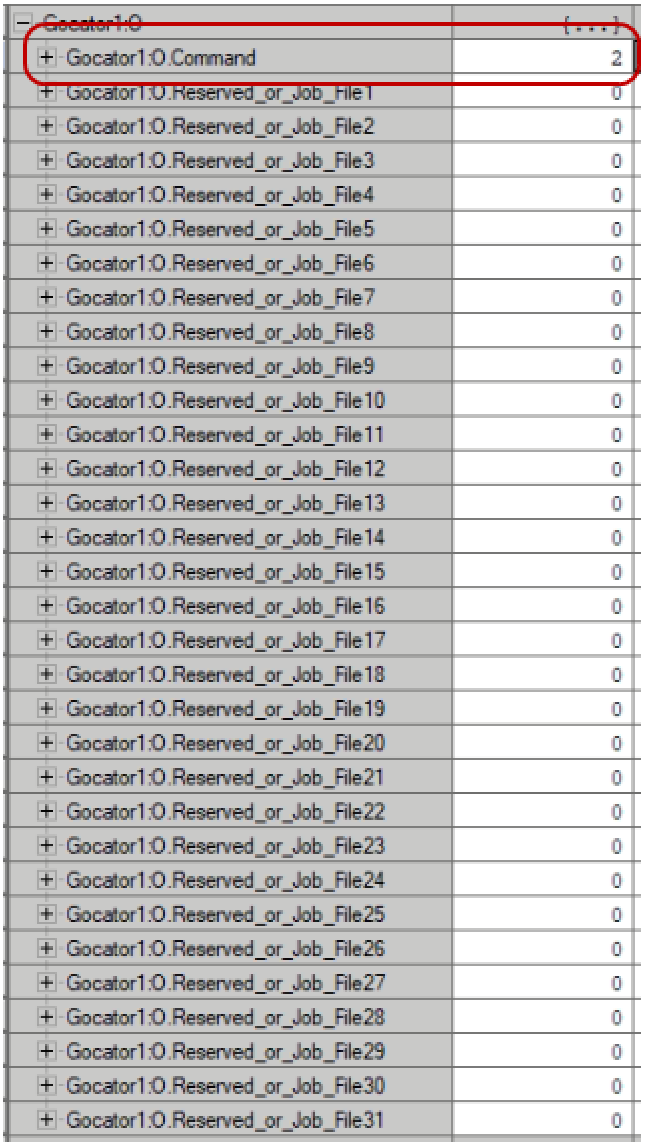
2. In the Controller Organizer, double-click **Controller Tags** to show them in the main screen



3. Click your Output data block to expand

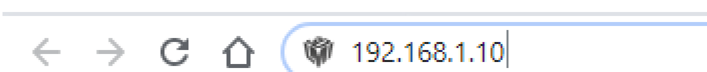


4. Write the integer value 2 to the first byte named **Command**.



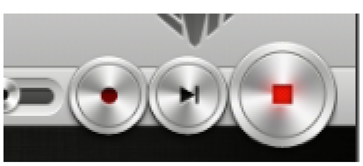
Gocator1:O	{...}
Gocator1:O.Command	2
Gocator1:O.Reserved_or_Job_File1	0
Gocator1:O.Reserved_or_Job_File2	0
Gocator1:O.Reserved_or_Job_File3	0
Gocator1:O.Reserved_or_Job_File4	0
Gocator1:O.Reserved_or_Job_File5	0
Gocator1:O.Reserved_or_Job_File6	0
Gocator1:O.Reserved_or_Job_File7	0
Gocator1:O.Reserved_or_Job_File8	0
Gocator1:O.Reserved_or_Job_File9	0
Gocator1:O.Reserved_or_Job_File10	0
Gocator1:O.Reserved_or_Job_File11	0
Gocator1:O.Reserved_or_Job_File12	0
Gocator1:O.Reserved_or_Job_File13	0
Gocator1:O.Reserved_or_Job_File14	0
Gocator1:O.Reserved_or_Job_File15	0
Gocator1:O.Reserved_or_Job_File16	0
Gocator1:O.Reserved_or_Job_File17	0
Gocator1:O.Reserved_or_Job_File18	0
Gocator1:O.Reserved_or_Job_File19	0
Gocator1:O.Reserved_or_Job_File20	0
Gocator1:O.Reserved_or_Job_File21	0
Gocator1:O.Reserved_or_Job_File22	0
Gocator1:O.Reserved_or_Job_File23	0
Gocator1:O.Reserved_or_Job_File24	0
Gocator1:O.Reserved_or_Job_File25	0
Gocator1:O.Reserved_or_Job_File26	0
Gocator1:O.Reserved_or_Job_File27	0
Gocator1:O.Reserved_or_Job_File28	0
Gocator1:O.Reserved_or_Job_File29	0
Gocator1:O.Reserved_or_Job_File30	0
Gocator1:O.Reserved_or_Job_File31	0

5. Go to a web browser and type in the sensor IP address to the URL bar. This should load the web GUI



6. Verify that the sensor started.

   If the Run button is a red square, then the sensor was successfully started.



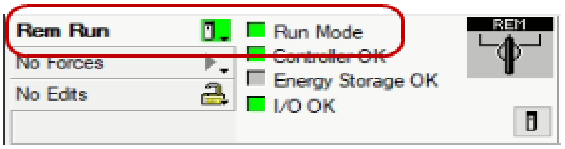
This process can be repeated to stop the sensor, clear alignment, start moving alignment, start stationary alignment, or issue a software trigger by typing the proper integer value into the Command byte of the

Output assembly. For additional commands and control options, *Implicit Messaging Command Assembly* on page 728, or refer to the provided sample Studio 5000 job file.

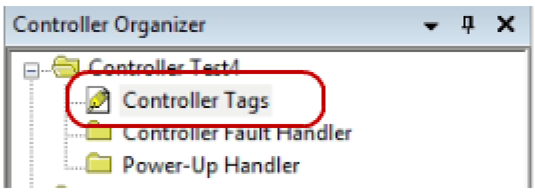
Loading a Sensor Job File

1. Make sure that you have downloaded your PLC program to the controller and that your controller is in Run mode.

   For information on downloading the the PLC program to the controller, see *Install EDS File* on page 734.



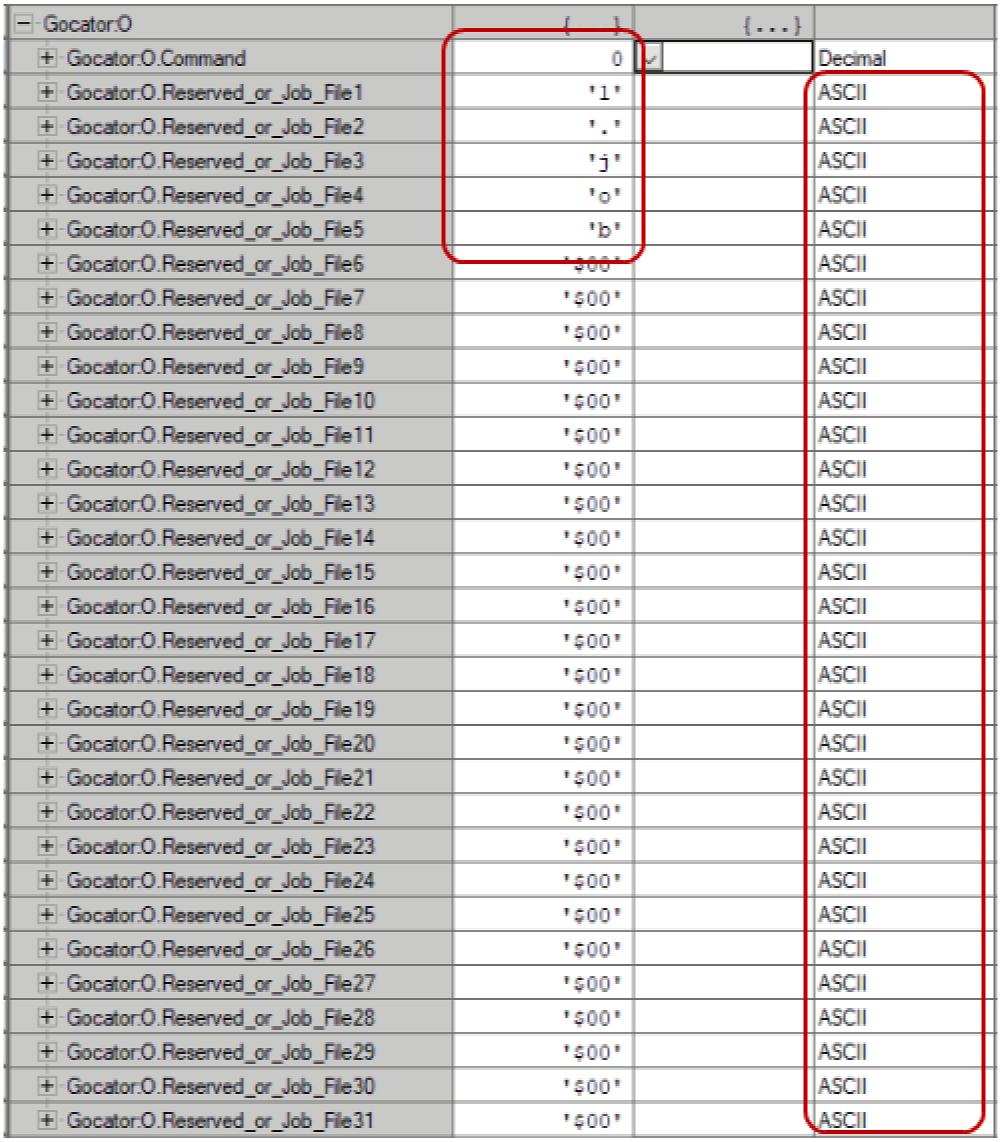
2. Double click **Controller Tags** to show them in the main screen



3. Click your Output data block to expand

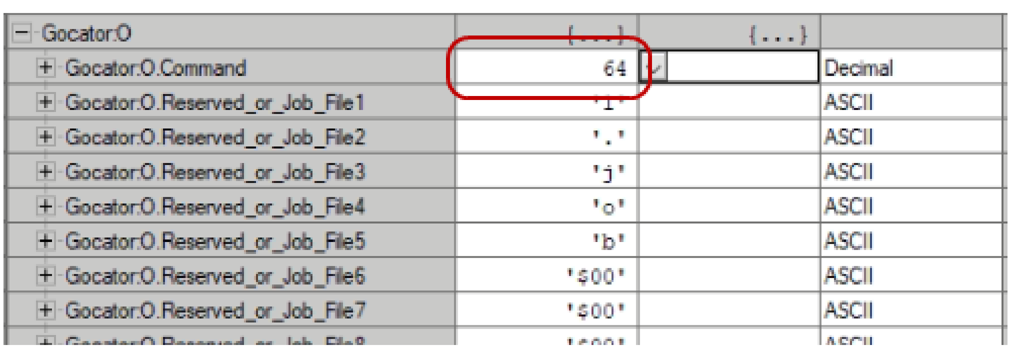
4. If 1.job is the name of the job file to be loaded on the sensor and it is not currently running, type each of the five characters making up the filename into the first five characters of the Reserved bytes of the Command assembly.

   The ASCII character inputs here are case sensitive and the extension, .job, must be included. All non-jobname characters must be null or empty values. Changing the display option from Decimal (which is the default) to ASCII can make this easier.



Gocator:O	{...}	{...}	
Gocator:O.Command	0		Decimal
Gocator:O.Reserved_or_Job_File1	'1'		ASCII
Gocator:O.Reserved_or_Job_File2	'.'		ASCII
Gocator:O.Reserved_or_Job_File3	'j'		ASCII
Gocator:O.Reserved_or_Job_File4	'o'		ASCII
Gocator:O.Reserved_or_Job_File5	'b'		ASCII
Gocator:O.Reserved_or_Job_File6	'$00'		ASCII
Gocator:O.Reserved_or_Job_File7	'$00'		ASCII
Gocator:O.Reserved_or_Job_File8	'$00'		ASCII
Gocator:O.Reserved_or_Job_File9	'$00'		ASCII
Gocator:O.Reserved_or_Job_File10	'$00'		ASCII
Gocator:O.Reserved_or_Job_File11	'$00'		ASCII
Gocator:O.Reserved_or_Job_File12	'$00'		ASCII
Gocator:O.Reserved_or_Job_File13	'$00'		ASCII
Gocator:O.Reserved_or_Job_File14	'$00'		ASCII
Gocator:O.Reserved_or_Job_File15	'$00'		ASCII
Gocator:O.Reserved_or_Job_File16	'$00'		ASCII
Gocator:O.Reserved_or_Job_File17	'$00'		ASCII
Gocator:O.Reserved_or_Job_File18	'$00'		ASCII
Gocator:O.Reserved_or_Job_File19	'$00'		ASCII
Gocator:O.Reserved_or_Job_File20	'$00'		ASCII
Gocator:O.Reserved_or_Job_File21	'$00'		ASCII
Gocator:O.Reserved_or_Job_File22	'$00'		ASCII
Gocator:O.Reserved_or_Job_File23	'$00'		ASCII
Gocator:O.Reserved_or_Job_File24	'$00'		ASCII
Gocator:O.Reserved_or_Job_File25	'$00'		ASCII
Gocator:O.Reserved_or_Job_File26	'$00'		ASCII
Gocator:O.Reserved_or_Job_File27	'$00'		ASCII
Gocator:O.Reserved_or_Job_File28	'$00'		ASCII
Gocator:O.Reserved_or_Job_File29	'$00'		ASCII
Gocator:O.Reserved_or_Job_File30	'$00'		ASCII
Gocator:O.Reserved_or_Job_File31	'$00'		ASCII

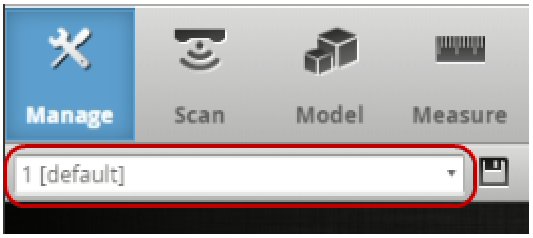
5. Then type the integer value 64 into the **Command** byte to transmit the job name for loading.



Gocator:O	{...}	{...}	
Gocator:O.Command	64		Decimal
Gocator:O.Reserved_or_Job_File1	'1'		ASCII
Gocator:O.Reserved_or_Job_File2	'.'		ASCII
Gocator:O.Reserved_or_Job_File3	'j'		ASCII
Gocator:O.Reserved_or_Job_File4	'o'		ASCII
Gocator:O.Reserved_or_Job_File5	'b'		ASCII
Gocator:O.Reserved_or_Job_File6	'$00'		ASCII
Gocator:O.Reserved_or_Job_File7	'$00'		ASCII

6. Go to a web browser and type in the sensor IP address to the URL bar

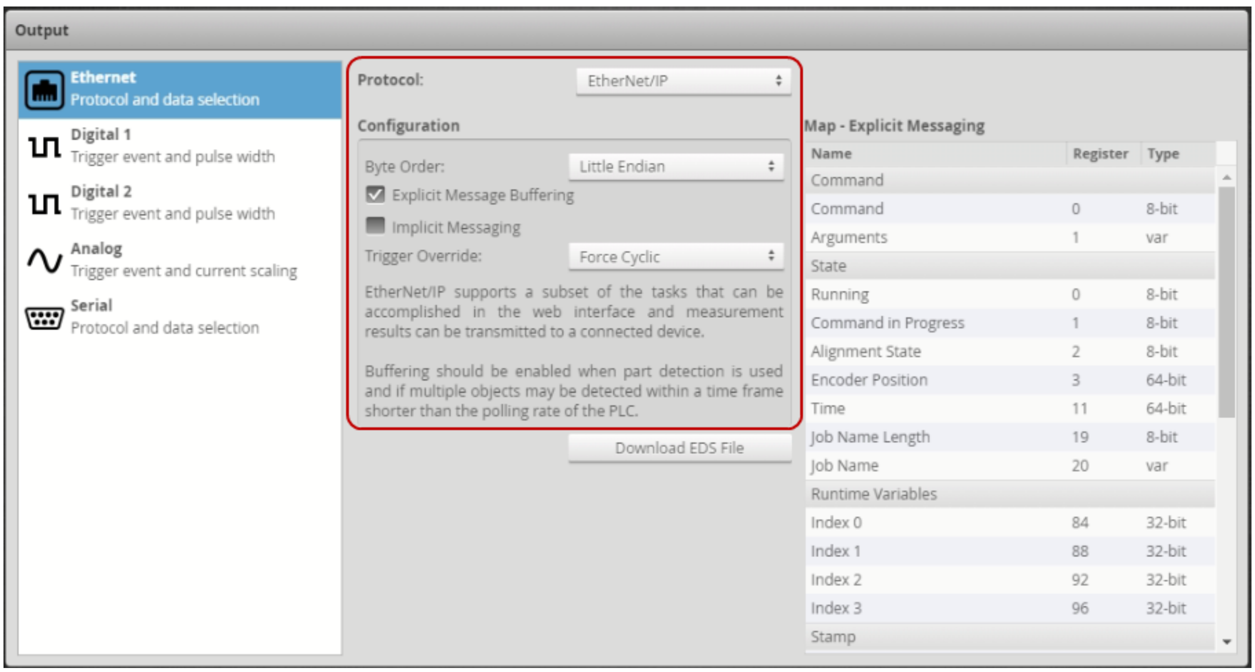
7. Once the web GUI loads, verify that the job was loaded on the Gocator by looking at the job name box



This process can be repeated to load runtime variables by typing the proper integer value into the Command byte of the Output assembly after preloading the runtime variable values into four successive bytes starting at byte 4 of the Reserved bytes. For additional commands and control options, *Implicit Messaging Command Assembly* on page 728, or refer to the provided sample Studio 5000 job file.

Setting Up Explicit Messaging on the Gocator

To output in EtherNet/IP explicit messaging mode on the sensor, you configure the sensor using the **Protocol** setting and the **Configuration** area on the **Output** page.



To configure the sensor for EtherNet/IP explicit messaging mode:

1. On the **Output** page, in the **Ethernet** category, choose **EtherNet/IP** as the protocol.
2. Choose **Little Endian** from the **Byte Order** dropdown box.
3. Check the **Explicit Message Buffering** option.
4. Make sure that **Implicit Messaging** is unchecked.

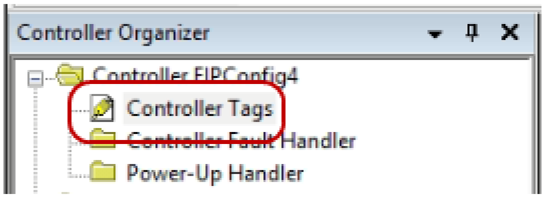
Reading Single Attribute on the PLC (Explicit Messaging)

This section shows how to read the serial number from a Gocator sensor, that is, attribute 6. (For more on the Identity Object, see *Identity Object (Class 0x01)* on page 722.)

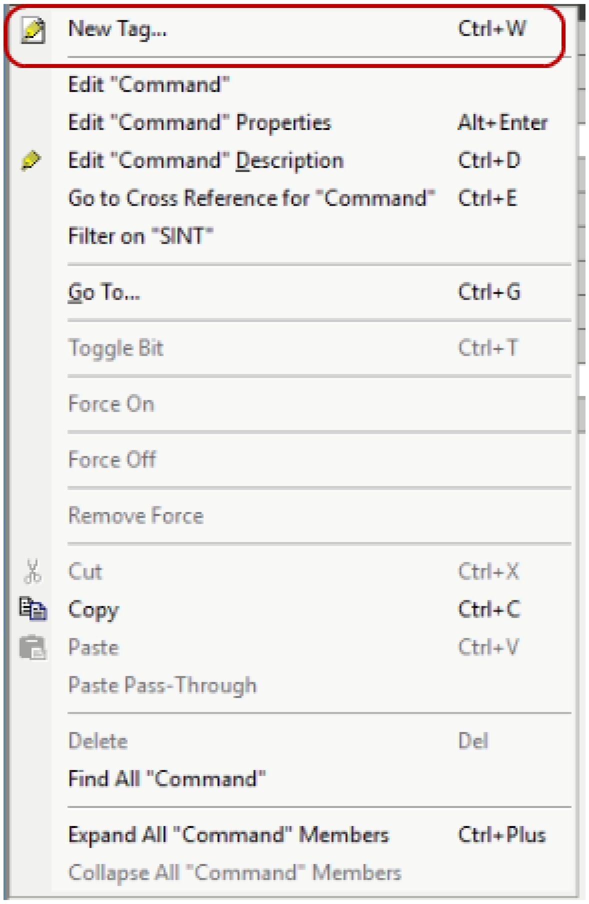
Before attempting to control and run the Gocator from the PLC, you should always verify the connection first by reading an attribute from the Identity Object, for example the sensor's serial number. LMI recommends following the steps described in this section before trying to control the sensor.

To read the sensor's serial number:

1. In Studio 5000, in the Controller Organizer, expand **Controller Tags** by double-clicking it.

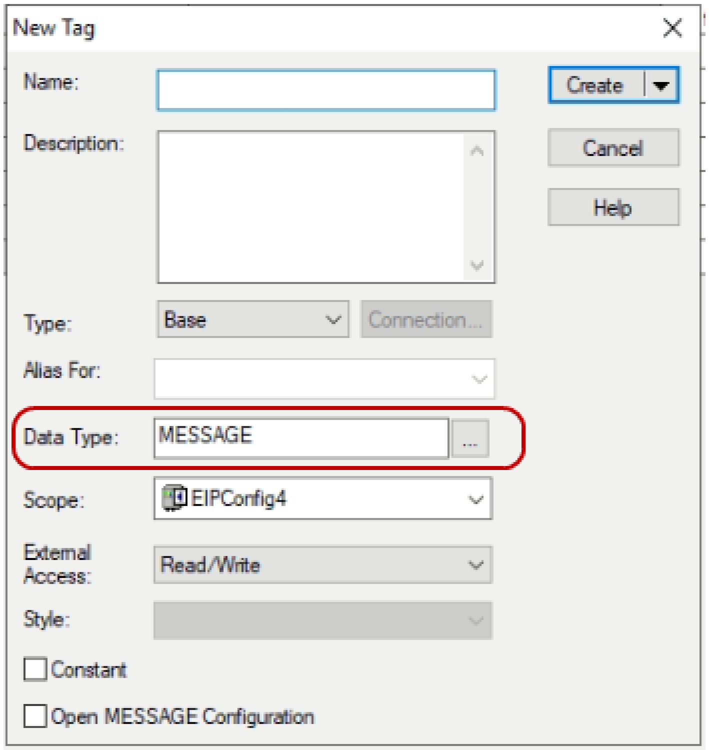


2. Right-click in the middle of the screen and choose **New Tag** from the context menu.

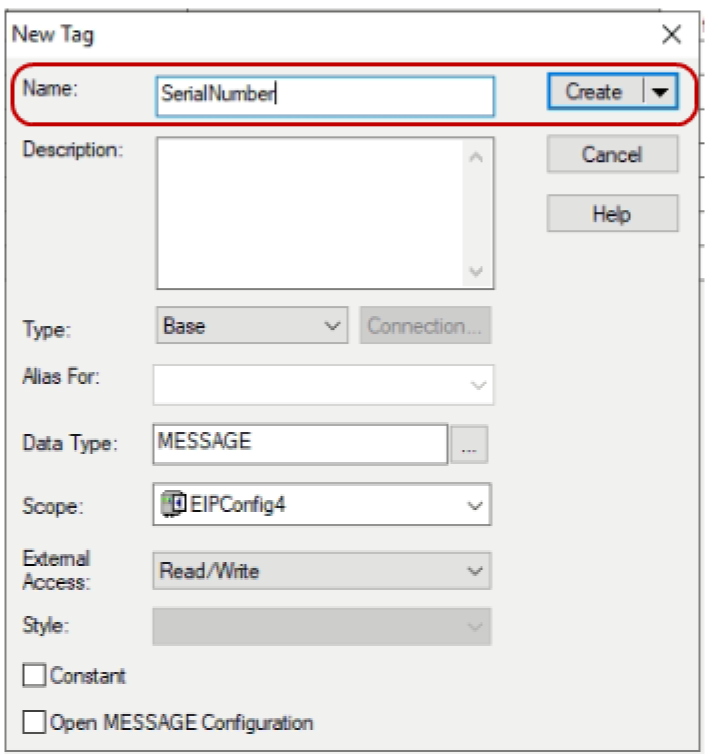


3. In the New Tag dialog, change the data type to MESSAGE.

   This creates a block to store parameters for requesting data from the Gocator.

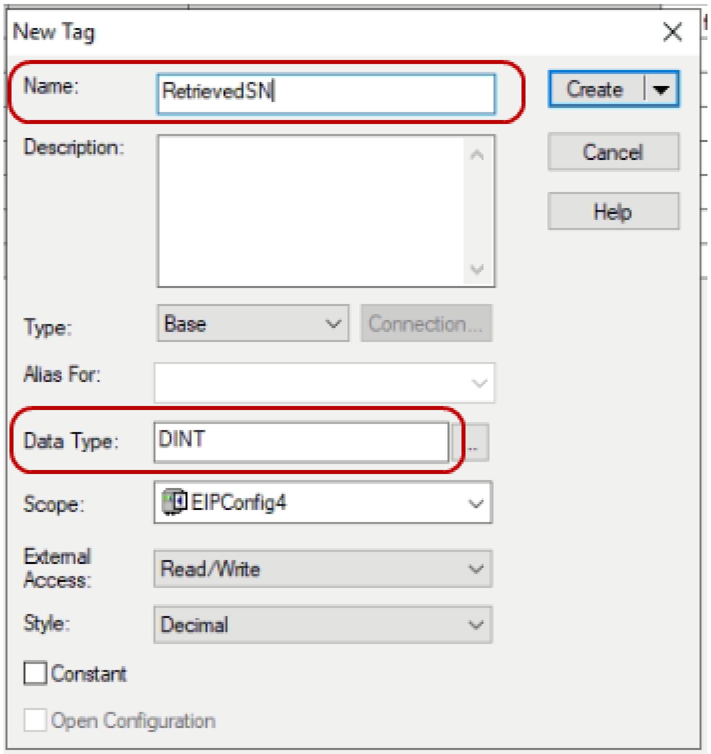


4. Name the tag and click **Create** to the right.



5. Right-click in the middle of the screen again and choose **New Tag** from the context menu.
6. Change the data type to DINT and name the tag.

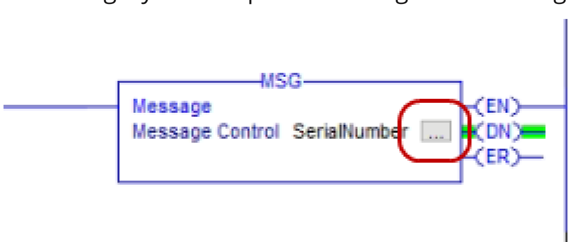
This will create a tag to store the serial number in. The type must match the data type of the attribute you want to get. To determine the type of the attribute, see *Identity Object (Class 0x01)* on page 722.



7. In the ladder, navigate to the Input/Output function blocks and click MSG to add a Message function block. You may need to add a new rung to allow this.

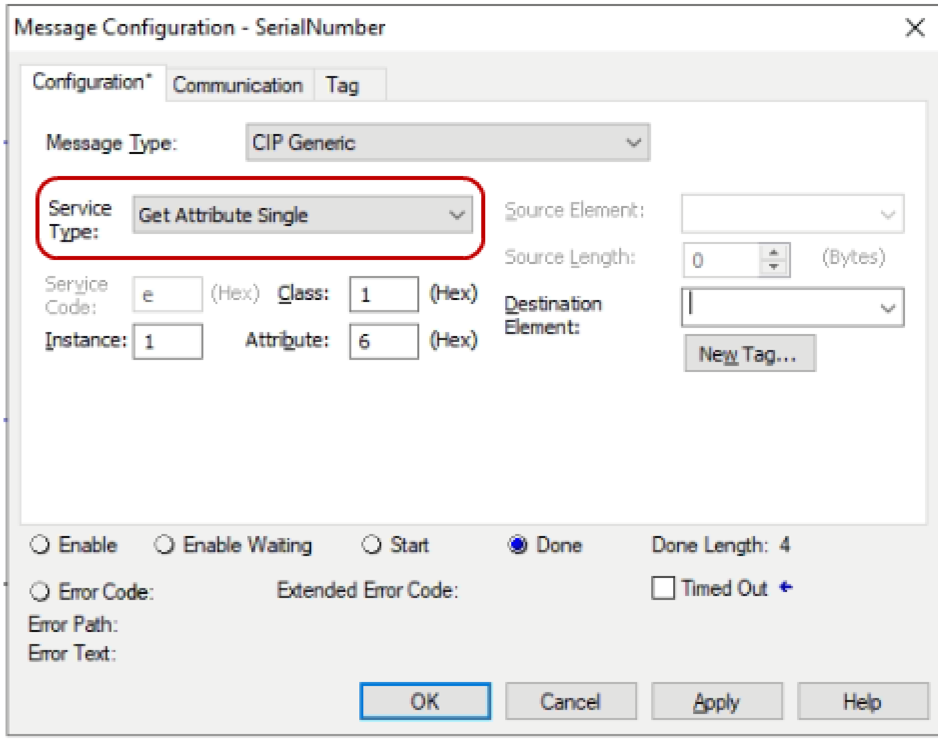
8. Once the new MSG function block has been added, click the tag dropdown and select the MSG tag you created earlier.

9. Click the grey box to open the Configuration Dialog box.



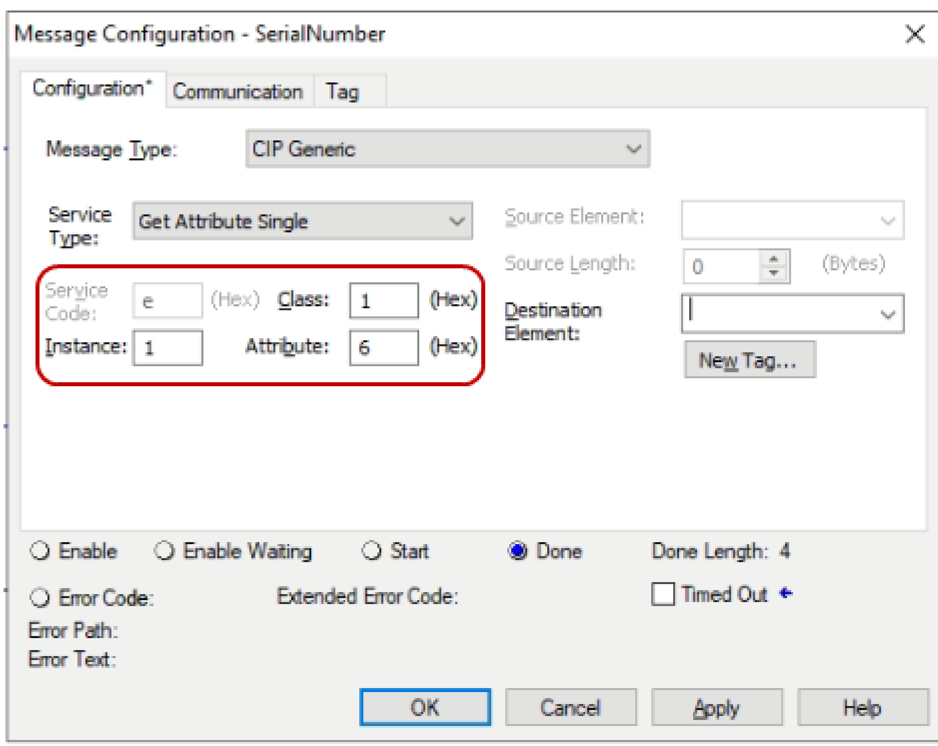
10. Choose the **Get Attribute Single** function from the **Service Type** dropdown.

    This will auto-populate the Service Code hex character.

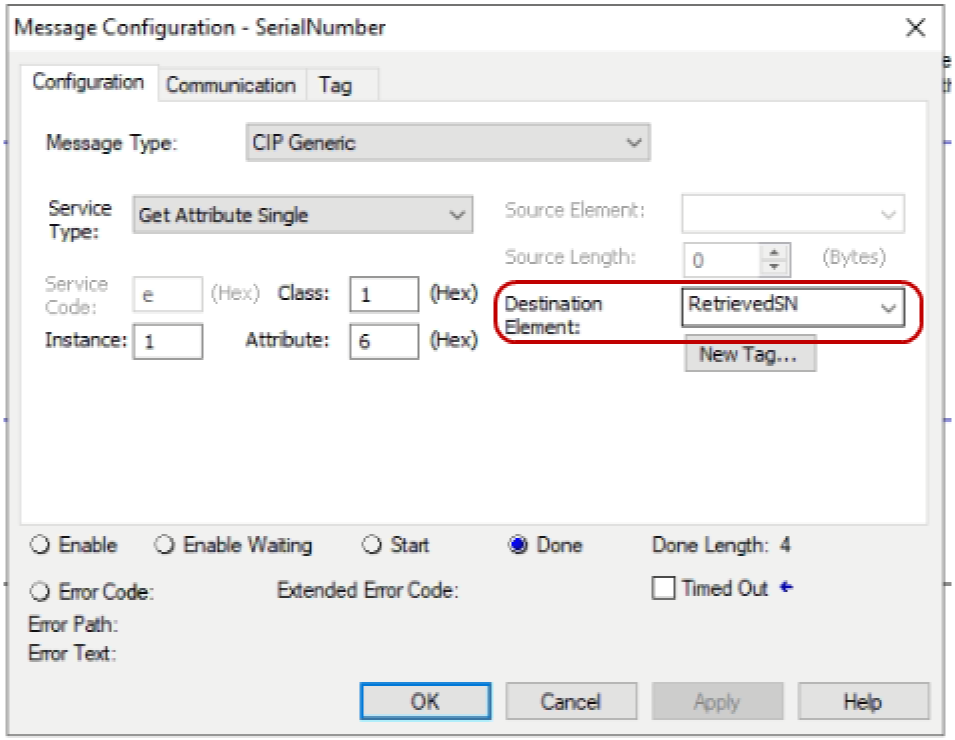


11. Type 1 in **Class**, 1 in **Instance**, and 6 in **Attribute**.

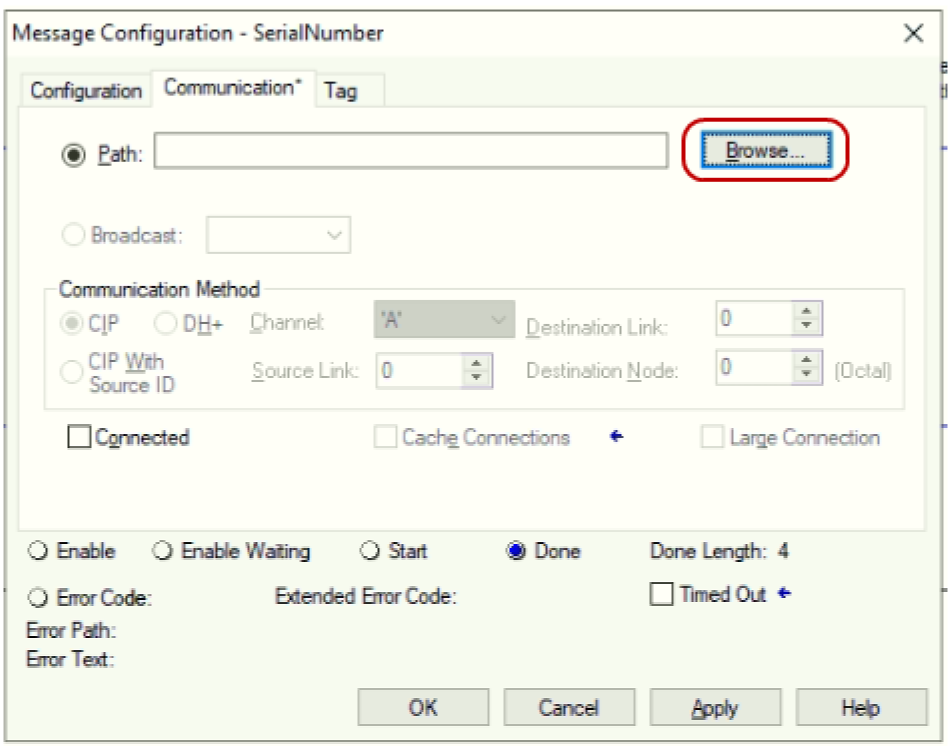
    These settings indicate that the sensor's serial number will be retrieved.



12. Choose the DINT tag you created to store the serial number from the **Destination Element** dropdown.

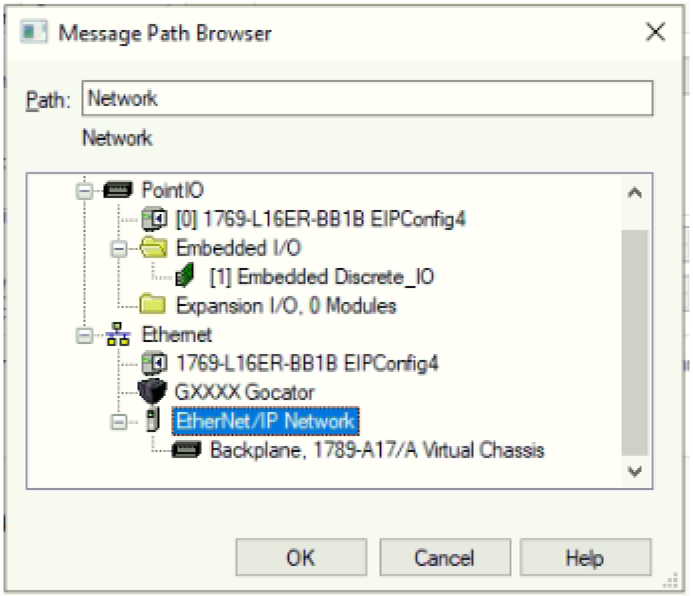


13. On the **Communication** tab, click **Browse**.



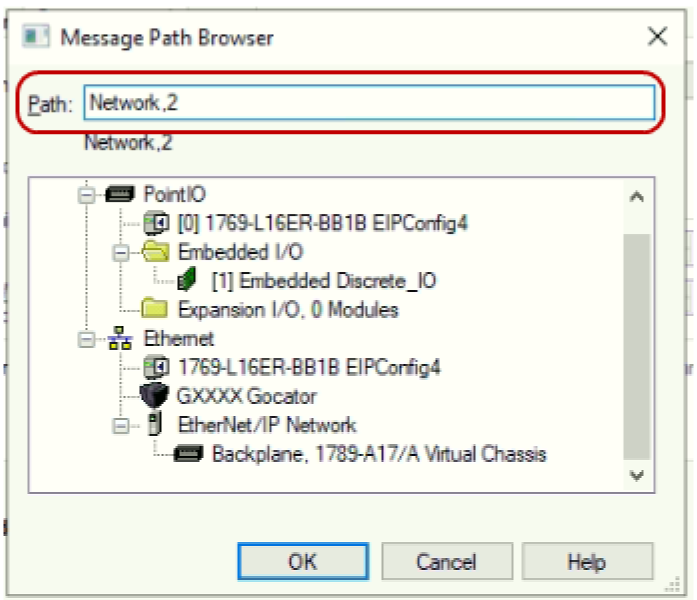
14. In the Message Path Browser dialog, choose the **EtherNet/IP Network** node.

    This will route communication messages to the EtherNet/IP network.



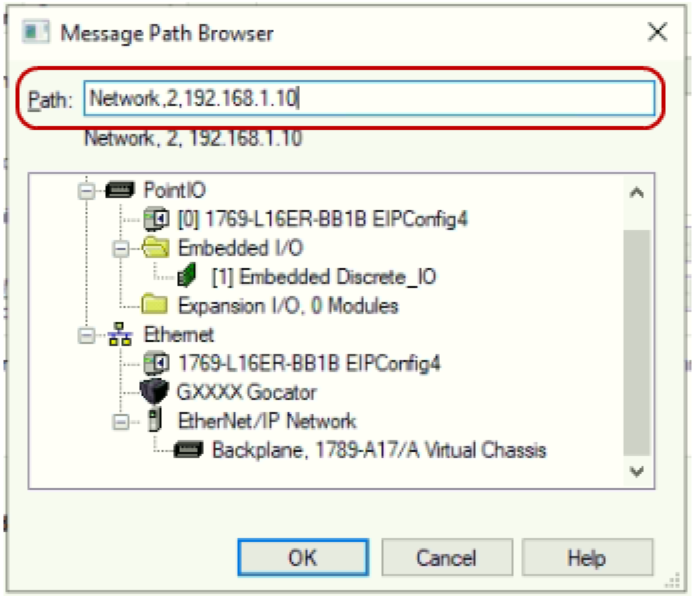
15. In **Path**, type the Ethernet port on the PLC that is physically connected to the Gocator, after the name in the field.

    Here, the port "2" is added.

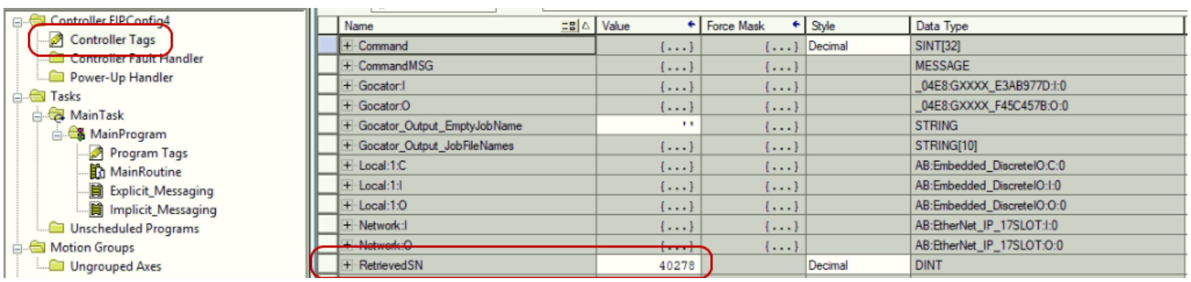


16. In **Path**, type the IP address of the Gocator to complete the path.

    Double-check that the network, port, and IP address are separated by commas in the form "networkname,port,IPaddress".



17. Click **OK** to exit the Message Path Browser dialog, and click **OK** again to exit the Message Configuration dialog.
18. In the Controller Organizer, verify that the serial number is updated in the RetrievedSN tag by going to the Controller Tags node.



To obtain a measurement result, use the procedure described above but change the messaging block class to 4, the instance to 801, and the attribute to 3. The data storage location for this attribute will have to be the proper type and length; for more information, see *Sensor State Assembly* on page 725. You will now have to create ladder logic to copy the correct bits in the raw data stream into Controller Tags holding the individual results. This can be done with the Bit Field Distribute (BTD) block. For

Setting Single Attribute to Gocator on the PLC (Explicit Messaging)

You use the Command assembly to do the following:

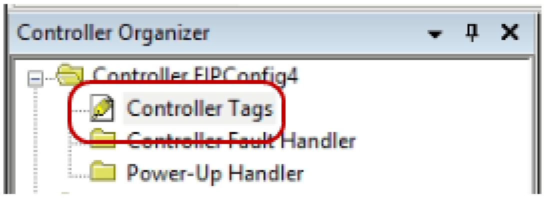
- Start a sensor
- Stop a sensor
- Align a sensor
- Clear sensor alignment
- Set a sensor's runtime variables
- Load a job on a sensor
- Trigger a sensor

To see the information needed to properly configure the control byte, see *Command Assembly* on page 723.

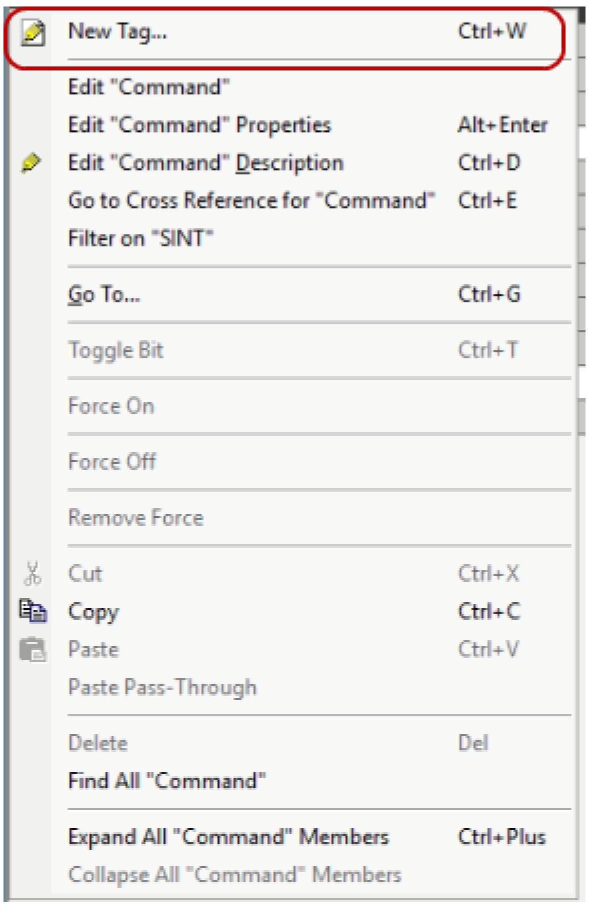
🗋	LMI recommends following the steps in *To read the sensor's serial number:* on page 758 to verify the communication path and message block *before* attempting to control a sensor.

To set a single attribute to the sensor on the PLC, do the following:

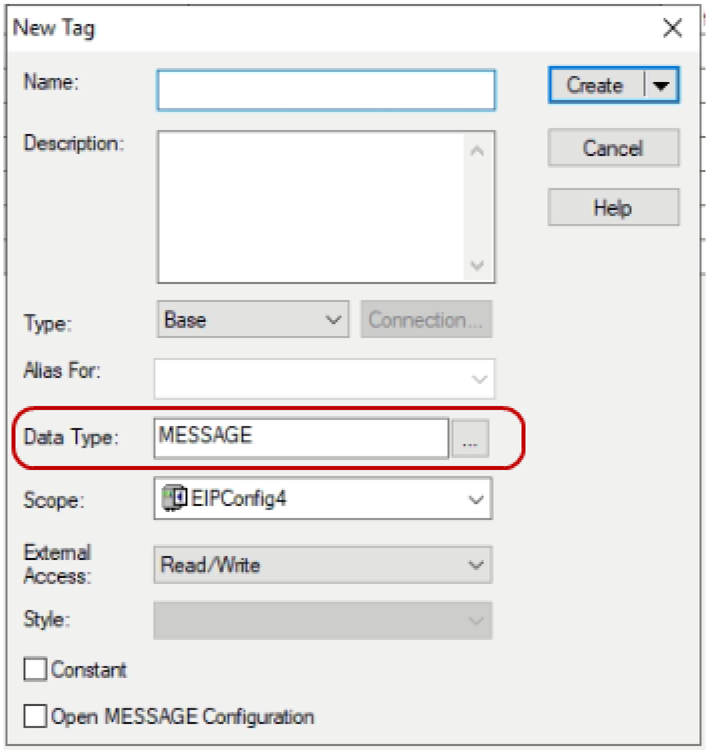
1. In Studio 5000, in the Controller Organizer, expand Controller Tags by double-clicking it.



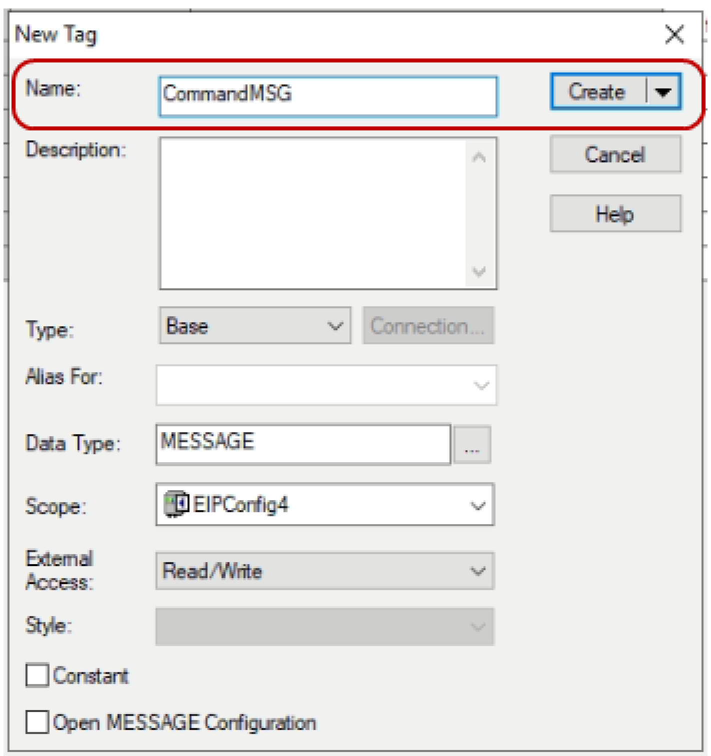
2. Right-click in the middle of the screen and select **New Tag** from the context menu.



3. Change the data type to MESSAGE.

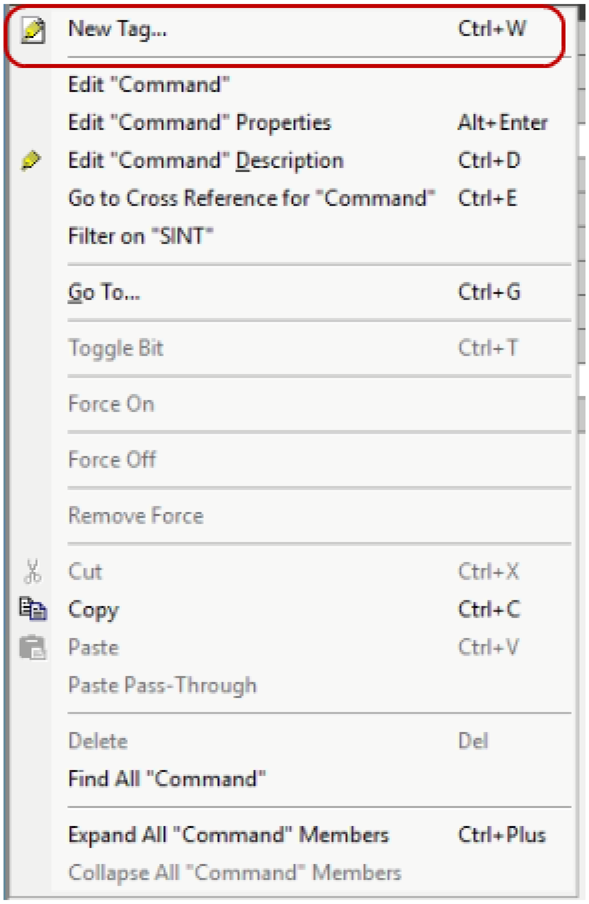


4. Name the tag and click Create.

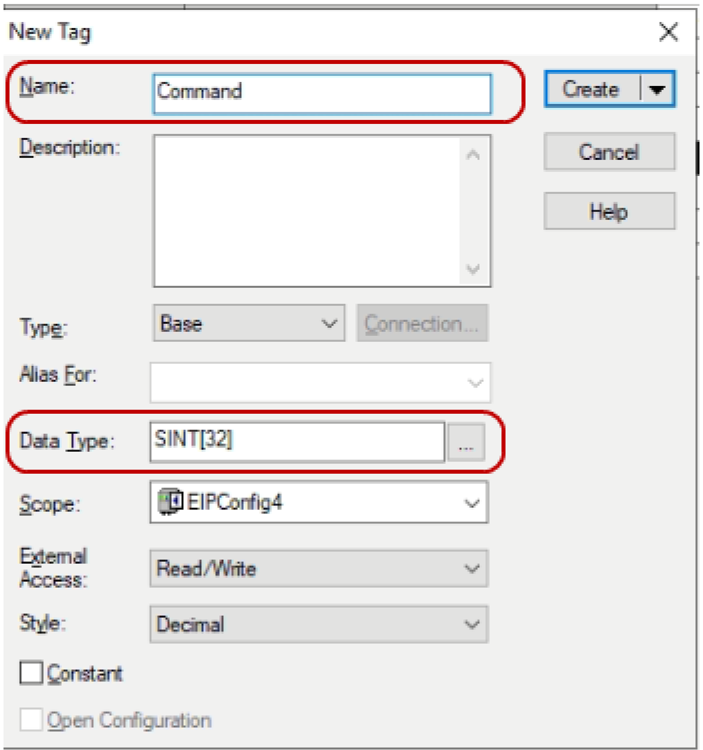


This creates a block to store parameters for sending data to the Gocator.

5. Right-click in the middle of the screen *again* and choose **New Tag** from the context menu.

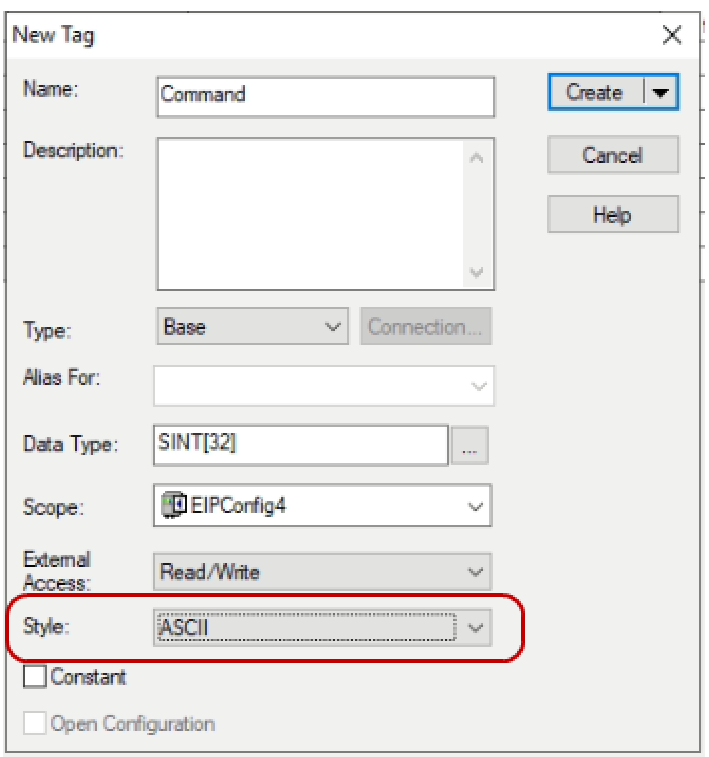


6. Change the data type to SINT[32] and name the tag.



7. Set Style to one of the following:

   If you will be loading job files on the Gocator over the protocol, change **Style** from the default to **ASCII**. This will make editing the command assembly easier later.



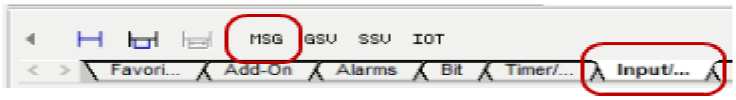
   If you will only be starting or stopping the sensor, leave **Style** at the default setting of **Decimal**.

8. Click **Create**.

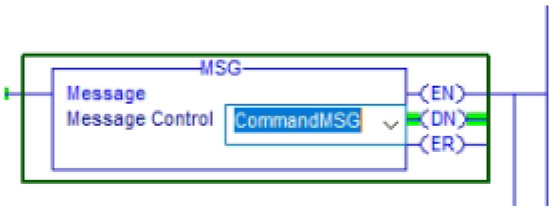
   This creates a tag to store the command data before sending it.

9. In the ladder, navigate to the Input/Output function blocks and click MSG to add a Message function block.

   You may need to add a new rung to allow this.



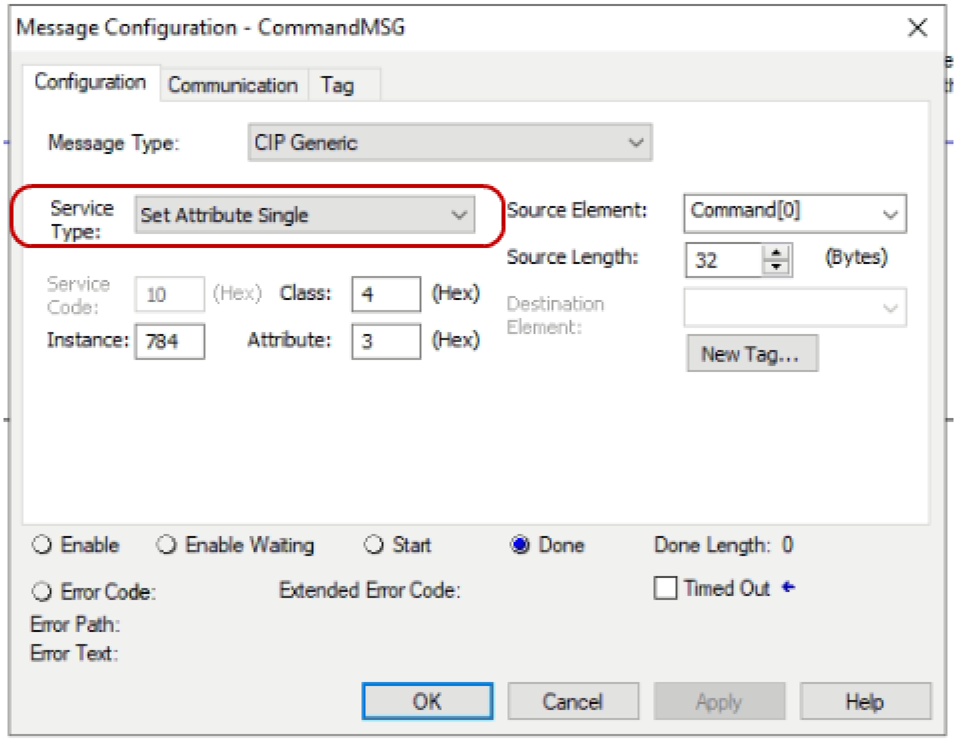
10. Once the new MSG function block has been added, click the tag dropdown and select the MSG tag you created earlier.



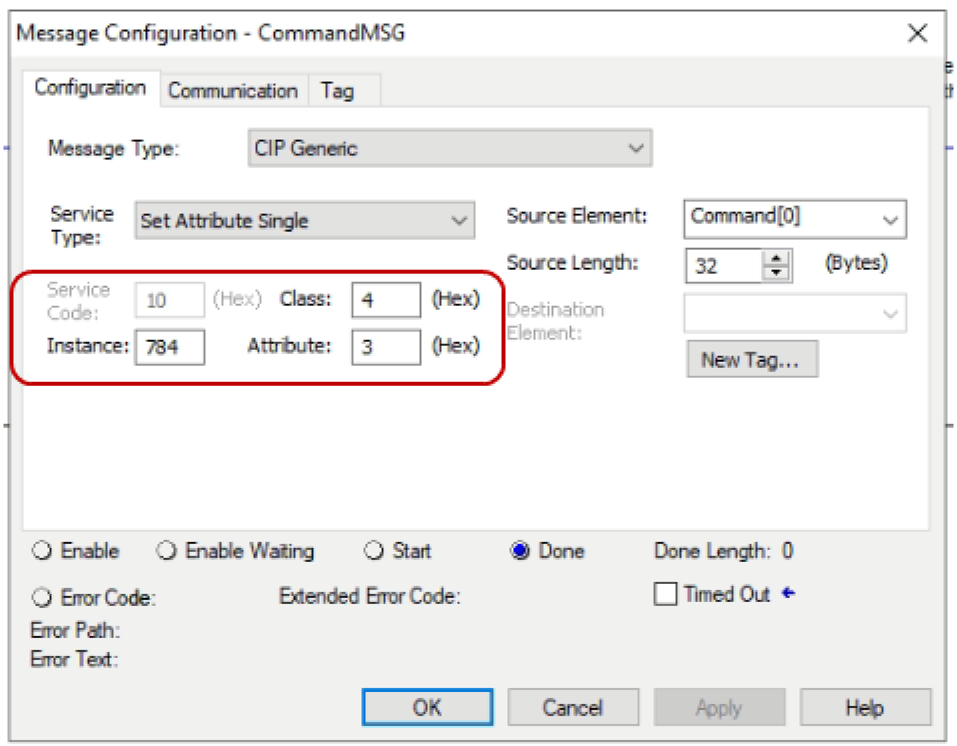
11. Click the grey box to open the Configuration Dialog box

12. In the Message Configuration dialog, choose the **Set Attribute Single** function from the **Service Type** drop-down.

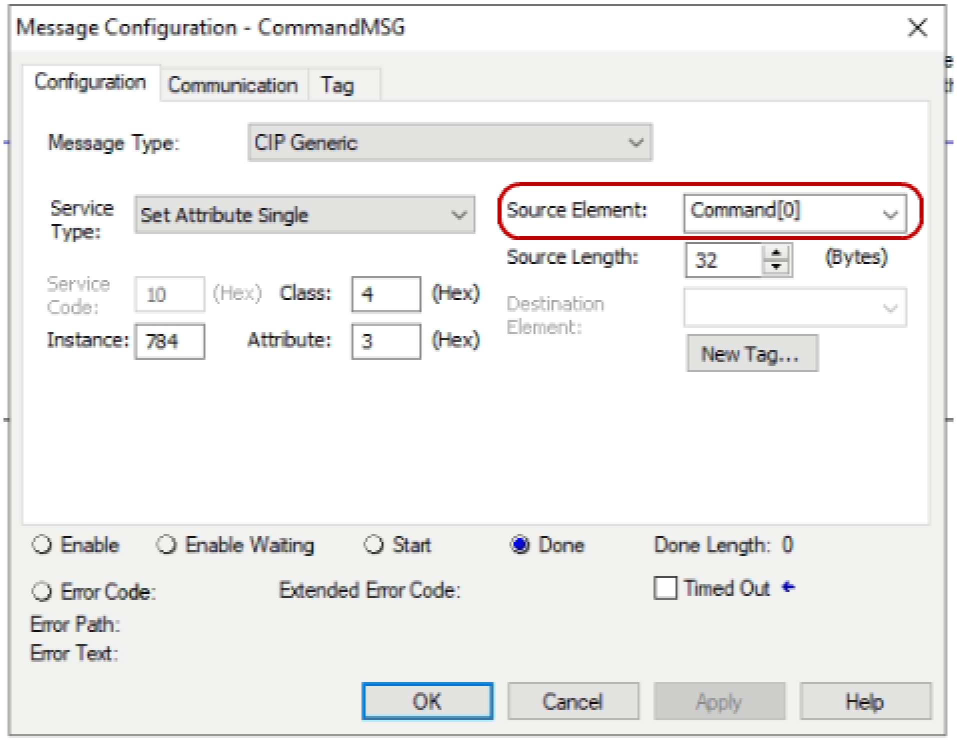
    This will auto populate the Service Code hex character.



13. Enter 4 for Class, 784 for Instance, and 3 for Attribute to set the sensor's command assembly.

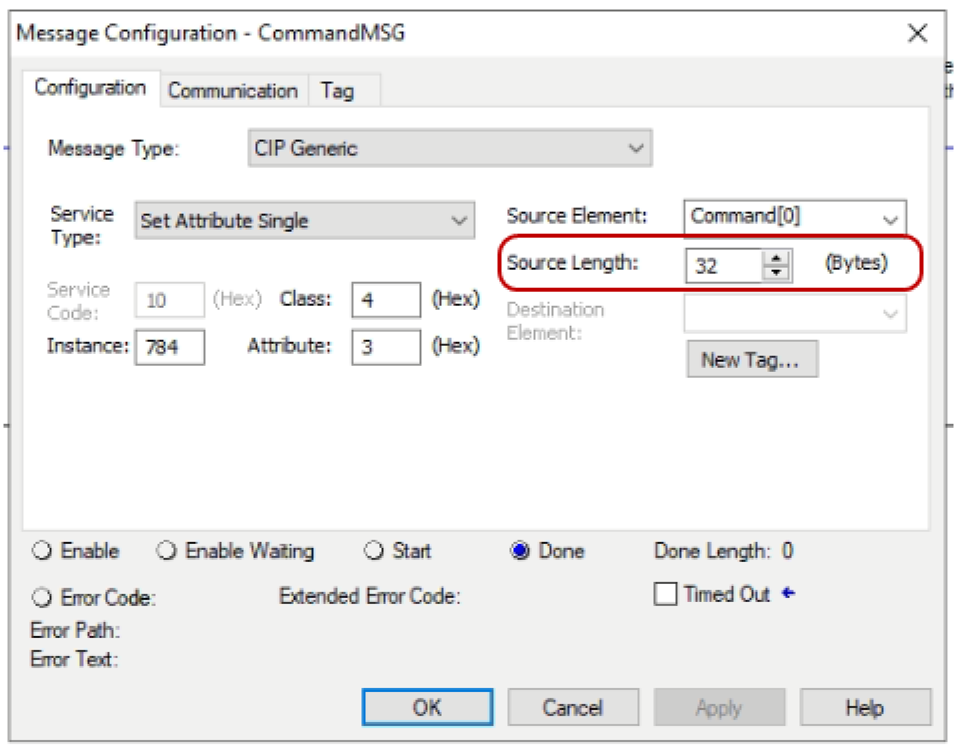


14. Select the SINT[32] tag you created to store the command assembly from the Destination Element dropdown

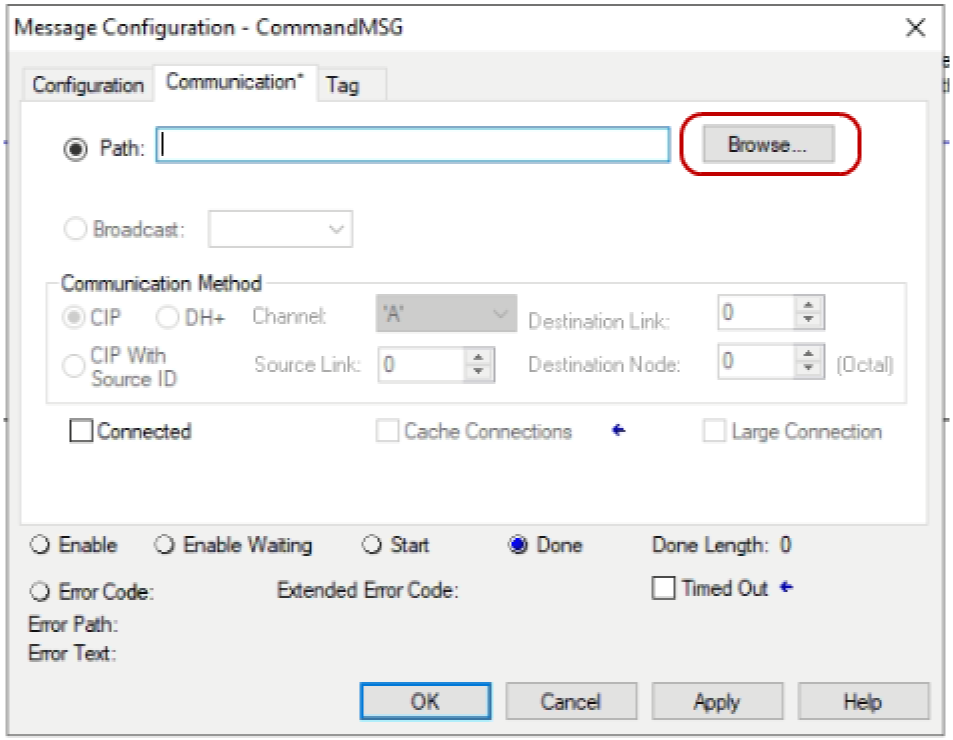


15. Make sure that the length is set to 32 bytes so that the entire command assembly is transmitted.

    A partial transmission may result in an unexecuted command.

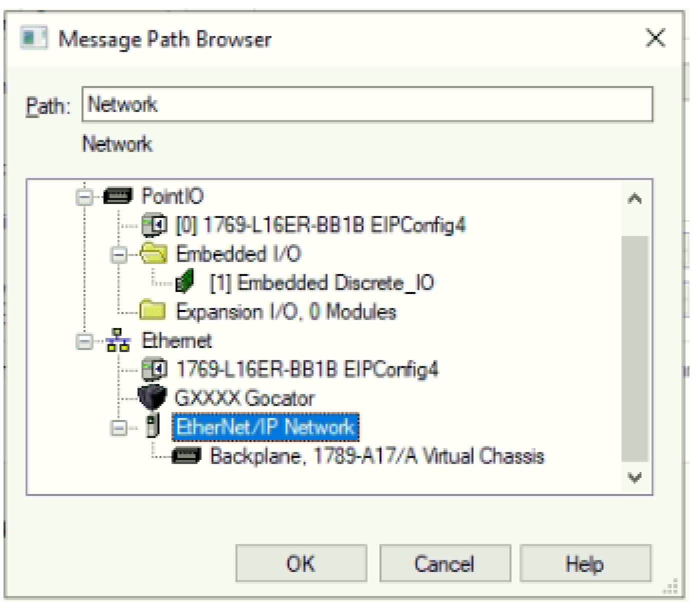


16. On the **Communication** tab, click **Browse**



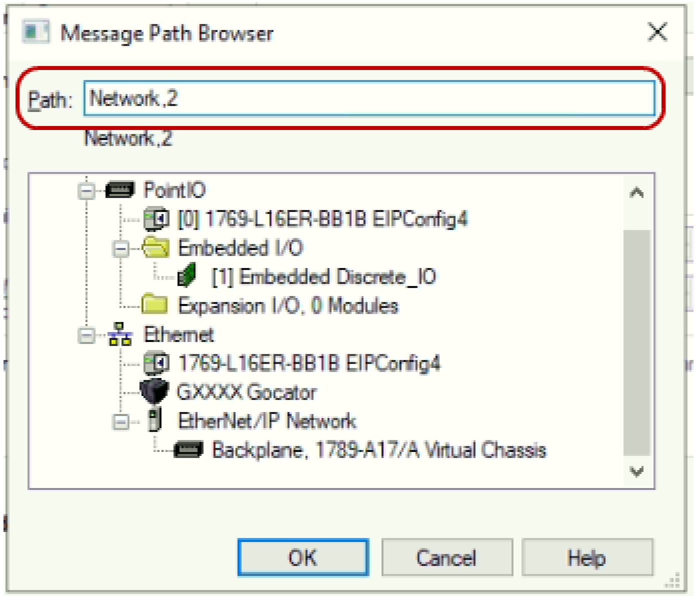
17. Click the **EtherNet/IP Network** node.

    This will route communication messages to the EtherNet/IP network.



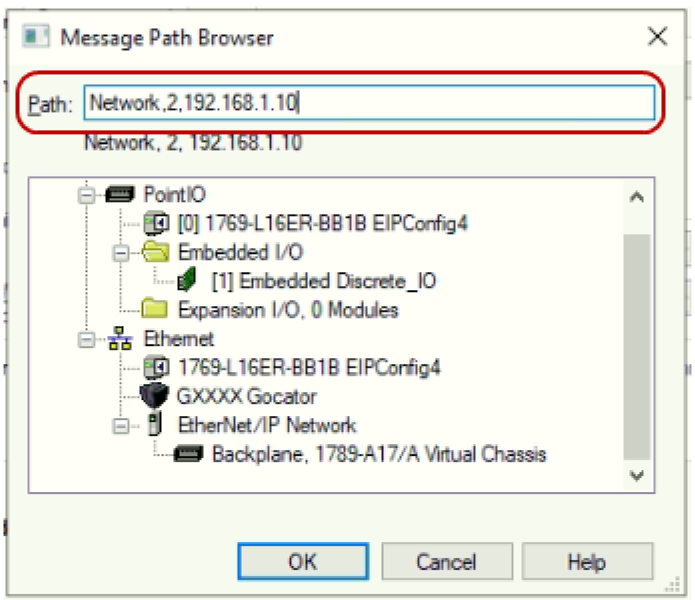
18. Add the ethernet port that is physically connected to the Gocator.

    This will add the specific port address to your communication path.

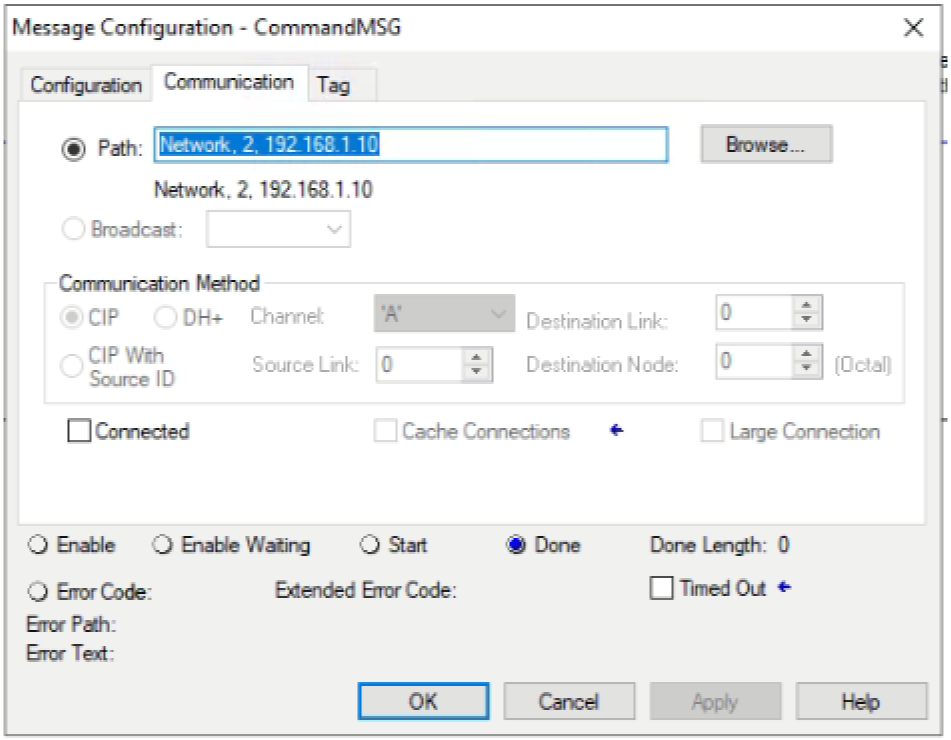


19. Type the IP address of the Gocator to complete the path.

    It is important to double-check that the network, port, and IP address are separated by commas in the form "networkname,port,IPaddress".

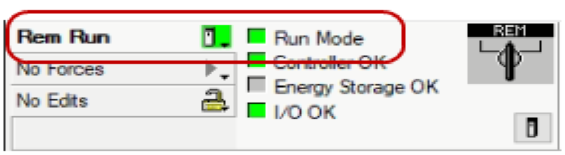


20. Click **OK**.
21. Click **OK** to exit the Message Configuration dialog.

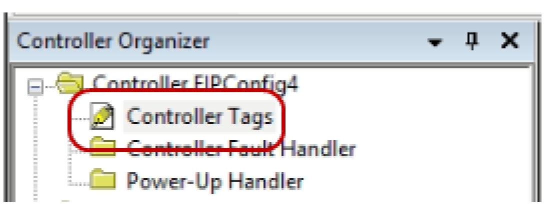


To start a sensor over explicit messaging, the Command assembly must be correctly modified for the integer-based command byte.

1. Make sure that you have downloaded your PLC program to the controller and that your controller is in Run Mode.



2. Expand Controller Tags by double-clicking it.



3. Expand the Command assembly tag.

Name	Value	Force Mask	Style	Data Type
Command	{...}	{...}	ASCII	SINT[32]
Command[0]	'$00'		ASCII	SINT
Command[1]	'$00'		ASCII	SINT
Command[2]	'$00'		ASCII	SINT
Command[3]	'$00'		ASCII	SINT
Command[4]	'$00'		ASCII	SINT
Command[5]	'$00'		ASCII	SINT
Command[6]	'$00'		ASCII	SINT
Command[7]	'$00'		ASCII	SINT
Command[8]	'$00'		ASCII	SINT
Command[9]	'$00'		ASCII	SINT
Command[10]	'$00'		ASCII	SINT
Command[11]	'$00'		ASCII	SINT
Command[12]	'$00'		ASCII	SINT
Command[13]	'$00'		ASCII	SINT
Command[14]	'$00'		ASCII	SINT
Command[15]	'$00'		ASCII	SINT
Command[16]	'$00'		ASCII	SINT
Command[17]	'$00'		ASCII	SINT
Command[18]	'$00'		ASCII	SINT
Command[19]	'$00'		ASCII	SINT
Command[20]	'$00'		ASCII	SINT
Command[21]	'$00'		ASCII	SINT
Command[22]	'$00'		ASCII	SINT
Command[23]	'$00'		ASCII	SINT
Command[24]	'$00'		ASCII	SINT
Command[25]	'$00'		ASCII	SINT
Command[26]	'$00'		ASCII	SINT
Command[27]	'$00'		ASCII	SINT
Command[28]	'$00'		ASCII	SINT
Command[29]	'$00'		ASCII	SINT
Command[30]	'$00'		ASCII	SINT
Command[31]	'$00'		ASCII	SINT

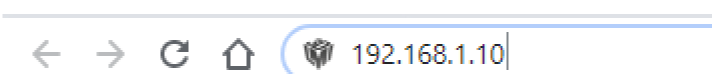
If you changed the formatting of the Command tag array to ASCII, then change the display of only the first byte, Command[0], back to Decimal as the control command are sent as integer-based values.

Command	{...}	{...}	ASCII	SINT[32]
Command[0]	0		Decimal	SINT
Command[1]	'$00'		ASCII	SINT
Command[2]	'$00'		ASCII	SINT

4. Type the number 1 into the value field of Command[0].

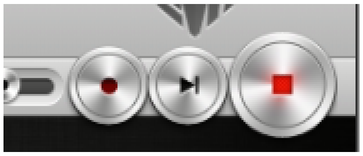
Command	{...}	{...}	ASCII	SINT[32]
Command[0]	1		Decimal	SINT
Command[1]	'$00'		ASCII	SINT
Command[2]	'$00'		ASCII	SINT
Command[3]	'$00'		ASCII	SINT
Command[4]	'$00'		ASCII	SINT

5. Go to a web browser and type in the sensor IP address to the URL bar. This should load the web GUI



6. Verify that the sensor started.

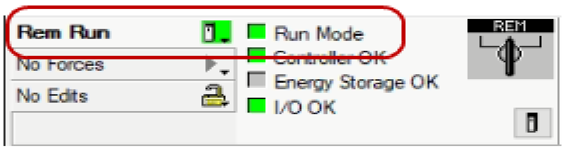
   If the Run button is a red square, then the sensor was successfully started.



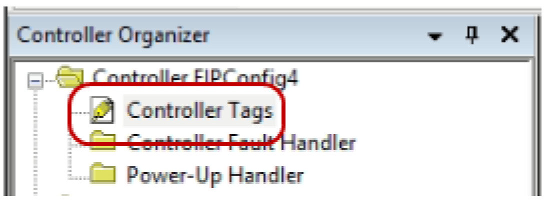
Your ladder logic should only be able to edit the Command assembly 1 time. Since Explicit Message Buffering is checked from the Gocator setup, multiple message transfers from improper ladder logic will end up buffering on the Gocator side of the network. The only way to easily clear the messaging buffer is to power cycle the sensor.

Loading a Sensor Job File

1. Make sure that you have downloaded your PLC program to the controller and that your controller is in Run Mode



2. Expand **Controller Tags** by double-clicking it



3. Expand the Command assembly tag

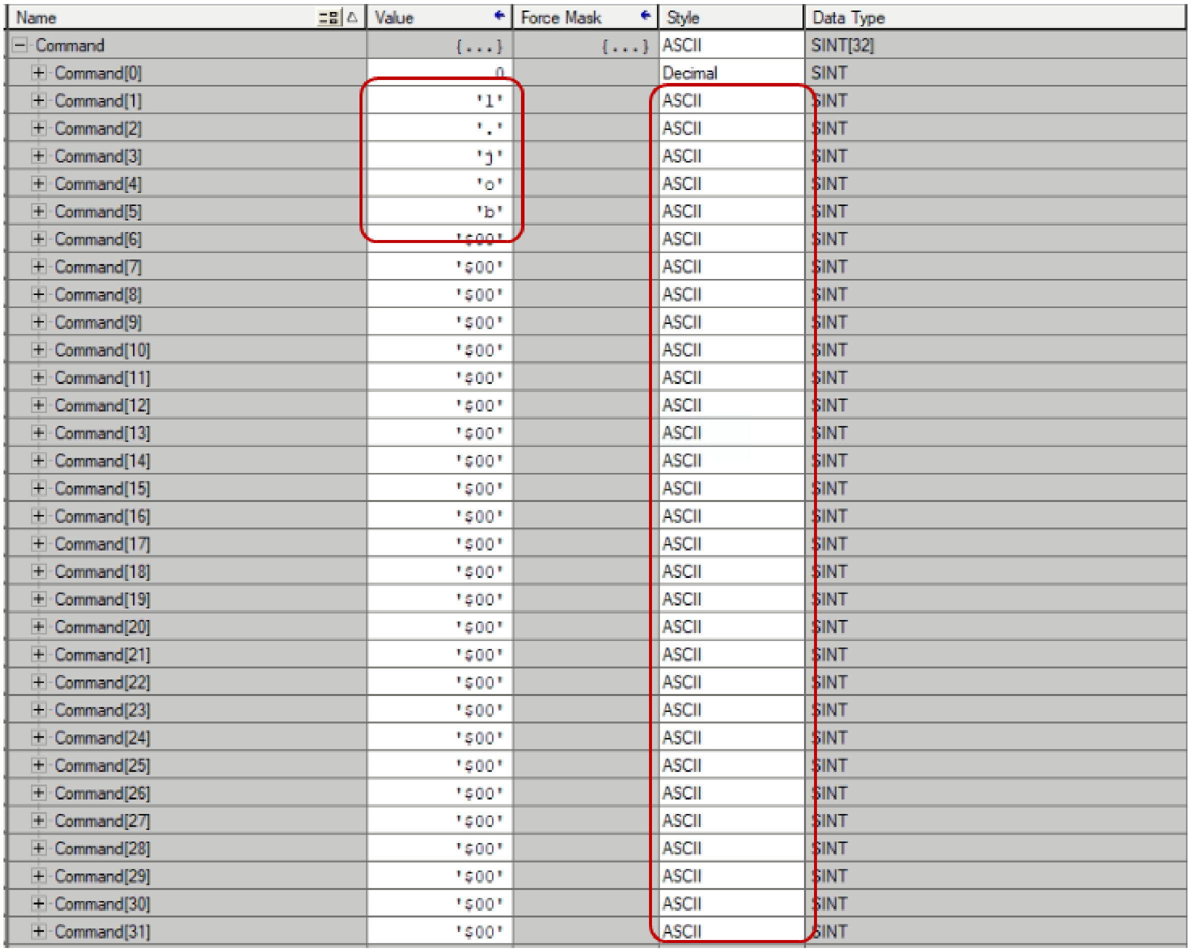
Name	Value	Force Mask	Style	Data Type
− Command	{...}	{...}	ASCII	SINT[32]
+ Command[0]	'$00'		ASCII	SINT
+ Command[1]	'$00'		ASCII	SINT
+ Command[2]	'$00'		ASCII	SINT
+ Command[3]	'$00'		ASCII	SINT
+ Command[4]	'$00'		ASCII	SINT
+ Command[5]	'$00'		ASCII	SINT
+ Command[6]	'$00'		ASCII	SINT
+ Command[7]	'$00'		ASCII	SINT
+ Command[8]	'$00'		ASCII	SINT
+ Command[9]	'$00'		ASCII	SINT
+ Command[10]	'$00'		ASCII	SINT
+ Command[11]	'$00'		ASCII	SINT
+ Command[12]	'$00'		ASCII	SINT
+ Command[13]	'$00'		ASCII	SINT
+ Command[14]	'$00'		ASCII	SINT
+ Command[15]	'$00'		ASCII	SINT
+ Command[16]	'$00'		ASCII	SINT
+ Command[17]	'$00'		ASCII	SINT
+ Command[18]	'$00'		ASCII	SINT
+ Command[19]	'$00'		ASCII	SINT
+ Command[20]	'$00'		ASCII	SINT
+ Command[21]	'$00'		ASCII	SINT
+ Command[22]	'$00'		ASCII	SINT
+ Command[23]	'$00'		ASCII	SINT
+ Command[24]	'$00'		ASCII	SINT
+ Command[25]	'$00'		ASCII	SINT
+ Command[26]	'$00'		ASCII	SINT
+ Command[27]	'$00'		ASCII	SINT
+ Command[28]	'$00'		ASCII	SINT
+ Command[29]	'$00'		ASCII	SINT
+ Command[30]	'$00'		ASCII	SINT
+ Command[31]	'$00'		ASCII	SINT

If you changed the formatting of the Command tag array to ASCII, then change the display of only the first byte, Command[0], back to Decimal as the control command are sent as integer-based values.

− Command	{...}	{...}	ASCII	SINT[32]
+ Command[0]	0		Decimal	SINT
+ Command[1]	'$00'		ASCII	SINT
+ Command[2]	'$00'		ASCII	SINT

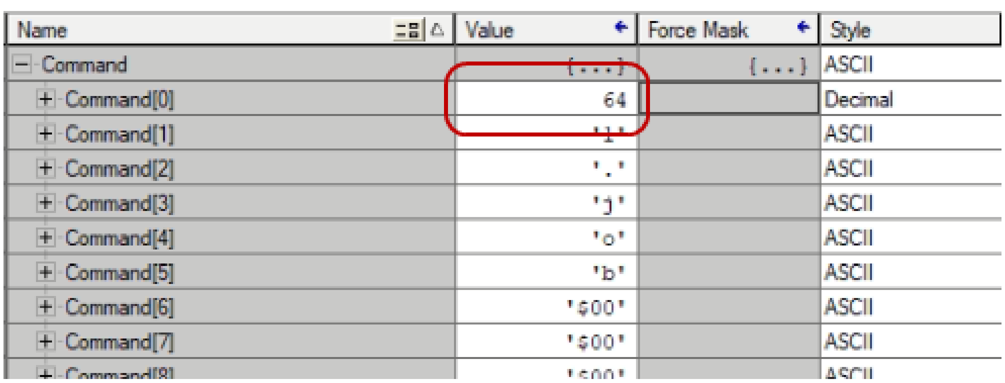
4. If 1.job is the job file to be loaded on the sensor and it is not currently running, type each of the five characters making up the filename into Command[1] through Command[5] of the Command assembly.

   The ASCII character inputs here are case sensitive and the extension, .job, must be included. All non-jobname characters must be null or empty values. If the style was changed to ASCII as the default during the tag creation, this will be done already, and the alphanumeric characters can be directly typed into the value column of the bytes.



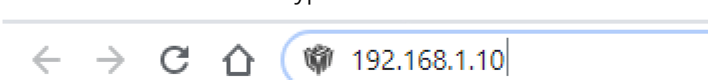
Name	Value	Force Mask	Style	Data Type
Command	{...}	{...}	ASCII	SINT[32]
Command[0]	0		Decimal	SINT
Command[1]	'1'		ASCII	SINT
Command[2]	'.'		ASCII	SINT
Command[3]	'j'		ASCII	SINT
Command[4]	'o'		ASCII	SINT
Command[5]	'b'		ASCII	SINT
Command[6]	'$00'		ASCII	SINT
Command[7]	'$00'		ASCII	SINT
Command[8]	'$00'		ASCII	SINT
Command[9]	'$00'		ASCII	SINT
Command[10]	'$00'		ASCII	SINT
Command[11]	'$00'		ASCII	SINT
Command[12]	'$00'		ASCII	SINT
Command[13]	'$00'		ASCII	SINT
Command[14]	'$00'		ASCII	SINT
Command[15]	'$00'		ASCII	SINT
Command[16]	'$00'		ASCII	SINT
Command[17]	'$00'		ASCII	SINT
Command[18]	'$00'		ASCII	SINT
Command[19]	'$00'		ASCII	SINT
Command[20]	'$00'		ASCII	SINT
Command[21]	'$00'		ASCII	SINT
Command[22]	'$00'		ASCII	SINT
Command[23]	'$00'		ASCII	SINT
Command[24]	'$00'		ASCII	SINT
Command[25]	'$00'		ASCII	SINT
Command[26]	'$00'		ASCII	SINT
Command[27]	'$00'		ASCII	SINT
Command[28]	'$00'		ASCII	SINT
Command[29]	'$00'		ASCII	SINT
Command[30]	'$00'		ASCII	SINT
Command[31]	'$00'		ASCII	SINT

5. Type the integer value 64 into the Command byte to transmit the job name for loading.

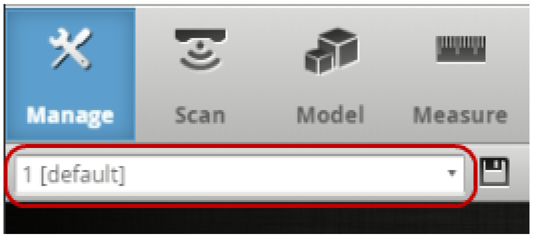


Name	Value	Force Mask	Style
Command	{...}	{...}	ASCII
Command[0]	64		Decimal
Command[1]	'1'		ASCII
Command[2]	'.'		ASCII
Command[3]	'j'		ASCII
Command[4]	'o'		ASCII
Command[5]	'b'		ASCII
Command[6]	'$00'		ASCII
Command[7]	'$00'		ASCII
Command[8]	'$00'		ASCII

6. Go to a web browser and type in the sensor IP address to the URL bar.



7. Once the web GUI loads, verify that the job was loaded on the Gocator by looking at the job name box.


Manage
Scan
Model
Measure
1 [default]

PROFINET Protocol

PROFINET is an Industrial Ethernet network protocol that allows controllers such as PLCs to communicate with sensors. Sensors are PROFINET IO devices with Conformance Class A.

> The Gocator emulator and accelerator (software and GoMax) do not support the PROFINET protocol.

This section describes the PROFINET modules that let a controller do the following:

- Switch jobs.
- Align and run sensors.
- Receive sensor states, stamps, and measurement results.
- Set and retrieve runtime variables.

To use the PROFINET protocol, it must be enabled and configured in the active job. For information on configuring the protocol using the Web interface, see *Ethernet Output* on page 512.

> The Gocator 4.x/5.x firmware uses mm, mm^2, mm^3, and degrees as standard units. In all protocols, values are scaled by 1000, as values in the protocols are represented as integers. This results in effective units of mm/1000, mm^2/1000, mm^3/1000, and deg/1000 in the protocols.

Control Module

The client sends the Control module to the sensor. The length of the Control module is 256 bytes. Unused space is for future expansion.

Control Module Elements

Byte Index	Type	Description
0	Command Register	Takes a 8-bit command as given in the table below.
1-64	Command Parameters. (Job filename in the case of command 5)	For command 5, these registers contains the null terminated job file name. The ".job" extension is optional.

Command Definitions

Value	Name	Description
0	Stop running	Stop the sensor. If already stopped, do nothing
1	Start Running	Start the sensor. If already running, do nothing
2	Stationary Alignment	Start the stationary alignment process. State register 301 will be set to 1 (busy) until the alignment process is complete, then back to zero.
3	Moving Alignment	Start the moving alignment process. State register 301 will be set to 1 (busy) until the alignment process is complete, then back to zero.
4	Clear Alignment	Clear the alignment
5	Load Job	Set bytes 1 - 64 for the null terminated file name, one file name character per 8-bit register, including the null terminator character. The ".job" extension is optional. If the extension is missing, it is automatically appended to the file name.

Value	Name	Description
6	Set Runtime Variables	The runtime variables are expected to be sent in the Runtime Variables module. The runtime variables are not included as part of the Control module.
7	Software trigger	Software trigger the sensor to capture one frame. The sensor must already be running, in trigger mode “Software”. Otherwise, software trigger has no effect.

Runtime Variables Module

The length of the Runtime Variables module is 16 bytes. The client sends the variables to the sensor in big endian format.

Byte Index	Name	Data Type	Description
0-3	Runtime Variable 0	32s	Stores the intended value of the Runtime Variable at index 0.
4-7	Runtime Variable 1	32s	Stores the intended value of the Runtime Variable at index 1.
8-11	Runtime Variable 2	32s	Stores the intended value of the Runtime Variable at index 2.
12-15	Runtime Variable 3	32s	Stores the intended value of the Runtime Variable at index 3.

State Module

The length of the State module is 116 bytes. The sensor sends the module to the client. The runtime variables are received from the sensor in big endian format. The extra unused space is for future expansion.

Byte Index	Name	Data Type	Description
0	Sensor state		0= stopped, 1 = running
1	Command in progress		1 when the sensor is busy performing the last command, 0 when done. Bytes 2, 19->83 below are only valid when there is no command in progress
2	Alignment State		0 - not calibrated, 1 calibrated (valid when byte 1 = 0)
3-10	Encoder Position	64s	Encoder position
11-18	Time	64s	Timestamp
19	Current Job filename length	8u	Number of characters in the current job filename. (eg. 11 for “current.job”) (valid when byte 1 = 0)
20-83	Current job filename		Name of currently loaded job, including extension. Each byte contains a single character. Max 64 bytes.

Byte Index	Name	Data Type	Description
			(valid when byte 1 = 0)
84-87	Runtime Variable 0	32s	Runtime variable value at index 0
…	…		
96-99	Runtime Variable 3	32s	Runtime variable value at index 3

Stamp Module

The length of the Stamp module is 45 bytes. The sensor sends the module to the client. The extra unused space is for future expansion.

Byte Index	Name	Data Type	Description
0-1	Inputs	16u	Digital input state of the last frame.
2-9	zPosition	64u	Encoder position at time of last index pulse of the last frame.
10-13	Exposure	32u	Laser exposure in µs of the last frame.
14-17	Temperature	32u	Sensor temperature in degrees celsius * 100 (centidegrees) of the last frame.
18-25	Encoder Position	64u	Encoder position of the last frame when the image data was scanned/taken.
26-33	Time	64u	Time stamp in microseconds of the last frame.
34-41	Frame Count	64u	The frame number of the last frame.

Measurements Module

The length of the Measurement module is 800 bytes. The sensor sends the module to the client. The measurements and decisions are sent in big endian format only. Each measurement plus decision takes 5 bytes so this module can hold a maximum of 800/5 = 160 measurements + decisions.

Byte Index	Name	Data Type	Description
0-3	Measurement 0	32s	measurement value (0x80000000 if invalid)
4	Decision 0	8u	Measurement decision is a bit mask where: Bit 0: 1 – Pass, 0 – Fail Bits [1-7]: 0 – Measurement value OK 1 – Invalid Value 2 – Invalid Anchor

Byte Index	Name	Data Type	Description
5-8	Measurement 1		
9	Decision 1		
…	…		
795-798	Measurement 159		
799	Decision 159		

The byte mapping of each measurement/decision pair depends on its ID as specified in the measurement interface. Each measurement will begin at byte (0 + 5*ID). For example, a measurement with ID set to 4 can be read from bytes 20 (high byte) to 23 (low byte) and the decision at 24.

ASCII Protocol

This section describes the ASCII protocol.

The ASCII protocol is available over either serial output or Ethernet output. Over serial output, communication is asynchronous (measurement results are automatically sent on the Data channel when the sensor is in the running state and results become available). Over Ethernet, communication can be asynchronous or can use polling. For more information on polling commands, see *Polling Operation Commands (Ethernet Only)* on the next page.

The protocol communicates using ASCII strings. The output result format from the sensor is user-configurable.

To use the ASCII protocol, it must be enabled and configured in the active job.

The Gocator 4.x/5.x firmware uses mm, mm^2, mm^3, and degrees as standard units. In all protocols, values are scaled by 1000, as values in the protocols are represented as integers. This results in effective units of mm/1000, mm^2/1000, mm^3/1000, and deg/1000 in the protocols.

For information on configuring the protocol with the Web interface (when using the protocol over Ethernet), see *Ethernet Output* on page 512.

For information on configuring the protocol with the Web interface (when using the protocol over Serial), see *Serial Output* on page 522.

Connection Settings

Ethernet Communication

With Ethernet ASCII output, you can set the connection port numbers of the three channels used for communication (Control, Data, and Health):

Ethernet Ports for ASCII

Name	Description	Default Port
Control	To send commands to control the sensor.	8190
Data	To retrieve measurement output.	8190
Health	To retrieve specific health indicator values.	8190

Channels can share the same port or operate on individual ports. The following port numbers are reserved for sensor internal use: 2016, 2017, 2018, and 2019. Each port can accept multiple connections, up to a total of 16 connections for all ports.

Serial Communication

Over serial, ASCII communication uses the following connection settings:

Serial Connection Settings for ASCII

Parameter	Value
Start Bits	1
Stop Bits	1
Parity	None
Data Bits	8
Baud Rate (b/s)	115200
Format	ASCII
Delimiter	CR

Up to 16 users can connect to the sensor for ASCII interfacing at a time. Any additional connections will remove the oldest connected user.

Polling Operation Commands (Ethernet Only)

On the Ethernet output, the Data channel can operate asynchronously or by polling.

Under asynchronous operation, measurement results are automatically sent on the Data channel when the sensor is in the running state and results become available. The result is sent on all connected data channels.

Under polling operation, a client can:

- Switch to a different job.
- Align, run, and trigger sensors.
- Receive sensor states, health indicators, stamps, and measurement results

A sensor sends Control, Data, and Health messages over separate channels. The Control channel is used for commands such as starting and stopping the sensor, loading jobs, and performing alignment (see *Command Channel* on the next page).

The Data channel is used to receive and poll for measurement results. When the sensor receives a Result command, it will send the latest measurement results on the same data channel that the request is received on. See *Data Channel* on page 789 for more information.

The Health channel is used to receive health indicators (see *Health Channel* on page 791).

Command and Reply Format

Commands are sent from the client to the sensor. Command strings are not case sensitive. The command format is:

<COMMAND><DELIMITER><PARAMETER><TERMINATION>

If a command has more than one parameter, each parameter is separated by the delimiter. Similarly, the reply has the following format:

<STATUS><DELIMITER><OPTIONAL RESULTS><DELIMITER>

The status can either be "OK" or "ERROR". The optional results can be relevant data for the command if successful, or a text based error message if the operation failed. If there is more than one data item, each item is separated by the delimiter.

The delimiter and termination characters are configured in the Special Character settings.

Special Characters

The ASCII Protocol has three special characters.

Special Characters

Special Character	Explanation
Delimiter	Separates input arguments in commands and replies, or data items in results. Default value is ",".
Terminator	Terminates both commands and result output. Default value is "%r%n".
Invalid	Represents invalid measurement results. Default value is "INVALID"

The values of the special characters are defined in the Special Character settings. In addition to normal ASCII characters, the special characters can also contain the following format values.

Format values for Special Characters

Format Value	Explanation
%t	Tab
%n	New line
%r	Carriage return
%%	Percentage (%) symbol

Command Channel

The following sections list the actions available on the command channel.

Optional parameters are shown in italic. The placeholder for data is surrounded by brackets (<>). In the examples, the delimiter is set to ','.

Start

The Start command starts the sensor system (causes it to enter the Running state). This command is only valid when the system is in the Ready state. If a start target is specified, the sensor starts at the target time or encoder (depending on the trigger mode).

Formats

Message	Format
Command	Start,start target The start target (optional) is the time or encoder position at which the sensor will be started. The time and encoder target value should be set by adding a delay to the time or encoder position returned by the Stamp command. The delay should be set such that it covers the command response time of the Start command.
Reply	OK or ERROR, <Error Message>

Examples:

```
Command: Start
Reply: OK
Command: Start,1000000
Reply: OK
Command: Start
Reply: ERROR, Could not start the sensor
```

Stop

The stop command stops the sensor system (causes it to enter the Ready state). This command is valid when the system is in the Ready or Running state.

Formats

Message	Format
Command	Stop
Reply	OK or ERROR, <Error Message>

Examples:

```
Command: Stop
Reply: OK
```

Trigger

The Trigger command triggers a single frame capture. This command is only valid if the sensor is configured in the Software trigger mode and the sensor is in the Running state.

Formats

Message	Format
Command	Trigger
Reply	OK or ERROR, <Error Message>

Examples:

```
Command: Trigger
Reply: OK
```

LoadJob

The Load Job command switches the active sensor configuration.

Formats

Message	Format
Command	LoadJob,job file name If the job file name is not specified, the command returns the current job name. An error message is generated if no job is loaded. ".job" is appended if the filename does not have an extension.
Reply	OK or ERROR, <Error Message>

Examples:

```
Command: LoadJob,test.job
Reply: OK,test.job loaded successfully
Command: LoadJob
Reply: OK,test.job
Command: LoadJob,wrongname.job
Reply: ERROR, failed to load wrongname.job
```

Stamp

The Stamp command retrieves the current time, encoder, and/or the last frame count.

Formats

Message	Format
Command	Stamp,time,encoder,frame If no parameters are given, time, encoder, and frame will be returned. There could be more than one selection.
Reply	If no arguments are specified: OK, time, <time value>, encoder, <encoder position>, frame, <frame count> ERROR, <Error Message> If arguments are specified, only the selected stamps will be returned.

Examples:

```
Command: Stamp
Reply: OK,Time,9226989840,Encoder,0,Frame,6
Command: Stamp,frame
Reply: OK,6
```

Clear Alignment

The Clear Alignment command clears the alignment record generated by the alignment process.

Formats

Message	Format
Command	ClearAlignment
Reply	OK or ERROR, <Error Message>

Examples:

```
Command: ClearAlignment
Reply: OK
```

Stationary Alignment

The Stationary Alignment command performs an alignment based on the settings in the sensor's live job file. A reply to the command is sent when the alignment has completed or failed. The command is timed out if there has been no progress after one minute.

Formats

Message	Format
Command	StationaryAlignment
Reply	If no arguments are specified OK or ERROR, <Error Message>

Examples:

```
Command: StationaryAlignment
Reply: OK
Command: StationaryAlignment
Reply: ERROR,ALIGNMENT FAILED
```

Set Runtime Variables

The Set Runtime Variables command sets the runtime variables, using the specified index, length, and data. Values are integers.

Formats

Message	Format
Command	setvars,index,length,data Where *data* is the delimited integer values to be set.
Reply	OK or ERROR

Examples:

```
Command: setvars,0,4,1,2,3,4
Reply: OK
```

Get Runtime Variables

The Get Runtime Variables command gets the runtime variables, using the specified index and length.

Formats

Message	Format
Command	setvars,index,length
Reply	OK,data Where *data* is the delimited data for the passed length.

Examples:

```
Command: getvars,0,4
Reply: OK,1,2,3,4
```

Data Channel

The following sections list the actions available on the data channel.

Optional parameters are shown in italic. The placeholder for data is surrounded by brackets (<>). In the examples, the delimiter is set to ','.

Result

The Result command retrieves measurement values and decisions.

Formats

Message	Format
Command	Result,measurement ID,measurement ID...
Reply	If no arguments are specified, the custom format data string is used. OK, <custom data string> ERROR, <Error Message> If arguments are specified, OK, <data string in standard format> ERROR, <Error Message>

Examples:

Standard data string for measurements ID 0 and 1:

```
Result,0,1
```

```
OK,M00,00,V151290,D0,M01,01,V18520,D0
```

Standard formatted measurement data with a non-existent measurement of ID 2:

```
Result,2
```

```
ERROR,Specified measurement ID not found. Please verify your input
```

Custom formatted data string (%time, %value[0], %decision[0]):

```
Result

OK,1420266101,151290,0
```

Value

The Value command retrieves measurement values.

Formats

Message	Format
Command	Value,measurement ID,measurement ID...
Reply	If no arguments are specified, the custom format data string is used. OK, <custom data string> ERROR, <Error Message> If arguments are specified, OK, <data string in standard format, except that the decisions are not sent> ERROR, <Error Message>

Examples:

Standard data string for measurements ID 0 and 1:

```
Value,0,1

OK,M00,00,V151290,M01,01,V18520
```

Standard formatted measurement data with a non-existent measurement of ID 2:

```
Value,2

ERROR,Specified measurement ID not found. Please verify your input
```

Custom formatted data string (%time, %value[0]):

```
Value

OK, 1420266101, 151290
```

Decision

The Decision command retrieves measurement decisions.

Formats

Message	Format
Command	Decision,measurement ID,measurement ID...
Reply	If no arguments are specified, the custom format data string is used. OK, <custom data string> ERROR, <Error Message> If arguments are specified,

Message	Format
	OK, <data string in standard format, except that the values are not sent> ERROR, <Error Message>

Examples:

Standard data string for measurements ID 0 and 1:

```
Decision,0,1
```

```
OK,M00,00,D0,M01,01,D0
```

Standard formatted measurement data with a non-existent measurement of ID 2:

```
Decision,2
```

```
ERROR,Specified measurement ID not found. Please verify your input
```

Custom formatted data string (%time, %decision[0]):

```
Decision
```

```
OK,1420266101, 0
```

Health Channel

The following sections list the actions available on the health channel.

Optional parameters are shown in italic. The placeholder for data is surrounded by brackets (<>). In the examples, the delimiter is set to ','.

Health

The Health command retrieves health indicators. See *Health Results* on page 708 for details on health indicators.

Formats

Message	Format
Command	Health,health indicator ID.Optional health indicator instance ... More than one health indicator can be specified. Note that the health indicator instance is optionally attached to the indicator ID with a '.'. If the health indicator instance field is used the delimiter cannot be set to '.'.
Reply	OK, <health indicator of first ID>, <health indicator of second ID> ERROR, <Error Message>

Examples:

```
health,2002,2017
```

```
OK,46,1674
```

```
Health
```

```
ERROR,Insufficient parameters.
```

Standard Result Format

A sensor can send measurement results either in the standard format or in a custom format. In the standard format, you select in the web interface which measurement values and decisions to send. For each measurement the following message is transmitted:

M	t_n	,	i_n	,	V	v_n	,	D	d_1	CR

Field	Shorthand	Length	Description
MeasurementStart	M	1	Start of measurement frame.
Type	t_n	n	Hexadecimal value that identifies the type of measurement. The measurement type is the same as defined elsewhere (see *Data Results* on page 694).
Id	i_n	n	Decimal value that represents the unique identifier of the measurement.
ValueStart	V	1	Start of measurement value.
Value	v_n	n	Measurement value, in decimal. The unit of the value is measurement-specific.
DecisionStart	D	1	Start of measurement decision.
Decision	d_1	1	Measurement decision, a bit mask where: Bit 0: 1 – Pass 0 – Fail Bits 1-7: 0 – Measurement value OK 1 – Invalid value 2 - Invalid anchor

Custom Result Format

In the custom format, you enter a format string with place holders to create a custom message. The default format string is "%time, %value[0], %decision[0]".

Result Placeholders

Format Value	Name	Explanation
%time	Time	Timestamp in microseconds of the last frame.
%encoder	Encoder Position	Encoder position of the last frame when the image data was scanned/taken.
%frame	Frame Index	Frame number of the last frame.

Format Value	Name	Explanation
%value[Measurement ID]	Value	Measurement value of the specified measurement ID. The ID must correspond to an existing measurement. The value output will be displayed as an integer in micrometers.
%decision [Measurement ID]	Decision	Measurement decision, where the selected measurement ID must correspond to an existing measurement. Measurement decision is a bit mask where: Bit 0: 1 – Pass 0 – Fail Bits 1-7: 0 – Measurement value OK 1 – Invalid value 2 - Invalid anchor

C language *printf*-style formatting is also supported: for example, %sprintf[%09d, %value[0]]. This allows fixed length formatting for easier input parsing in PLC and robot controller logic.

GenICam GenTL Driver

GenICam is an industry standard for controlling and acquiring data from an imaging device. Gocator sensors support GenICam through a GenTL Producer driver.

The included GenTL driver allows GenICam-compliant third-party software applications such as Halcon and Common Vision Blox to acquire and process 3D data and intensity generated from the sensor.

The following sensor scan modes are supported:

- Video
- Profile (with **Uniform Spacing** disabled). In this mode, the raw profiles are resampled and accumulated into a surface.
- Surface (with **Uniform Spacing** enabled)

For more information on scan modes and uniform spacing, see *Scan Modes* on page 88.

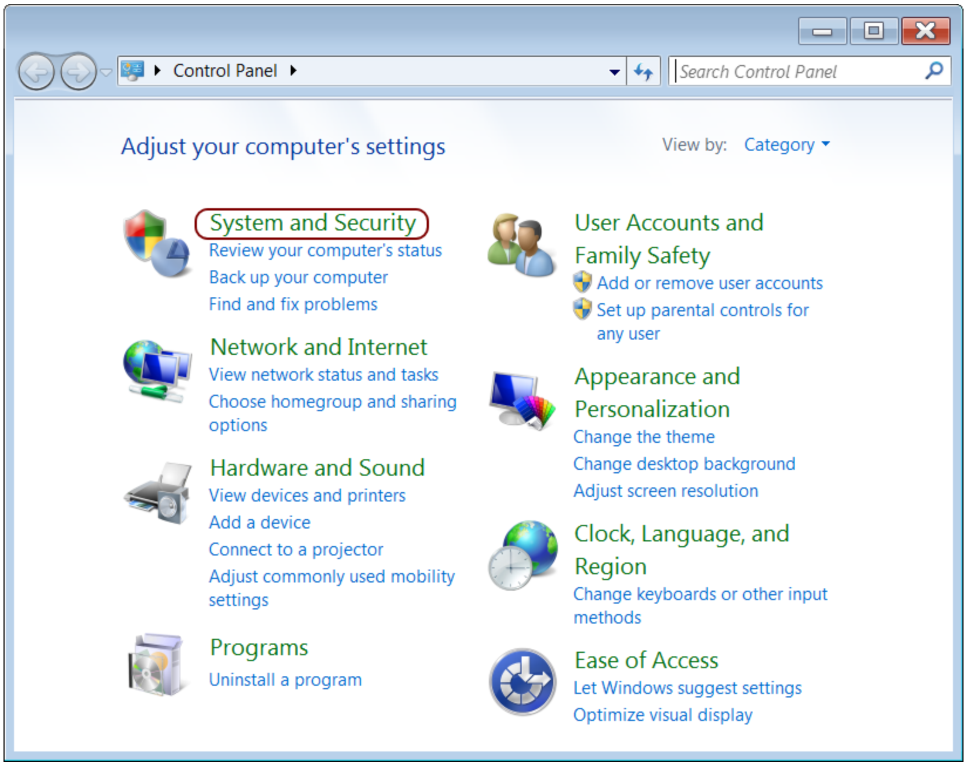
	To use these third-party software applications, you *must* configure a system variable so the software can access the GenTL driver. For instructions, see *To configure system variables to use the driver in Windows 7*, below.

To get the utilities package containing the driver (14405-x.x.x.x_SOFTWARE_GO_Utilities.zip), go to https://downloads.lmi3d.com/, choose your product from the Product Downloads section, and download it from the Download Center.

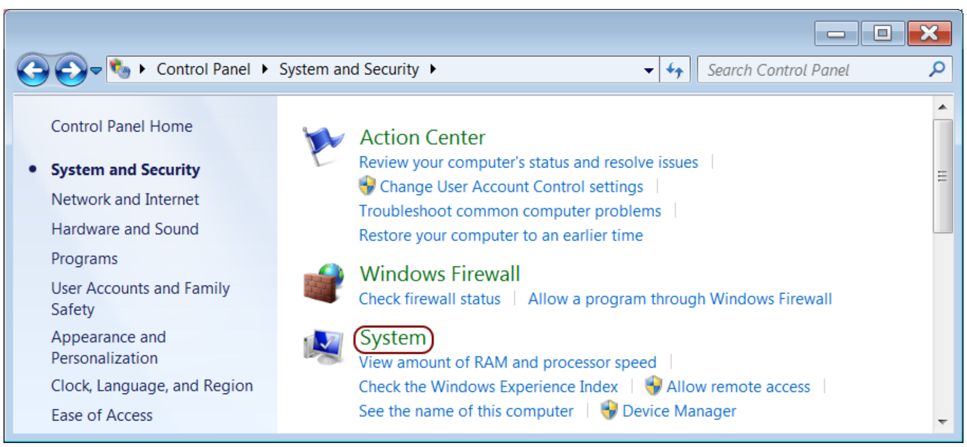
After downloading the package and unzipping the file to a location you will remember, you will find the driver in the GenTL\x86 or GenTL\x64 subfolder under Integration > GenTL (you can move the GenTL folder to a more convenient location).

To configure system variables to use the driver in Windows 7:

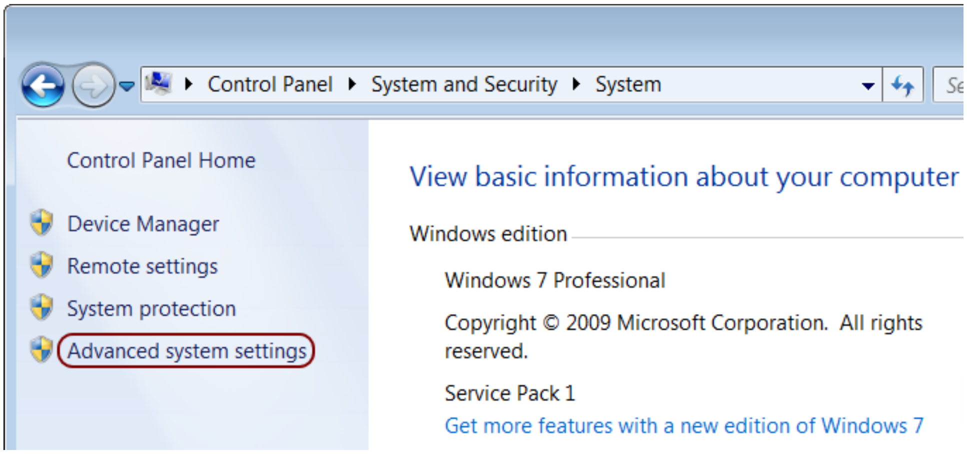
1. From the Start menu, open the **Control** panel and then click **System and Security**.



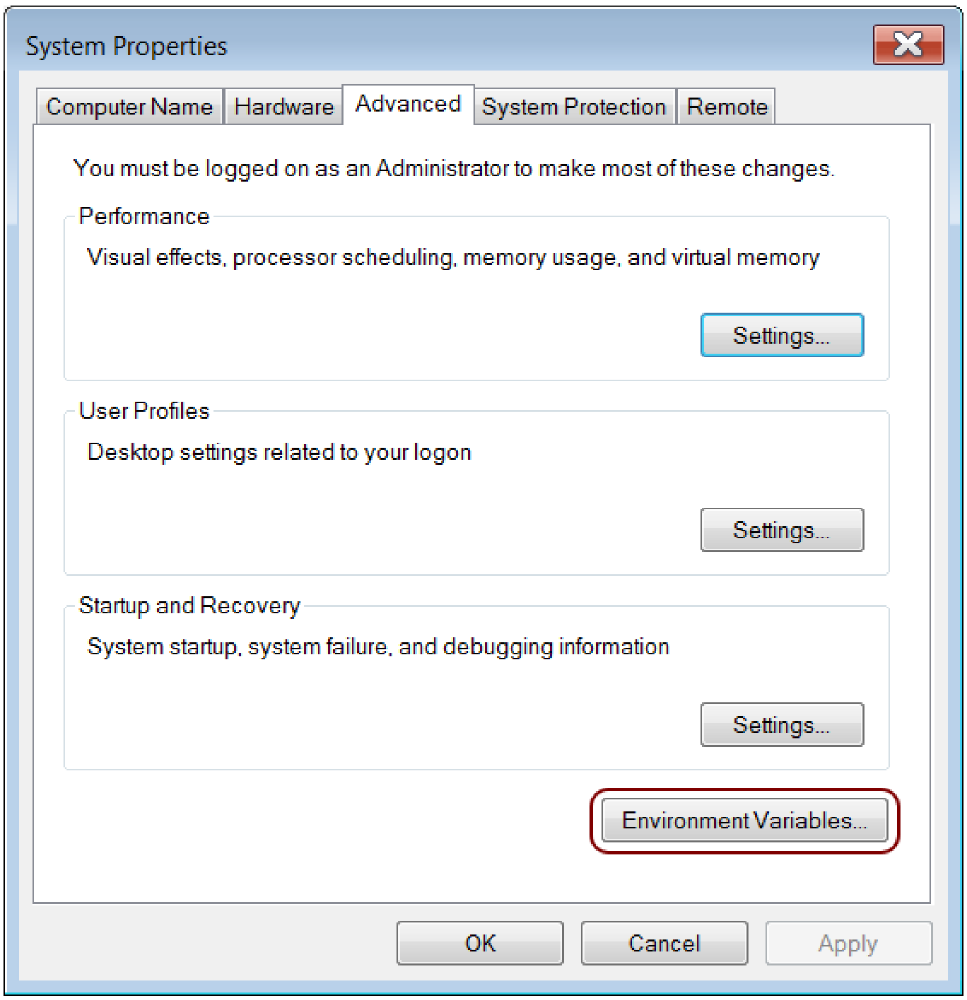
2. Click **System**.



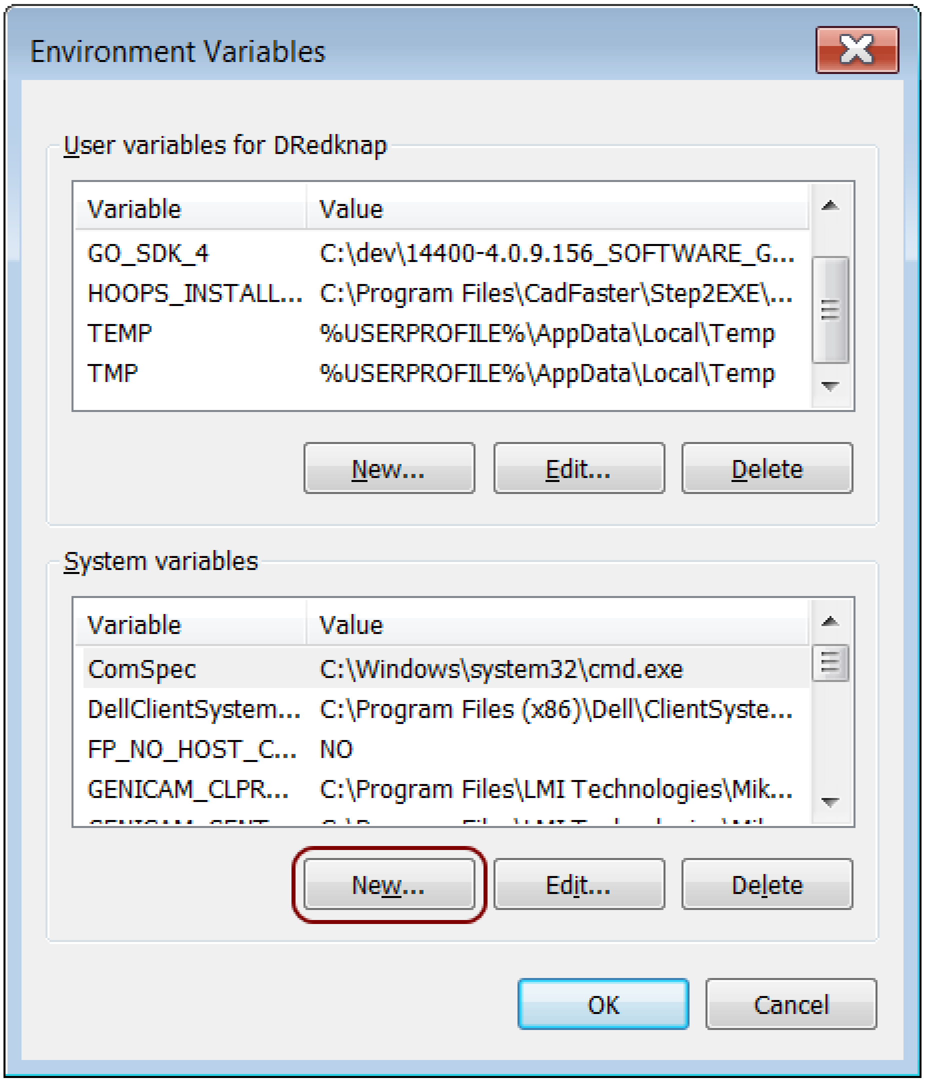
3. Click **Advanced System Settings**.



4. In the **System Properties** dialog, on the **Advanced** tab, click **Environment Variables...**

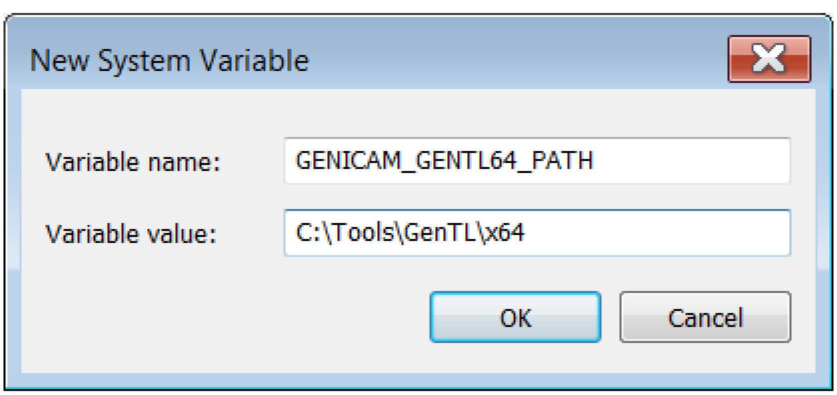


5. In the **Environment Variables** dialog, under the **System variables** list, click **New**.



6. In the **New System Variable** dialog, enter the following information, depending on your system:

	Variable name	Variable value
32-bit system	GENICAM_GENTL32_PATH	The full path to the GenTL\x86 folder.
64-bit system	GENICAM_GENTL64_PATH	The full path to the GenTL\x64 folder.



7. Click OK in the dialogs until they are all closed.

To work with the GenTL driver, the sensor must operate with the appropriate output enabled in the **Ethernet** panel in the **Output** page. Check **Acquire Intensity** in the **Scan Mode** panel on the **Scan** page and enable intensity output in the **Ethernet** panel if intensity data is required.

The GenTL driver packs the output, intensity, and stamps (e.g., time stamp, encoder index, etc.) into either a 16-bit RGB image or a 16-bit grey scale image. You can select the format in the Go2GenTL.xml setting file.

The width and height of the 16-bit RGB or grey scale image is calculated from the maximum number of columns and rows needed to accommodate the sensor's field of view and the maximum part length.

16-bit RGB Image

When the 16-bit RGB format is used, the height map, intensity, and stamps are stored in the red, green, and blue channel respectively.

Channel	Details
Red	Height map information. The width and height of the image represent the dimensions in the X and Y axis. Together with the pixel value, each red pixel presents a 3D point in the real-world coordinates. The following formula can be used to calculate the real-world coordinates (X, Y, Z) from pixel coordinates (Px, Py, Pz): X = X offset + Px * X resolution Y = Y offset + Py * Y resolution Z = Z offset + Pz * Z resolution Refer to the blue channel on how to retrieve the offset and resolution values. If Pz is 0 if the data is invalid. The Z offset is fixed to -32768 * Z resolution. Z is zero if Pz is 32768.
Green	Intensity information. Same as the red channel, the width and height of the image represent the dimension in the X and the Y axis. Together with the pixel value, each blue pixel represents an intensity value in the real-world coordinates. The following formula can be used to calculate the real-world coordinates (X, Y, Z) from pixel coordinates (Px, Py, Pz): X = X offset + Px * X resolution Y = Y offset + Py * Y resolution Z = 16-bit intensity value The intensity value is 0 if the intensity image is not available. Gocator outputs 8-bit intensity values. The values stored in the 16-bit RGB image is multiplied by 256. To obtain the original values, divide the intensity values by 256. Refer to the blue channel on how to retrieve the offset and resolution values.
Blue	Stamp information. Stamps are 64-bit auxiliary information related to the height map and intensity content. The next table explains how the stamps are packed into the blue pixel channel

Channel	Details
	See *Data Results* on page 694 for an explanation of the stamp information.

The following table shows how the stamp information is packed into the blue channel. A stamp is a 64-bit value packed into four consecutive 16-bit blue pixels, with the first byte position storing the most significant byte.

Stamp Information from GenTL driver

Stamp Index	Blue Pixel Position	Details
0	0..3	Version
1	4..7	Frame Count
2	8..11	Timestamp (µs)
3	12..15	Encoder value (ticks)
4	16..19	Encoder index (ticks) This is the encoder value when the last index is triggered
5	20..23	Digital input states
6	24..27	X offset (nm)
7	28..31	X resolution(nm)
8	32..35	Y offset (nm)
9	36..39	Y resolution (nm)
10	40..43	Z offset (nm)
11	44..47	Z resolution (nm)
12	48..51	Height map Width (in pixels)
13	52..55	Height map length (in pixels)
14	56..59	Specify if the intensity is enabled

16-bit Grey Scale Image

When the 16-bit grey scale format is used, the height map, intensity, and stamps are stored sequentially in the grey scale image.

The last row of the image contains the stamp information.

Rows	Details
0 .. (max part height - 1)	Height map information. The width and height of the image represent the dimensions in the X and Y axis. Together with the pixel value, each pixel presents a 3D point in the real-world coordinates. The following formula can be used to calculate the real-world coordinates (X, Y, Z) from pixel coordinates (Px, Py, Pz): X = X offset + Px * X resolution Y = Y offset + Py * Y resolution

Rows	Details
	Z = Z offset + Pz * Z resolution Refer to the blue channel on how to retrieve the offset and resolution values. If Pz is 0 if the data is invalid. The Z offset is fixed to -32768 * Z Resolution. Z is zero if Pz is 32768.
(max part height) .. 2* (max part height) If intensity is enabled	Intensity information. The width and height of the image represent the dimension in the X and the Y axis. Together with the pixel value, each blue pixel represents an intensity value in the real-world coordinates. The following formula can be used to calculate the real-world coordinates (X, Y, Z) from pixel coordinates (Px, Py, Pz): The following formula assumes Py is relative to the first row of the intensity information, not the first row of the whole 16-bit grey scale image. X = X offset + Px * X resolution Y = Y offset + Py * Y resolution Z = 16-bit intensity value This intensity value is 0 if the intensity image is not available. Gocator outputs 8-bit intensity values. The values stored in the 16-bit Grey scale image is multiplied by 256. To obtain the original values, divide the intensity values by 256. Refer to the stamps on how to retrieve the offset and resolution values.
The last row of the 16-bit grey scale image	Stamp information. Stamps are 64-bit auxiliary information related to the height map and intensity content. The next table explains how the stamps are packed into the blue pixel channel See *Data Results* on page 694 for an explanation of the stamp information.

The following table shows how the stamp information is packed into the last row. A stamp is a 64-bit value packed into four consecutive 16-bit pixels, with the first byte position storing the most significant byte.

Stamp Information from GenTL driver

Stamp Index	Column Position	Details
0	0..3	Version
1	4..7	Frame Count
2	8..11	Timestamp (µs)
3	12..15	Encoder value (ticks)
4	16..19	Encoder index (ticks) This is the encoder value when the last index is triggered
5	20..23	Digital input states
6	24..27	X offset (nm)
7	28..31	X resolution(nm)
8	32..35	Y offset (nm)

Stamp Index	Column Position	Details
9	36..39	Y resolution (nm)
10	40..43	Z offset (nm)
11	44..47	Z resolution (nm)
12	48..51	Height map Width (in pixels)
13	52..55	Height map length (in pixels)
14	56..59	Specify if intensity is enabled or not

Registers

GenTL registers are multiples of 32 bits. The registers are used to control the operation of the GenTL driver, send commands to the sensors, or to report the current sensor information.

Register Map Overview

Register Address	Name	Read/Write	Length (bytes)	Description
260	WidthReg	RO	4	Specify the width of the returned images. The part height map is truncated if it is wider than the specified width.
264	HeightReg	RO	4	Specify the height of the returned images (i.e., length of the part). The part height map is truncated if it is longer than the specified length.
292	ResampleMode	RO	4	Enable the resampling logic in the GenTL driver 0 – Disable resampling 1 – Enable resampling When resampling is enabled, the GenTL driver will resample the height map so that the pixel spacing is the same in the X and Y axis.
296	EncoderValue0	RO	4	Report the current encoder value (least significant 32-bit). The current encoder value is latched from the sensor when this register is read.
300	EncoderValue1	RO	4	Report the current encoder value (most significant 32-bit). The encoder value is latched when EncoderValue0 register is read. User should read EncoderValue0 before reading EncoderValue1.
304	Configuration File	RW	16	Read the name of sensor live configuration file or switch (write) the sensor configuration file. The configuration name is NULL terminated and includes the extension ".job". Writing to this register causes the sensor to switch to the

Register Address	Name	Read/Write	Length (bytes)	Description
				specified configuration.
320	Transformation X offset	RO	4	Return the sensor transformation X offset
324	Transformation Z offset	RO	4	Return the sensor transformation Z offset
328	Transformation Angle	RO	4	Return the sensor transformation angle
332	Transformation Orientation	RO	4	Return the sensor transformation orientation
336	Clearance distance	RO	4	Return the sensor clearance distance

XML Settings File

The settings file, Go2GenTL.xml, resides in the same directory as the Gocator GenTL driver. Users can set the resample mode and output format by changing the setting in this file.

Element	Type	Description
ResampleMode	32u	Settings to disable or enable resampling mode: 0 – Disable 1 – Enable When resampling mode is enabled, the GenTL driver will resample the height map so that the pixel spacing is the same in the X and Y axis. The default value is 1.
DataFormat	32u	Settings to choose 16-bit RGB or 16-bit grey scale image output: 0 – 16-bit RGB Image 1 – 16-bit grey scale Image The default value is 0.

Interfacing with Halcon

Halcon is a comprehensive software package for machine vision applications with an integrated development environment. A sensor can use the included GenTL driver to stream 3D point clouds and intensity data into Halcon in real-time.

> The current GenTL driver does not support scanning in profile mode.

For information on setting up the GenTL driver, see *GenICam GenTL Driver* on page 794.

This section describes how to configure Halcon to acquire data from the 4.x firmware. You should be familiar with the sensor's Surface mode. Before continuing, make sure Halcon is installed.

Requirements

Firmware	Firmware 4.0.9.136 or later
Halcon	Version 10.0 or later

Setting Up Halcon

Before using Halcon with a sensor, you must set up Halcon.

To set up Halcon:

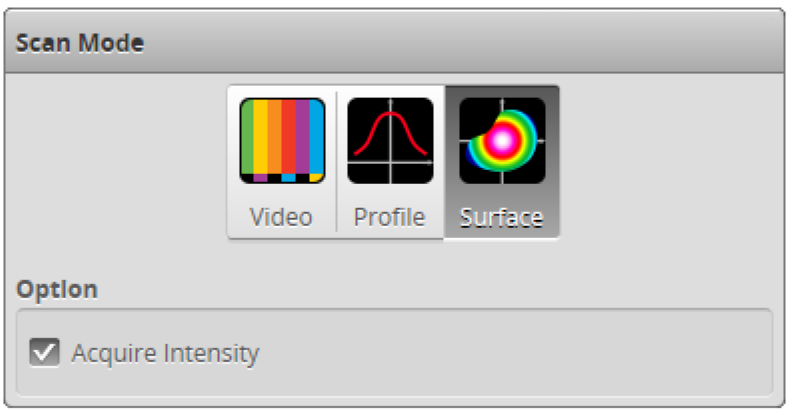
1. Connect a sensor to the PC running Halcon.

   You will need a Master hub to connect the sensor to the PC. For more information, see *Installation* on page 25 and *Network Setup* on page 35.

2. Click the **Scan** page icon.

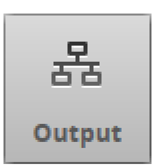
3. On the **Scan** page, click the **Surface** icon to switch to Surface mode.

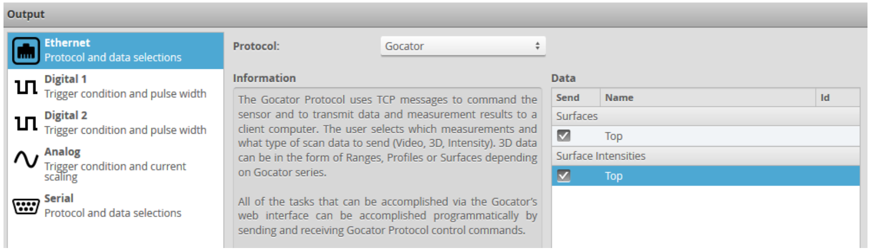
4. (Optional) If you need intensity data, check the **Acquire Intensity** option.
5. Configure the sensor to produce the desired surface data.

   For more information on configuring sensors, see *Scan Setup and Alignment* on page 87 and *Models* on page 126.

6. Click the **Output** page icon.

7. On the **Output** page, enable the required surface under **Data** and choose Gocator in **Protocol**.



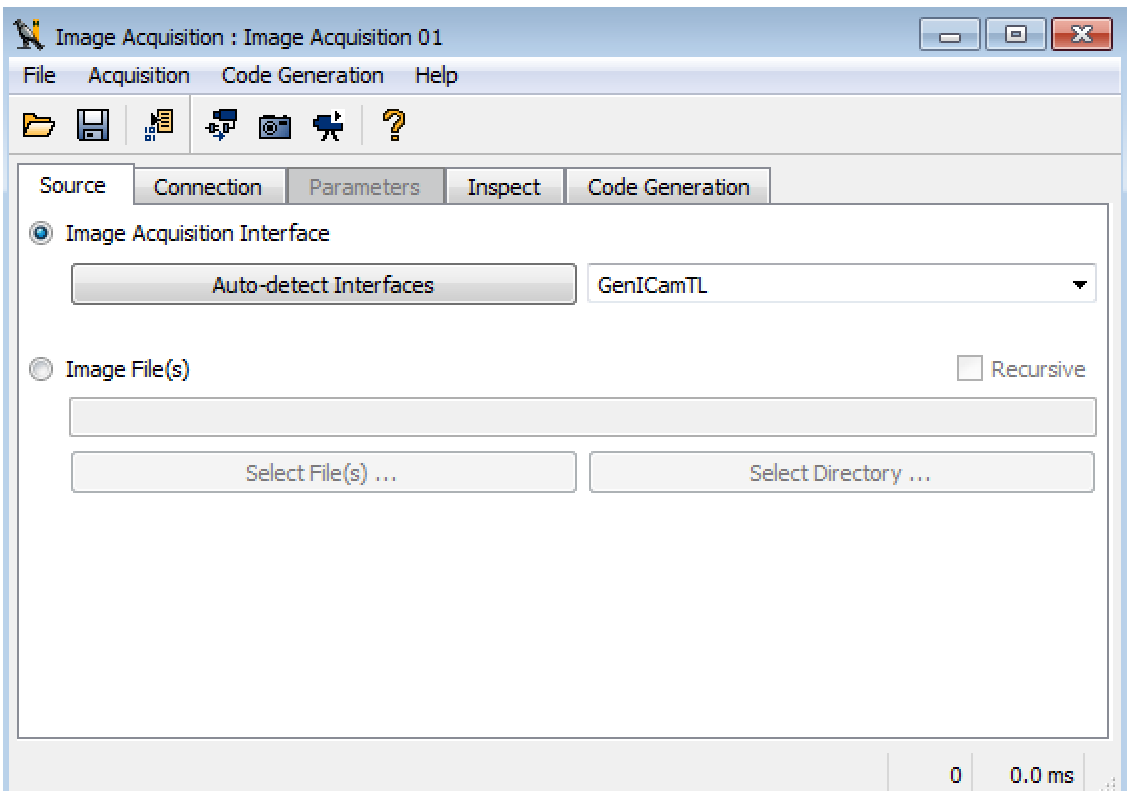
For more information on configuring Ethernet output, see *Ethernet Output* on page 512.

8. Make sure the sensor is running.

9. On the PC, launch Halcon.

10. In Halcon, in the **Assistants** menu, click **Open New Image Acquisition**.

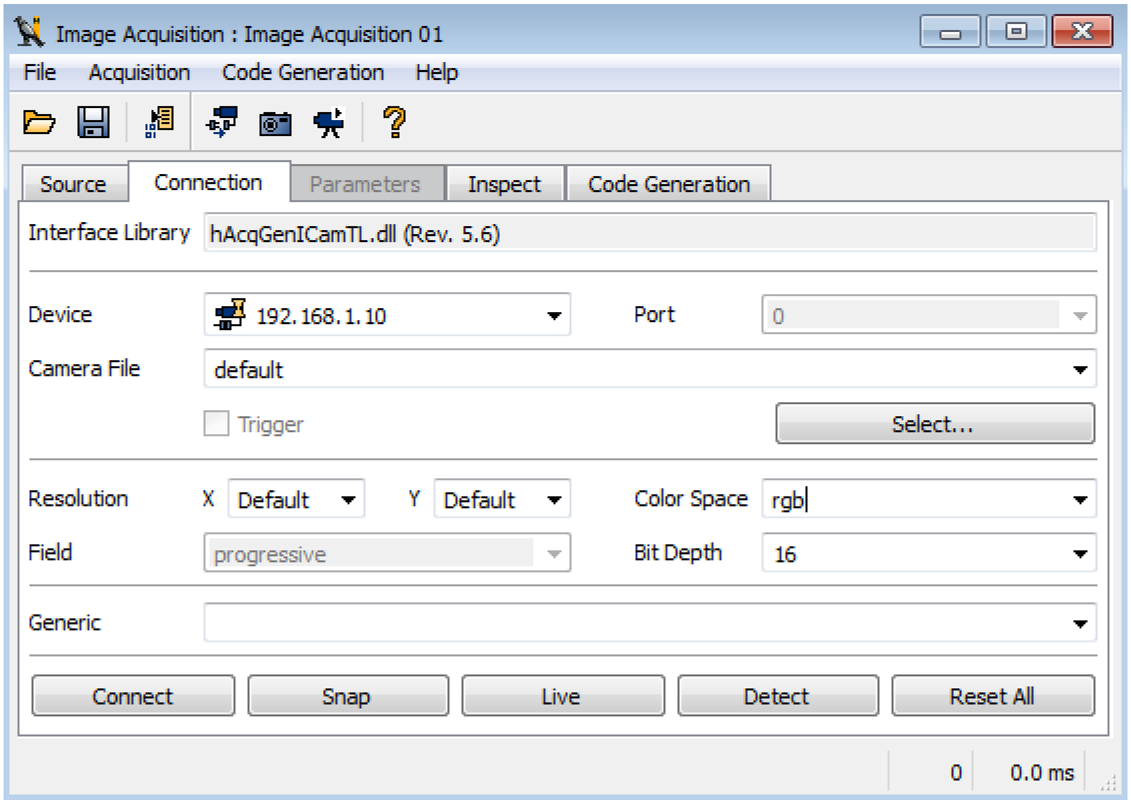
11. In the dialog that opens, in the **Source** tab, check the **Image Acquisition Interface** option and choose `GenICamTL` in the drop-down.



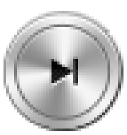
The driver uses the Gocator protocol discovery messages to search for available Gocator sensors. Discovery messages can be blocked by a PC's firewall. You should therefore turn off the firewall and try again if the sensor can't be detected.

12. Switch to the **Connection** tab.

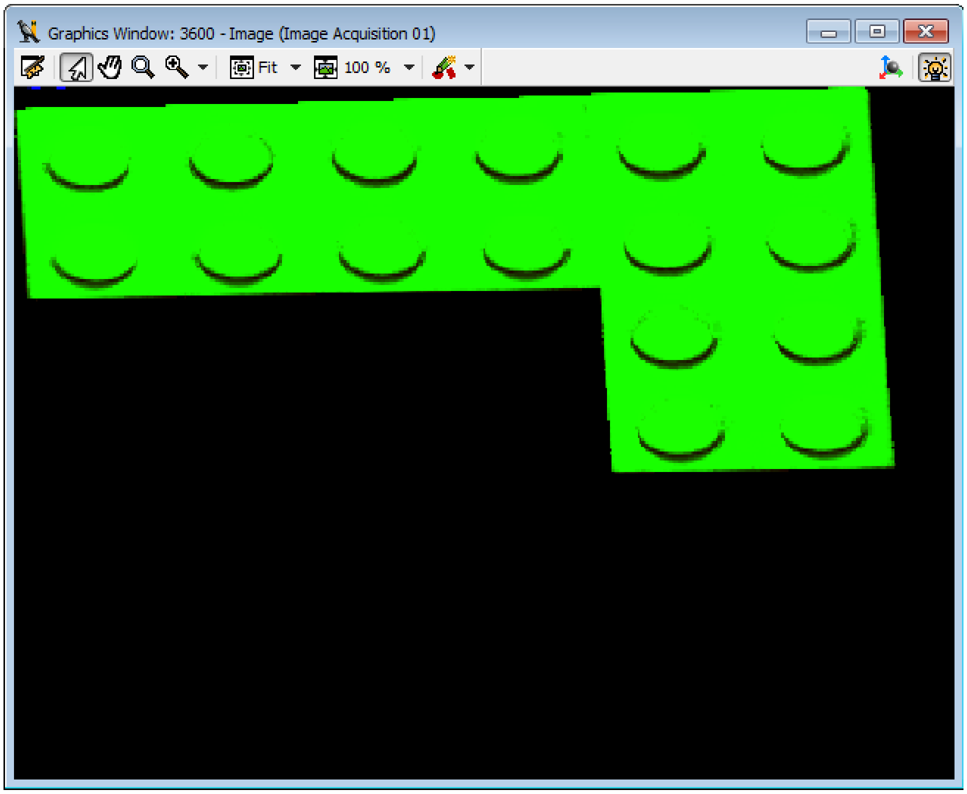
If Halcon detects a sensor, the sensor's IP will be listed next to **Device**.



13. In the **Connection** tab, set **Color Space** to `RGB` and **Bit Depth** to `16`.
14. In the sensor's web interface, click the Snapshot button to trigger the output of a surface.



The output displays in the Halcon **Graphics Window**.



Halcon is now configured for use with the sensor.

Halcon Procedures

The Halcon example code contains internal procedures that you can use to decompose the RGB image and to control registers that the GenTL driver opens.

You can import the procedures into your own code by selecting File > Insert Program > Insert Procedures and then choosing the example code `Continuous_Acq.hdev` under the `Examples/Halcon` directory.

	The Go2GenTL.xml file for Gocator 4.x has more fields than the Gocator 3.x version. Make sure you are using the correct version.

The following section describes each of these procedures.

Halcon Procedures

Procedures	Description
Go2GenTL_ ParseData	The GenTL driver packs the height map, intensity and stamp information into a 16-bit RGB image. The function is used to extract data from the RGB image. For details on how the information is packed in the data, see the sections under *GenICam GenTL Driver* on page 794. The function accepts the image acquired from `grab_image_async`, and returns the height map, intensity and stamps. *Parameters (Input)* **Image**: RGB Image acquired by using `grab_image_async`.

Procedures	Description
	Parameters (Output) **HeightMap**: The height map image. **Intensity**: The intensity image. **FrameCount**: The number of frames. **Timestamp**: The timestamp. **Encoder**: The encoder position. **EncoderIndex**: The last index of the encoder. **Inputs**: The digital input states. **xOffset**: The X offset in millimeters. **xResolution**: The X resolution in millimeters. **yOffset**: The Y offset in millimeters. **yResolution**: The Y resolition in millimeters. **zOffset**: The Z offset in millimeters. **zResolution**: The Z resolution in millimeters. **Width**: The width (number of columns) of the image that contains the part. The part width can be less than the image width requested by the user. **Height**: The height or length (number of rows) of the image that contains the part. The part length can be less than the image height requested by the user. **HasIntensity**: Specifies if the intensity image is available. The intensity image is available if **Acquire Intensity** is enabled in the sensor's web interface. Each output is returned as decimal value. *Example* `Go2GenTL_ParseData(Image, HeightMap, Intensity, frameCount, timestamp, encoderPosition, encoderIndex, inputs, xOffset, xResolution, yOffset, yResolution, zOffset, zResolution, width, height, hasIntensity)`
Go2GenTL_ ResampleMode	Returns the resample mode. *Parameters (Input)* **AcqHandle**: Acquisition handle created by `open_framegrabber`. *Parameters (Output)* **ResampleMode**: No - Resample is disabled. Yes - Resample is enabled. When resampling is enabled, the GenTL driver resamples the height map so that the pixel spacing is the same on the X and Y axis. *Example* `Go2GenTL_ResampleMode (AcqHandle, ResampleMode)` To set the resample mode, you must directly modify Go2GenTL.xml, which is in the same directory as the sensor GenTL driver (Go2GenTL.cti).
Go2GenTL_ ConfigFileName	Returns the current live sensor job file name. *Parameters (Input)* **AcqHandle**: Acquisition handle created by `open_framegrabber`.

Procedures	Description
	Parameters (Output) **ConfigFile**: The name of the job file. The file name includes the extension .job. *Example* `Go2GenTL_ConfigFileName (AcqHandle, ConfigFile)`
Go2GenTL_ SetConfigFileName	Sets the sensor live configuration. *Parameters (Input)* **AcqHandle**: Acquisition handle created by `open_framegrabber`. **ConfigFile**: The name of the job file. The file name should include the extension .job. *Example* `Go2GenTL_SetConfigFileName (AcqHandle, 'test2.cfg')`
Go2GenTL_ Encoder	Returns the current encoder value. When this function is called, the GenTL driver retrieves the latest encoder value from the sensor. The value is returned as a two-element tuple. The first element is the least significant 32-bit value, and the second element is the most significant 32-bit value. *Parameters (Input)* **AcqHandle**: Acquisition handle created by `open_framegrabber`. *Parameters (Output)* **EncoderValue**: The current encoder value. *Example* `Go2GenTL_Encoder(AcqHandle, EncoderValue)`
Go2GenTL_ ImageSize	Returns the size of the image returned by the GenTL driver. *Parameters (Input)* **AcqHandle**: Acquisition handle created by `open_framegrabber`. *Parameters (Output)* **Width**: The width of the image. **Height**: The height of the image. *Example* `Go2GenTL_ImageSize(AcqHandle, Width, Height)` To set the image size, you must directly modify Go2GenTL.xml, which is in the same directory as the sensor GenTL driver (Go2GenTL.cti).
Go2GenTL_ CoordinateXYZ	Returns the real-world coordinates (X, Y, Z) of the part given the row and column position in the height map. The values of the offset and resolution input parameters can be retrieved using Go2GenTL_ ParseData. *Parameters (Input)* **HeightMap**: The height map image. **Row**: The row in the height map. **Column**: The column in the height map. **xOffset**: The X offset in millimeters.

Procedures	Description
	xResolution: The X resolution in millimeters. **yOffset**: The Y offset in millimeters. **yResoluion**: The Y resolution in millimeters. **zOffset**: The Z offset in millimeters. **zResolution**: The Z resolution in millimeters. *Parameters (Output)* **coordinateXYZ**: The real-world coordinates.
Go2GenTL_ Exposure	Returns the current exposure. *Parameters (Input)* **AcqHandle**: Acquisition handle created by `open_framegrabber`. *Parameters (Output)* **Exposure**: The current exposure value (in µs). The value is returned as an integer. Decimals are truncated. *Example* `Go2GenTL_Exposure(AcqHandle, exposure)`
Go2GenTL_ SetExposure	Sets the current exposure. *Parameters (Input)* **AcqHandle**: Acquisition handle created by `open_framegrabber`. **Exposure**: The current exposure value (in µs), as an integer. *Example* `Go2GenTL_SetExposure(AcqHandle, exposure)`

set_ framegrabber_ param	Generic Halcon function to set parameters on the scanner. Can be used to set scanner specific settings. For a complete list of settings that can be changed, see the SDK interface files. In the generic form:
	
	`set_framegrabber_param( AcqHandle, ‘Name’, ‘Value’)`
	
	Parameters (Input)
	AcqHandle: Acquisition handle created by `open_framegrabber`.
	Name: The name of the parameter to set on the scanner.
	Value: The parameter value to set on the scanner.
	
	Examples
	To set the format of the image buffer to 16-bit packed:
	`set_framegrabber_param( AcqHandle, ‘PixelFormat’, ‘RGB16Packed’)`
	
	To set the Scan mode to HDR (1 = no HDR, 2 = HDR, 3 = Super HDR):
	`set_framegrabber_param( AcqHandle, ‘Dynamic’, ‘2’)`
	
	To set the brightness to ‘3’:
	`set_framegrabber_param( AcqHandle, ‘Exposure’, ‘3’)`
	
	To schedule a system to start in 1000000 ticks or microseconds (depends on current domain unit):
	`set_framegrabber_param( AcqHandle, ‘XMLSetting’, ‘GenTL/System’)`

Procedures	Description

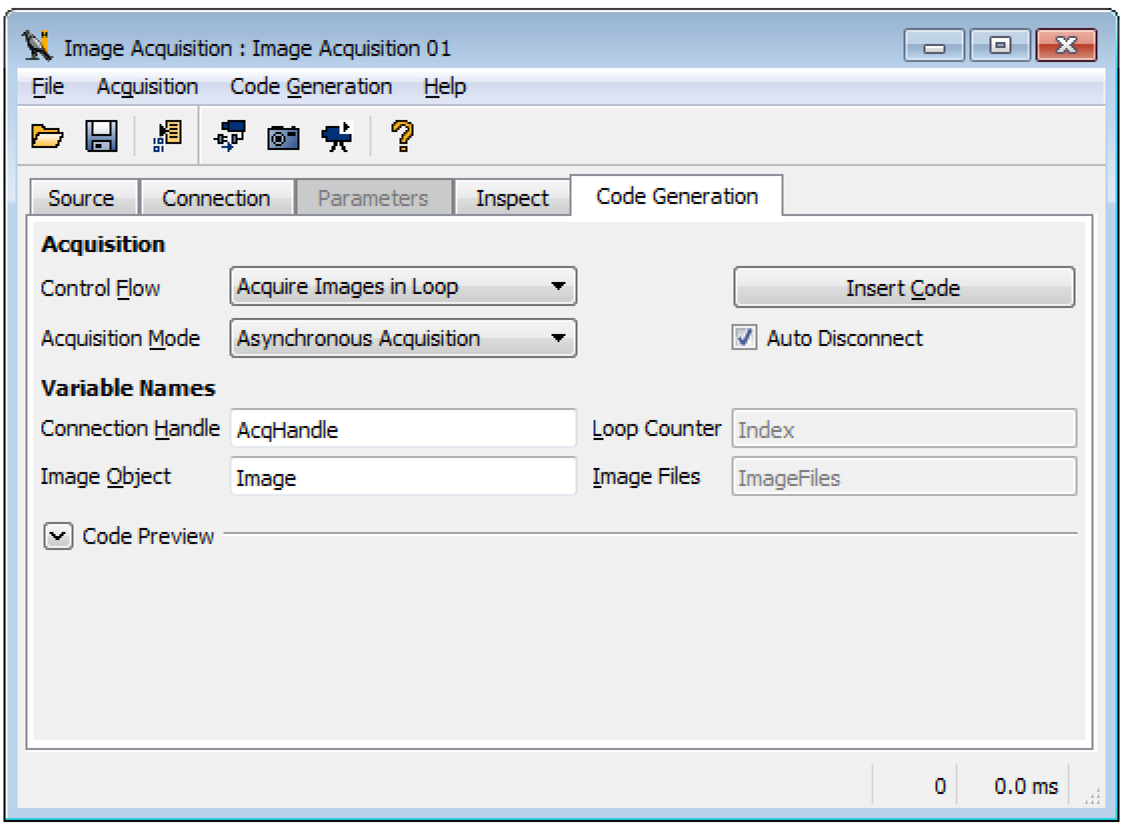
	`set_framegrabber_param( AcqHandle, ‘XMLSetting’, ‘ScheduledStart=1’)`
	`set_framegrabber_param( AcqHandle, ‘XMLSetting’, ‘000000’)`
	`set_framegrabber_param( AcqHandle, ‘XMLSetting’, ‘’)`
	
	To schedule a *sensor* to start after a delay (ticks or microseconds), pass `GenTL/Sensor` in the first call to `set_framegrabber_param`, followed by the remaining calls to the function as described in the previous example:
	`set_framegrabber_param( AcqHandle, ‘XMLSetting’, ‘GenTL/Sensor’)`
	
	To clear data buffers::
	`set_framegrabber_param(AcqHandle,'XmlCommand','GenTL/ClearData\n')`

Generating Halcon Acquisition Code

Halcon lets you insert acquisition code into your code in the IDE.

To generate acquisition code:

1. In Halcon, in the **Assistants** menu, click **Open New Image Acquisition**.
2. In the dialog that opens, in the **Code Generation** tab, set **Acquisition Mode** to **Asynchronous Acquisition**.



3. Under **Acquisition**, click **Insert Code** to generate the code that will open the acquisition device.

	To handle cases when the `grab_image` function times out while waiting for data, add a `try-catch` statement around the `grab_image` function code.

After the example code is generated, you should add a `catch` instruction to bypass the acquisition timeout event, and use the Go2GenTL_ParseData function to extract information from the returned image.

An example, `Continuous_Acq.hdev`, is included in the `Examples/Halcon` directory.

MountainsMap Transfer Tool

The MountainsMap transfer tool (MMTransfer.exe) lets you trigger scans on a connected sensor. The scan data is then automatically transferred to the MountainsMap component of the transfer tool. You can then work on the scan data within the tool. For more information on the tool, see *Using the Mountains Map Transfer Tool* on the next page.

	MountainsMap must be installed and properly licensed on the PC.

The MountainsMap transfer tool is available in the utilities package (14405-x.x.x.x_SOFTWARE_GO_Utilities.zip). To get the package, go to https://downloads.lmi3d.com/, choose your product from the Product Downloads section, and download it from the Download Center.

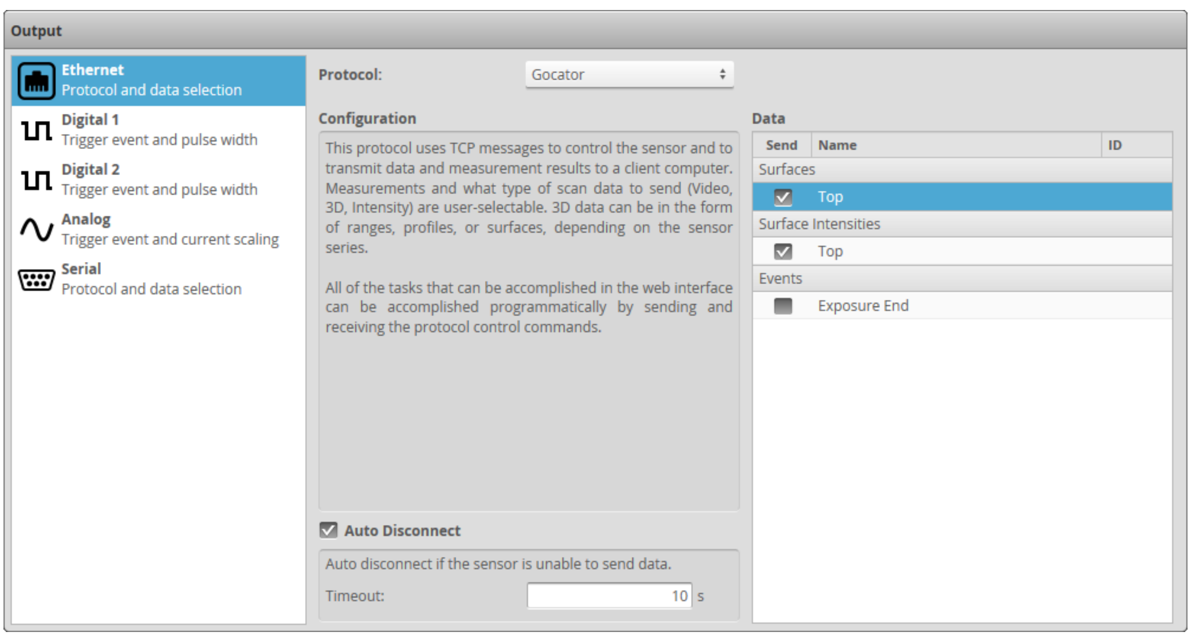
Configuring a Sensor to Work with the Transfer Tool

In order for scan data to be available for transfer, you must first configure the sensor.

To configure a sensor:

1. In the web interface, go to the **Output** tab.
2. In the **Ethernet** category, set **Protocol** to **Gocator**.
3. In the **Data** area of the panel, make sure a source for Surface data is checked under **Surfaces**.

   By default, a source for Surface data is already selected.

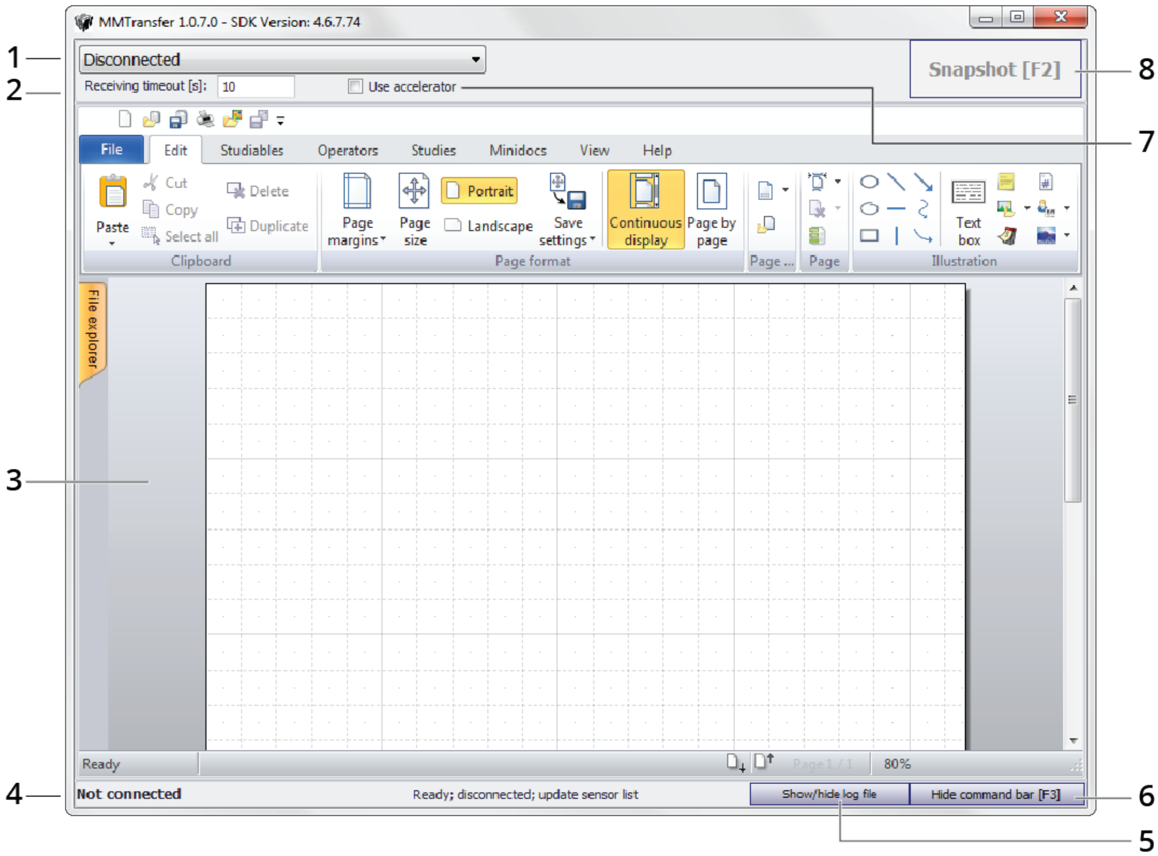


4. (Optional) If you want to transfer intensity data, check a source for intensity data under **Surface Intensities**.

For intensity to be available, you must also check **Acquire Intensity** on the **Scan** page.

Using the Mountains Map Transfer Tool

The following graphic and table show the functionalities available in the tool:



	Element	Description
1	Sensor selector	Lets you choose among connected sensors.
2	Receiving timeout	The number of seconds the transfer tool will wait to receive data from the sensor before timing out.
3	MountainsMap component	After the data transfers from the sensor to the tool, you can edit it directly in the transfer tool.
4	Status bar	Indicates whether the tool is connected to a sensor, and so on.
5	Show/hide log file	Toggles display of the log file. Useful for diagnosing connection and scan issues.
6	Hide command bar	Toggles display of the command bar at the top of the tool.

	Element	Description
7	Use accelerator	Attempts to accelerate the sensor chosen in the sensor selector drop-down. In order for the tool to accelerate the sensor, the sensor's firmware must match the Gocator SDK version indicated on the tool's title bar. If these versions do not match, the tool will not be able to accelerate the sensor. In this case, you can accelerate the sensor using the standalone accelerator application or an SDK-based application.
8	Snapshot button	Causes the connected sensor to take a snapshot. The data is then transferred to the tool. If the tool does not receive the data before the delay specified in **Receiving timeout**, the transfer fails.

To use the transfer tool:

1. Make sure the sensor you wish to work with is configured properly.

   See *Configuring a Sensor to Work with the Transfer Tool* on page 811.

2. (Optional) Modify the timeout or check the **Use accelerator** option.

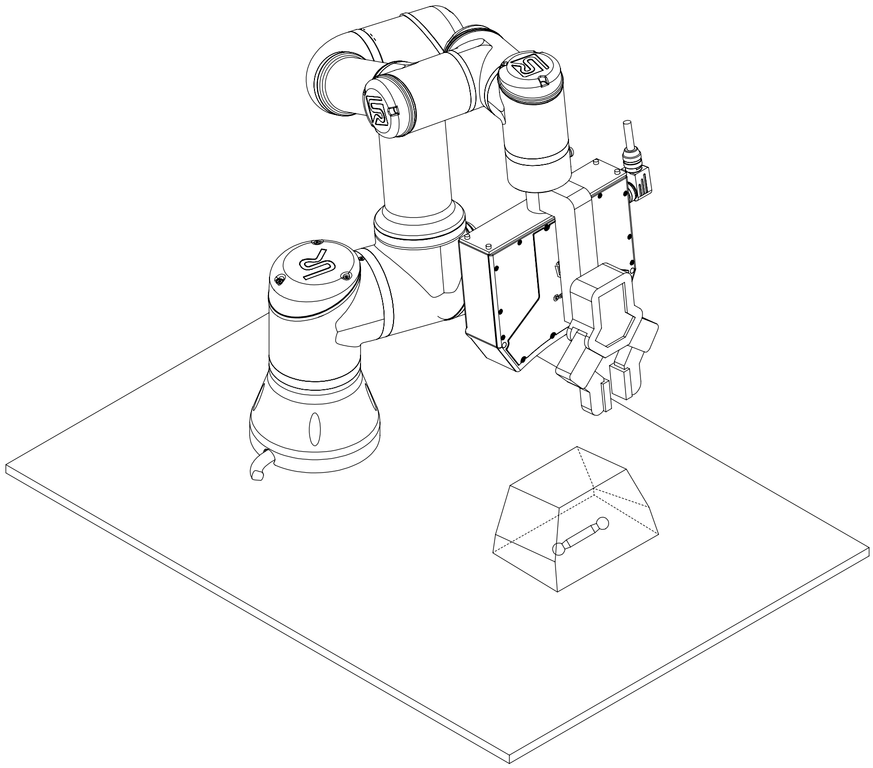
   See the table above for more information.

3. Click the Snapshot button or press F2.

   The sensor takes a snapshot, and the scan data transfers to the tool.

Universal Robots Integration

This section describes using the Gocator URCap with a sensor mounted on a Universal Robots robot arm to perform hand-eye calibration.



The Gocator URCap is an application that you install on the robot. The URCap triggers scans on the sensor and retrieves the information that describes the pose of the calibration target in the sensor's field of view. After a sufficient number of scans, the calibration is saved to the robot, and hand-eye calibration (between the sensor and the robot flange) is complete.

Note that the URCap currently only supports end-of-arm mounting. That is, the sensor is mounted to the end of the robot arm, rather than being mounted in a fixed position above the target.

Mounting the Sensor

Typically, you will mount the sensor to the robot flange using a metal plate. G3 Universal Robots kits for Gocator 3210 and Gocator 3506 are available for purchase from LMI; these kits include brackets. Otherwise, you will need to design your own mount. The design of the plate depends on how you will mount the sensor to the robot, whether and how you need to mount a manipulator, and finally which sensor model you are mounting. For sensor mounting hole locations and specifications, see *Sensors* on page 880.

The sensor must be mounted such that its positive Y axis is on the same side as the robot flange's positive Y axis, parallel to the robot flange's Y axis. Consult the robot's documentation to determine the robot flange's positive Y if necessary. To determine the positive Y axis of your sensor, see the coordinate system orientation for your sensor in the appropriate sensor model section in *Sensors* on page 880.

At a minimum, you will need to connect the Power & Ethernet cordset to the sensor's Power/LAN connector, connect the Ethernet end of the cordset to a switch, and wire the pigtailed leads to a power source; for pinout and lead information, see *Gocator Power/LAN Connector* on page 892.

If you intend to use the sensor to control an external device such as a PLC based on the sensor's measurement decisions, you will also need to connect an I/O cordset to the sensor's I/O connector and configure I/O. For pinout and lead information, see *Gocator I/O Connector* on page 894. For information on configuring I/O on the sensor, see *Digital Output* on page 516.

Make sure that cordsets:

- do not interfere with robot movement;
- do not block the sensor's view of targets;
- do not interfere with the robot's manipulator;
- do not touch targets.

Connecting to the Sensor

You typically configure a sensor (measurements and general settings) by connecting to the sensor's web interface from a client PC over an Ethernet connection. By default, a sensor's IP is 192.168.1.10. The client PC Ethernet card you use to connect to the sensor must share the same network ID as the sensor: we typically suggest setting the client PC's Ethernet card to 192.168.1.5. However, you may need to change the sensor and client PC IPs to work with your network. For more information on client PC setup, see *Client Setup* on page 35. For more information on setting the sensor's IP, see *Networking and Power* on page 76.

> You can also use the Discovery tool (kDiscovery.exe) to set the sensor's IP without needing to change the client PC's network ID. The tool is available in the utilities package (14405-x.x.x.x_SOFTWARE_GO_Utilities.zip) on LMI's website (https://downloads.lmi3d.com/), under the *Software* category.

To connect to the sensor:

1. If you have not already done so, apply power to the sensor.

   Sensors typically take less than 30 seconds to start.

2. From a web browser connected to the switch to which the sensor is connected, type in the IP address of the sensor.

3. If an Administrator password has been set, make sure the Administrator account type is selected in the first drop-down and type the password in the field that appears.

   Only Administrator account types can add tools.

4. Click **Login** to log in to the sensor.

Installing the Gocator URCap on the Robot

To perform the hand-eye calibration between the Gocator sensor and the Universal Robots robot, you must install the Gocator URCap on the robot.

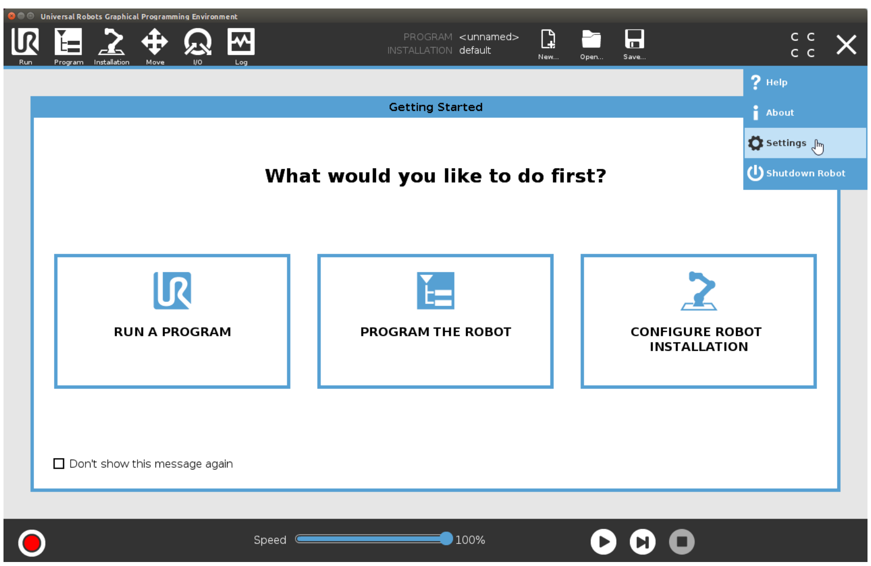
The Gocator URCap is available in the utilities package (`14405-x.x.x.x_SOFTWARE_GO_Utilities.zip`), available on LMI's website.

To access the Gocator Utilities package:

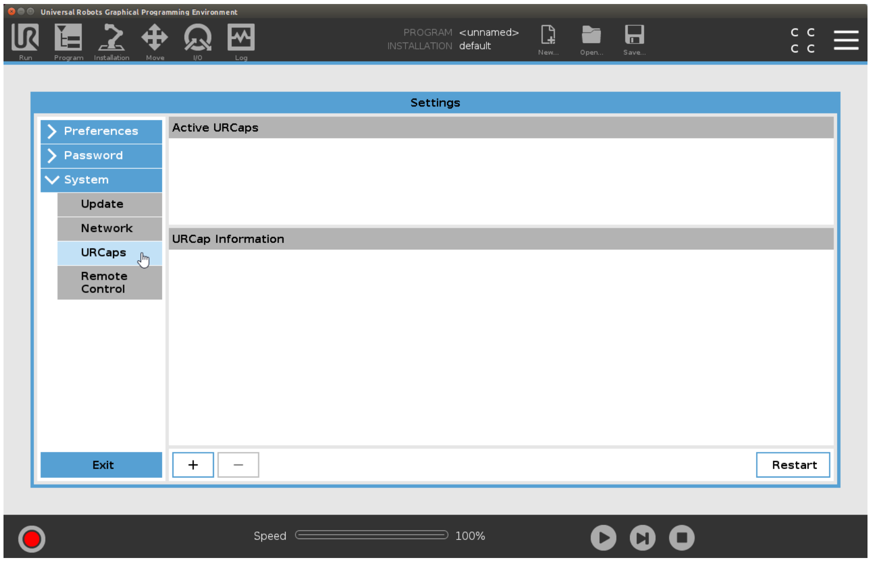
1. Go to [https://downloads.lmi3d.com/.](https://downloads.lmi3d.com/)
2. Choose *Gocator* as the brand and select your product from the product drop-down.
3. Click **Go**.
4. Expand the *Software* section and then the release section corresponding to the firmware of your sensor.
5. Click the **Download** button next to the Gocator Utilities package.

To install the URCap:

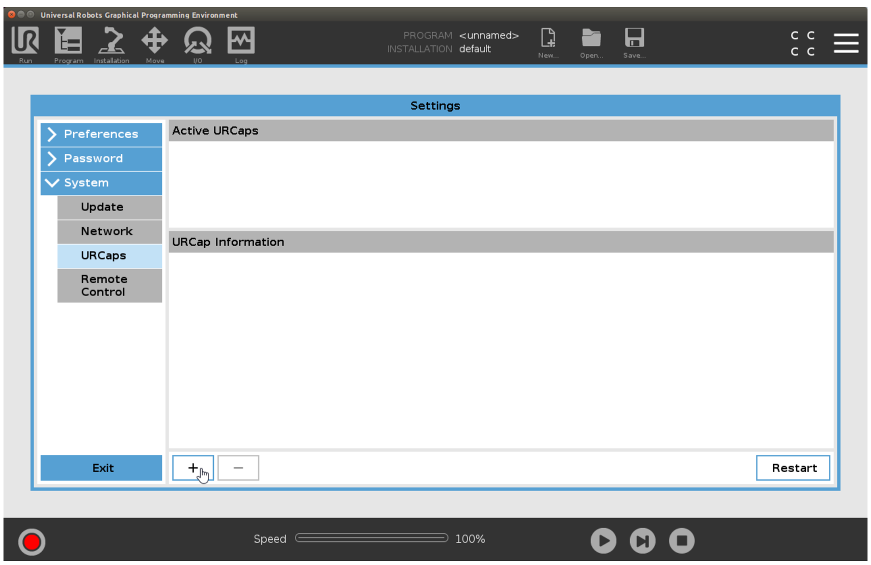
1. Unzip the contents of the zip file and copy the URCap file (a file ending with a .urcap extension) to a USB drive.
2. Insert the USB drive into the USB port on the robot's teach pendant.
3. In the Universal Robots interface, click **Settings**.



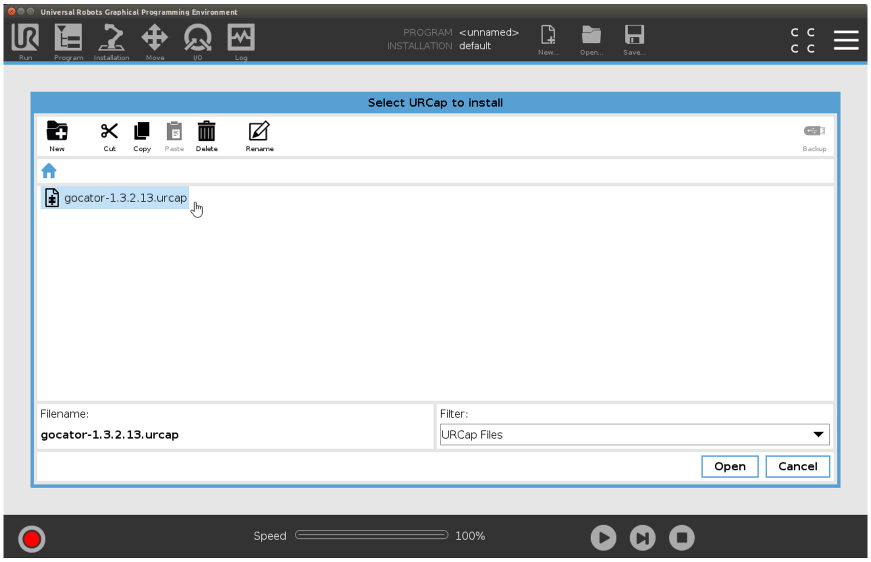
4. In the **Setup Robot** screen, click **URCaps**.



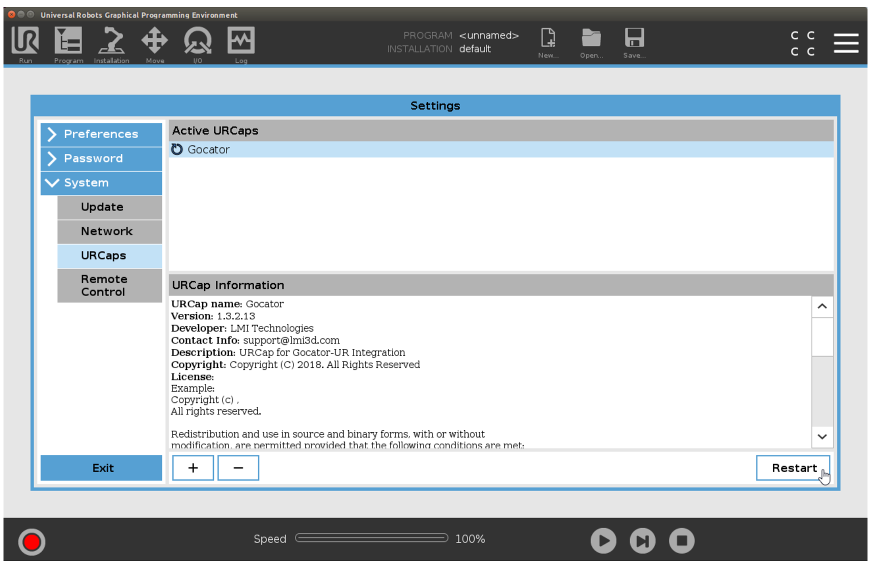
5. Click the Plus button.



6. Click the `gocator-x.x.x.x.urcap` file to select it and then click **Open**.



7. In the next screen, click **Restart** to finish the installation and restart the robot.



The Gocator URCap is now installed on the robot.

> To upgrade the Gocator URCap on the robot, simply install the new version to replace the old version.

Configuring the Sensor

To perform the hand-eye calibration, one or more measurement tools (which return the required information to the Gocator URCap on the robot) must be present and configured on the sensor.

In addition to the measurement tools, you must configure a few other settings to perform the hand-eye calibration. For more information, see *Other Sensor Settings* on page 821.

For the measurement tools you need to configure on the sensor, you have three options:

- If you are using an LMI Universal Robots kit, which provides a ball bar target, you can use the built-in Surface Ball Bar measurement tool. LMI recommends using the Gocator URCap on the UR robot to add the tool and to perform the initial configuration of the tool. After, see *Using the Surface Ball Bar Tool* below for information on completing configuration of the tool.
- Use other Gocator measurement tools and a script tool, following the procedures and information provided in *Using Other Measurement Tools* on the next page.
- Develop a custom GDK tool, following the procedures and information provided in *Using Other Measurement Tools* on the next page.

For information on adding measurement tools, see *Adding and Configuring a Measurement Tool* on page 149.

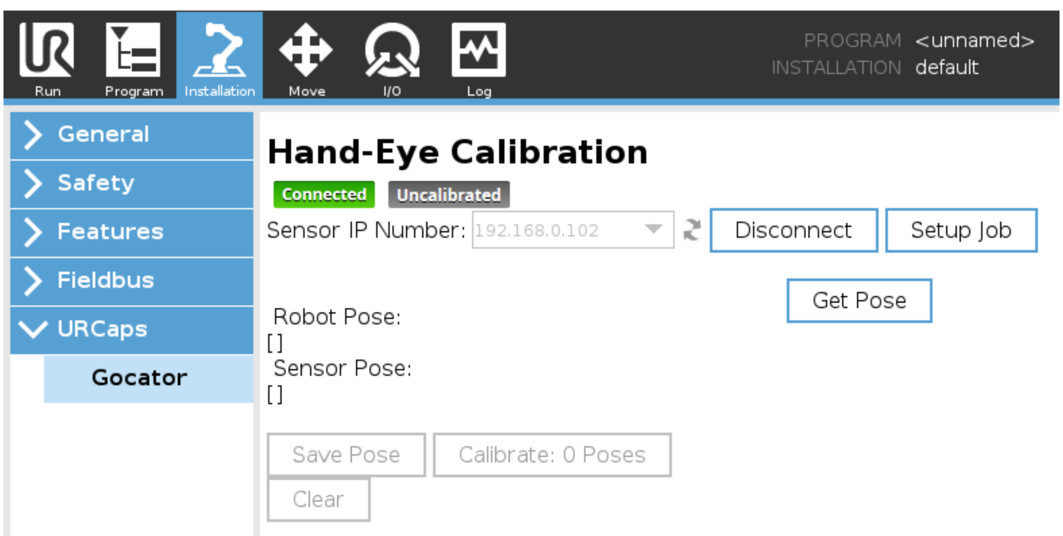
Using the Surface Ball Bar Tool

The Surface Ball Bar tool lets you quickly perform hand-eye calibration between a sensor and a Universal Robots robot using a ball-bar target. The Surface Ball Bar tool is a built-in Gocator measurement tool that detects the balls in a ball-bar and returns the measurements required by the Gocator URCap.

To set up the calibration job, do the following:

1. In the robot's interface, go to the **Installation** tab.

2. Open the **URCaps** category and click **Gocator**.



3. If the URCap is not connected to the sensor, select its IP address in the **Sensor IP Number** drop-down and click **Connect**.

> The following step removes any currently added measurement tools and creates a new job

called `GoRobotCalib` on the sensor. Make note of this job name, as you must use it in the Gocator Calibrate node; for more information, see *Gocator Calibrate* on page 828.

4. Click **Setup job**.

   The URCap adds and configures an instance of the Surface Ball Bar tool, and configures various other settings.

   In order to see the changes that the URCap makes on the sensor, you may need to refresh the browser page containing the sensor's web interface; for more information on connecting to a sensor, see *Connecting to the Sensor* on page 815

You can add this tool and configure it manually in the Gocator web interface. However, LMI recommends that you add the tool from the Gocator URCap. Doing this configures default values in the tool, makes sure that the required measurements and outputs are enabled and in the correct order (described in *Information required by the Gocator URCap* below), and configures other settings. For information on the Surface Ball Bar tool, see *Ball Bar* on page 281.

Using Other Measurement Tools

If you use a calibration target other than a ball bar, you can use a combination of the other built-in measurement tools and a Script tool to produce the measurement values required by the Gocator URCap. The choice of measurement tools depends on your calibration target. In general, you need the following:

- Two Surface measurement tools that return positional information of two features, such as the center of a hole or the tip (maximum Z) of a vertical feature; for more information, see *Surface Measurement* on page 277.
- A Surface Plane (see *Plane* on page 409) tool on the surface surrounding the calibration target to retrieve X angle and Y angle information.
- A Script tool to use the positional information of the first two Surface tools and the X angle and Y angle information of the Plane tool to calculate the values of the rotation matrix (I, J, and K vectors). (For more information on script tools, see page 483.)

The Gocator URCap requires 12 values corresponding to the values of a 3x3 rotation matrix and a 3x1 translation vector. Each value is retrieved by the URCap from the sensor via either a measurement tool (such as the Surface Ball Bar tool) or a script tool that uses values from two or more measurement tools to calculate the other values.

Whichever Gocator tools you use, the Gocator URCap expects the measurements to be in a specific order and to use ID numbers 0 to 11. Otherwise, the hand-eye calibration will not be correct.

Information required by the Gocator URCap

Type of Information	ID	Description

Ix	0	X, Y, and Z components of the I, J, and K unit vectors defining the coordinate system orientation.
Iy	1	
Iz	2	
Jx	3	
Jy	4	
Jz	5	
Kx	6	
Ky	7	
Kz	8	
Tx	9	X, Y, and Z components of the translation vector defining the coordinate system origin location.
Ty	10	
Tz	11	

The first time you add a measurement tool in the Gocator interface, its first measurement is automatically enabled, and its ID is set to 0. Each time you enable another measurement or add a new tool (which automatically enables another measurement), the ID is set to the next available number. Because the Gocator URCap expects the measurements in a specific order and with specific IDs, you will have to enable and set the IDs accordingly. For more information, see *Enabling and Disabling Measurements* on page 174 and *Changing a Measurement ID* on page 175.

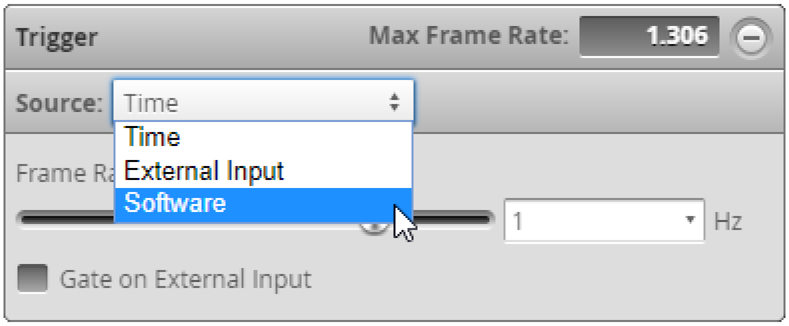
Other Sensor Settings

The following settings on the Gocator sensor are required or recommended.

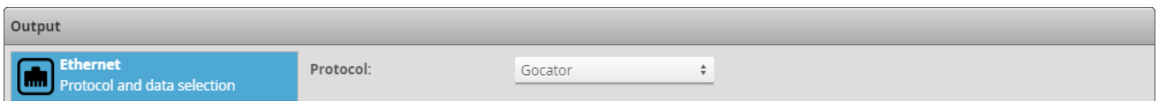
> If you add an instance of the Surface Ball Bar tool using the **Setup Job** button in the Gocator URCap plugin, the plugin automatically sets the trigger source to software and enables the required measurements and outputs.

- You do not usually need to perform the sensor alignment procedure available in the **Sensor** panel on the **Scan** page (for more information, see *Alignment* on page 106). In an unaligned sensor, the origin is in the middle of the sensor's scan volume. Note that if the alignment of the sensor changes at any time after you perform the hand-eye calibration (either by performing the alignment for the first time or by re-aligning the sensor), you will have to perform hand-eye calibration again.

- On the **Scan** page, in the **Trigger** panel, set **Source** to **Software**.

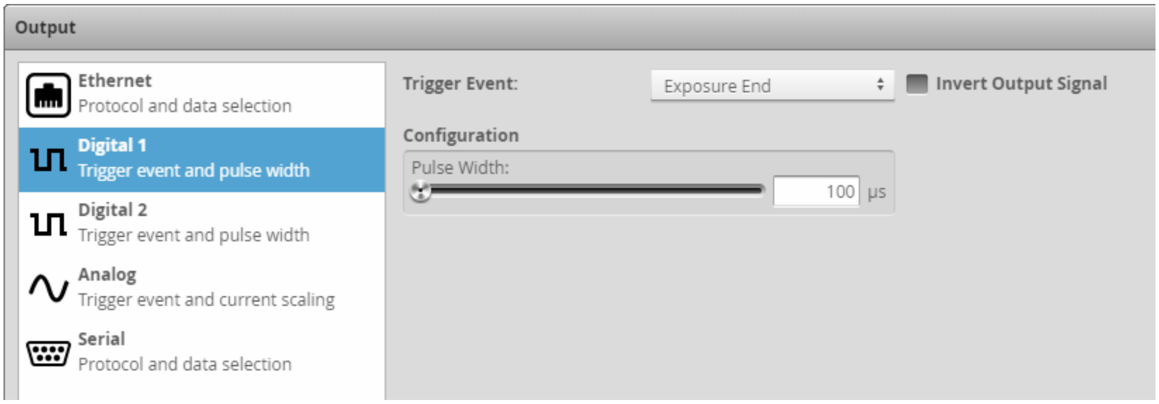


- On the **Scan** page, in the **Exposure** panel, set the exposure as required.
- On the **Output** page, set **Protocol** to **Gocator** and enable the surface data (normally labeled "Top"), as well as all 12 of the required measurements.



  Make sure that the measurements are in the correct order and that their IDs are 0 to 11. For more information, see the table entitled *Information required by the Gocator URCap* on page 820.

- Part matching is not recommended. To locate parts, consider using the Surface Segmentation tool; for information on this tool, see *Segmentation* on page 426. If you do use part matching, make sure you set **Frame of Reference** in the **Part Detection** panel to **Sensor** to keep the results in the correct coordinate system; for more information, see *Part Detection* on page 111.
- You should use separate jobs for calibration and for scanning. For more information, see *Creating, Saving and Loading Jobs (Settings)* on page 64. For information on managing jobs, see *Jobs* on page 78
- If you need to use the Exposure End event as a trigger for the robot, on the **Output** page, under Digital 1 or Digital 2, set **Trigger Event** to "Exposure End" and use this digital output as the input to the robot.



Performing the Hand-Eye Calibration

LMI recommends performing hand-eye calibration using the Gocator Calibrate program node, rather than the routine available in the robot's **Installation** tab. For more information, see *Gocator Calibrate* on page 828.

(Legacy) Performing the Hand-Eye Calibration

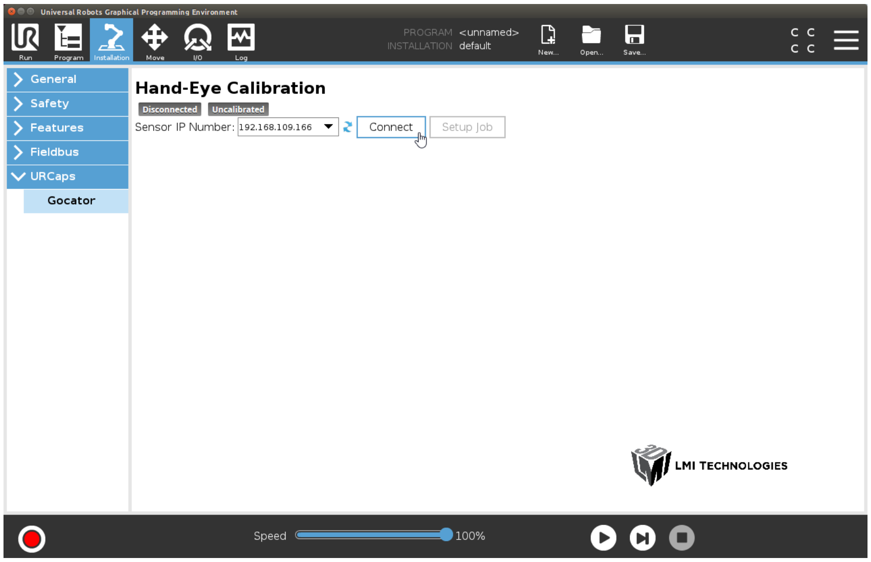
> Because this method of performing hand-eye calibration requires manually moving the sensor for multiple poses, LMI recommends using the Gocator Calibrate node instead. For more information, see *Gocator Calibrate* on page 828.

The following describes how to perform hand-eye calibration using the tool available in the robot's **Installation** tab.

For the calibration process, you will scan a calibration target from different poses. The URCap will then calculate a calibration matrix.

To perform the eye-in-hand hand-eye calibration (legacy):

1. In the Installation tab, under the URCaps category, click **Gocator**.
2. In the pane to the right, select the sensor’s IP address in the drop-down and click **Connect**.



The robot connects to the sensor and the interface for obtaining the target's pose displays in the pane.

If only 1 sensor is found in the system, it will be connected automatically.

3. (Optional) If you are using a ball bar for the calibration, click **Setup Job** to automatically add an instance of the Surface Ball Bar tool to the sensor.

   In addition to removing other measurement tools from the current job and adding an instance of the Surface Ball Bar tool, clicking **Setup Job** enables the required measurements and outputs, in the correct order, and sets the trigger mode to Software (for more information, see *Triggers* on page 89). For this reason, LMI recommends using this function.

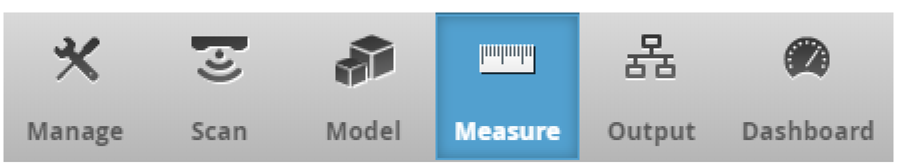
   ⚠ Clicking the **Setup Job** button removes *all* other tools from the current job!

4. Manually position the robot arm over the calibration target.

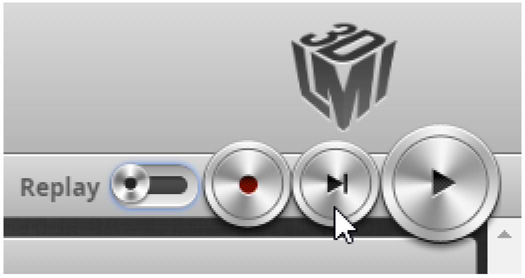
   Use the field of view, clearance distance, and measurement range specifications of your sensor to roughly position the sensor.The target should roughly in the middle of the sensor's scanning volume. For sensor specifications, see *Sensors* on page 880.

   You will determine and adjust the position of the sensor in relation to the calibration target in the following steps.

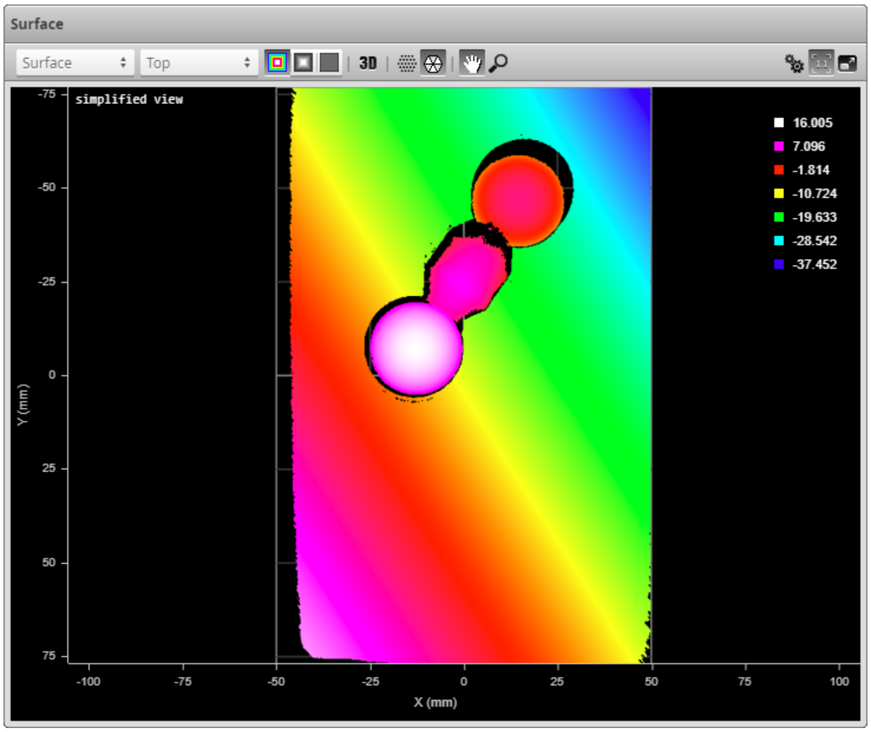
5. In the Gocator web interface, switch to the **Measure** page.



6. Click the Snapshot button to scan the target.

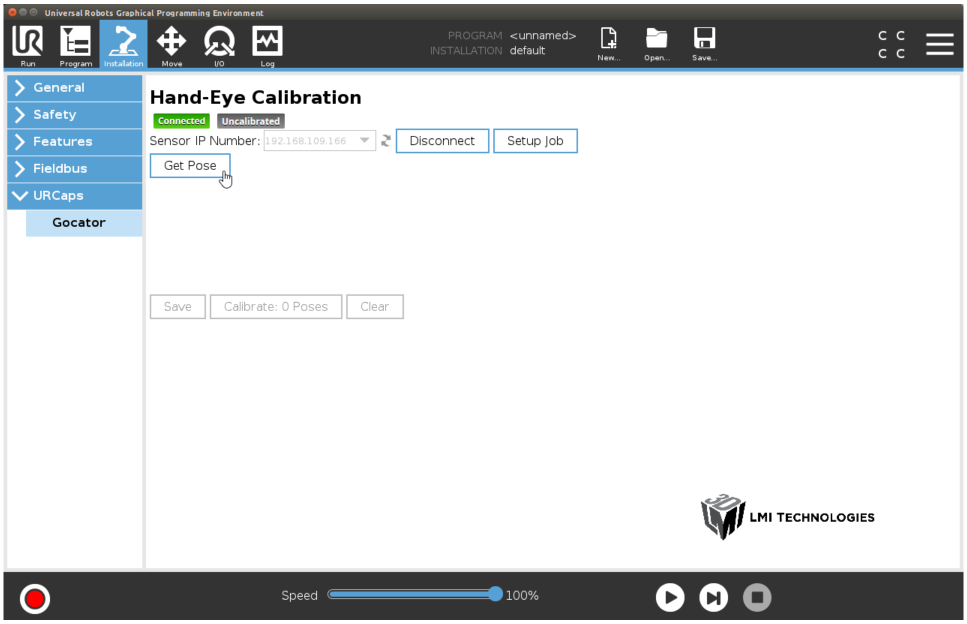


7. Examine the scan data in the data viewer to confirm that the calibration target is in the sensor's field of view (FOV), and adjust the position of the robot if required.



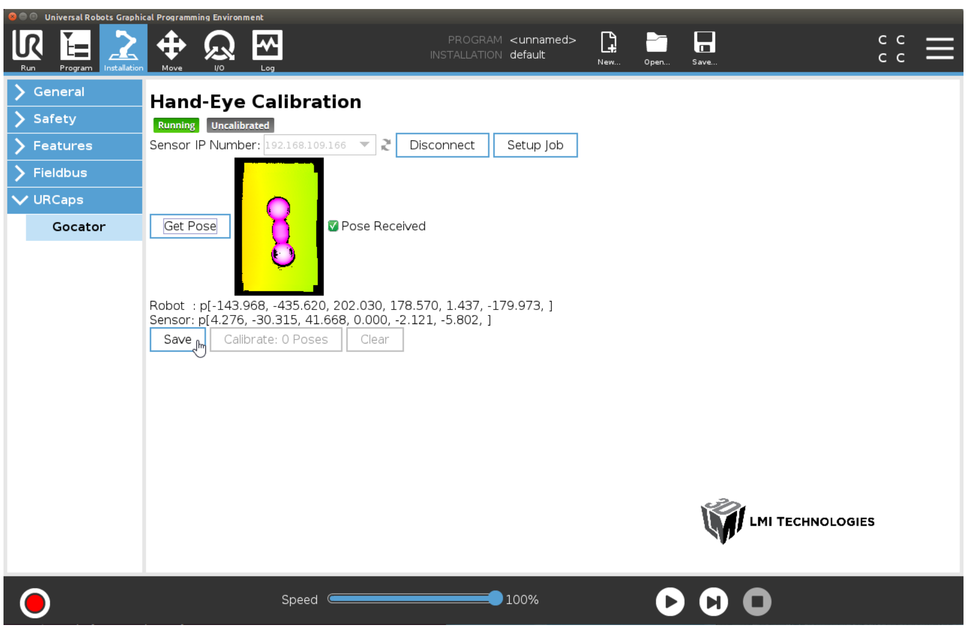
Typically, the entire calibration target must be in the sensor's field of view, and it should fill most of the field. There should be no other edges in the FOV.

8. In the Gocator URCap, click **Get Pose** and wait for an image of the calibration target to display in the URCap.



9. If the scan is successful, click **Save**.

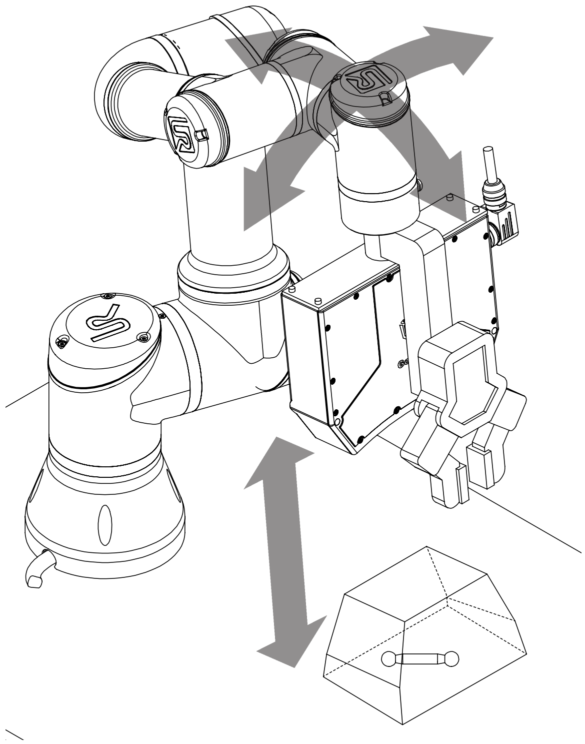
   The scan is successful if "Pose received" displays next to the scan data of the calibration target.



   The URCap displays numerical values representing the sensor and robot poses after the scan.

   If the scan is not successful, go to the next step without clicking **Save**.

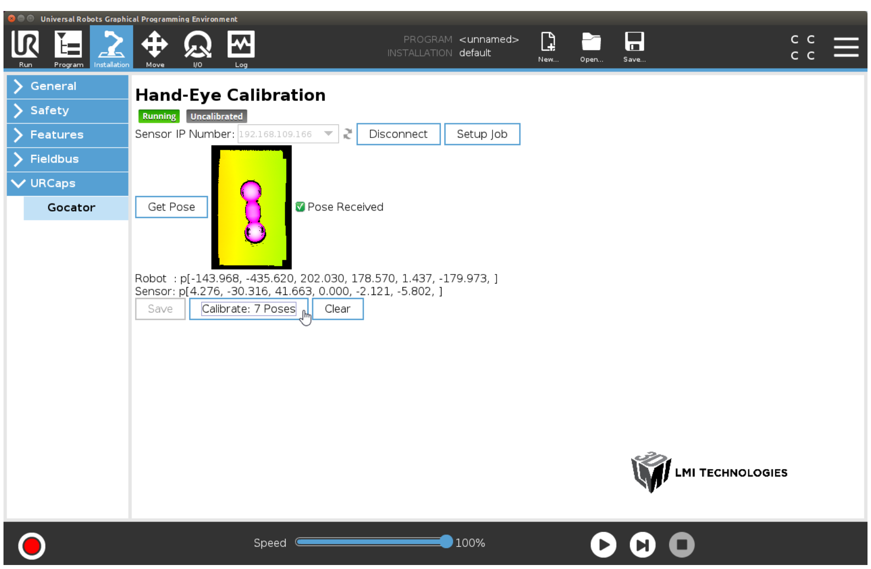
10. Reposition the robot slightly on all axes to change the position of the calibration target in the sensor's FOV, and repeat steps 5 to 9 to get another pose.



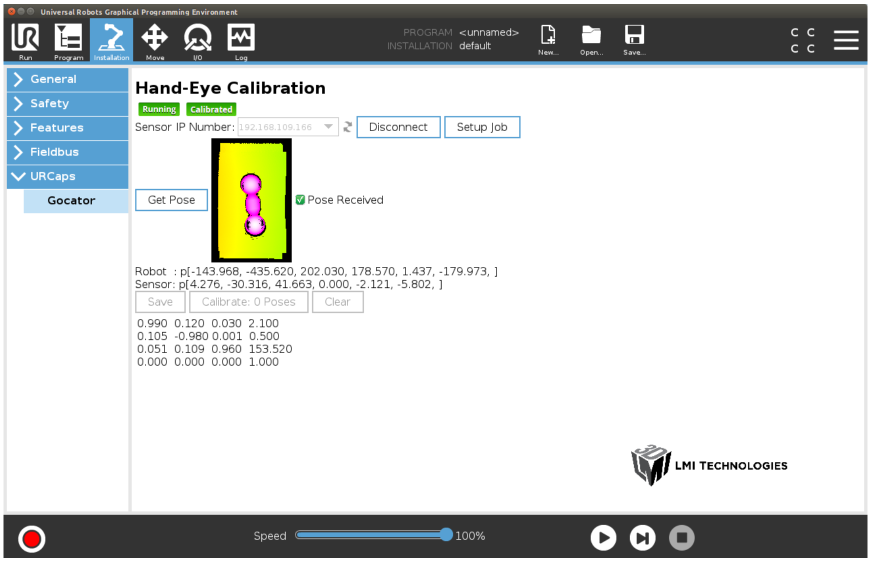
Note the following:

- The URCap requires at least 4 scans to calculate the calibration matrix and will not let you proceed to the next step otherwise. We recommend 8 scans or more.
- Do not rotate the sensor around the Z axis by more than 90 degrees: in the case of a ball-bar calibration target, the “upper” ball must remain the “upper” ball.

11. Once you have acquired a sufficient number of poses, click **Calibrate**.



The Gocator URCap calculates the calibration matrix and displays it. If the calibration fails, the URCap displays an error message.



Using the Gocator URCap Program Nodes

After you have performed the hand-eye calibration, you must add program nodes to the program tree to tell the robot to connect to the sensor, to load a job on the sensor, to trigger a scan, and then to receive measurements.

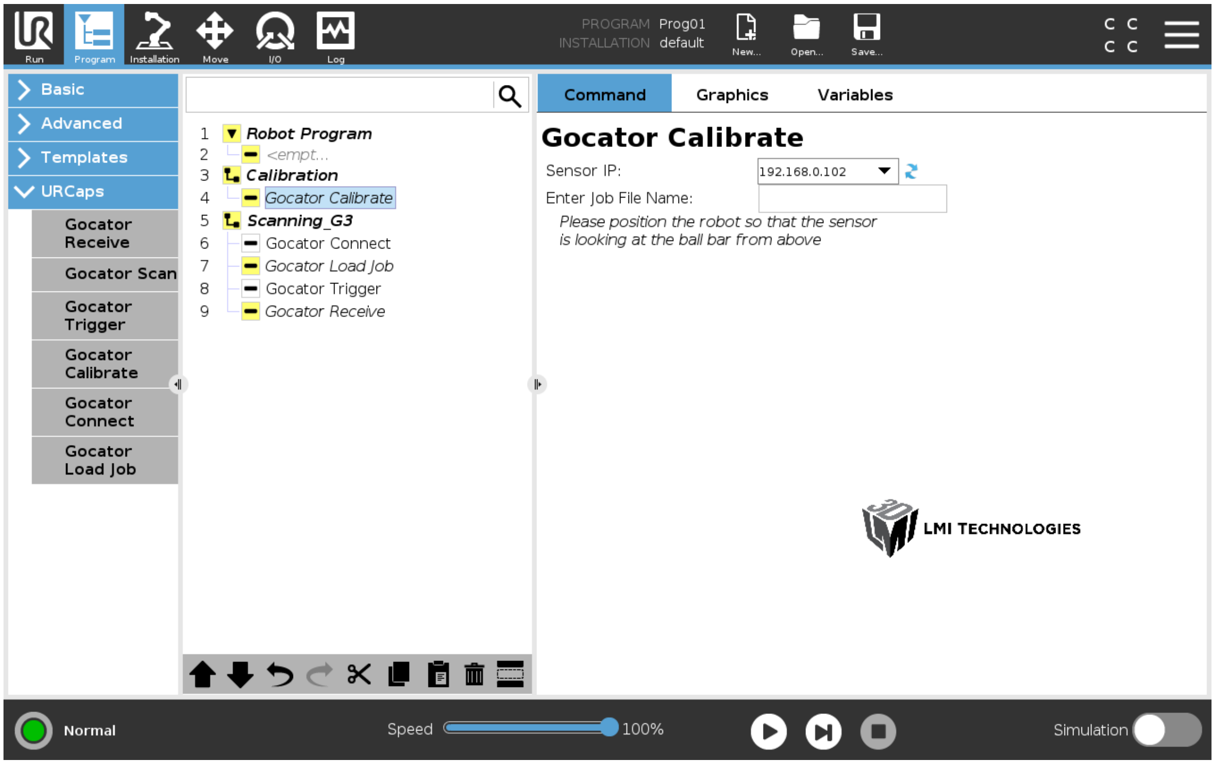
The program nodes are:

- Gocator Calibrate
- Gocator Connect
- Gocator Load Job
- Gocator Trigger
- Gocator Receive

The following sections describe the program nodes.

Gocator Calibrate

The Gocator Calibrate node automatically performs hand-eye calibration between a sensor and the robot flange. When the node runs, it moves the robot-mounted sensor over the calibration target multiple times to capture scans and return the required poses. After the node has run, the resulting sensor-robot transformation matrix is loaded onto the robot.



Before using the Gocator Calibrate node, make sure that a calibration job is present and configured on the sensor; for more information, see *Configuring the Sensor* on page 818. If you are using a ball bar target, LMI recommends following the procedure in *Using the Surface Ball Bar Tool* on page 819.

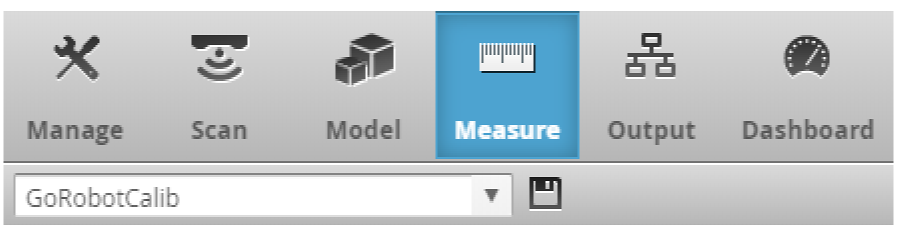
To configure the Gocator Calibrate node:

1. In the pane to the right in the Calibrate node, select the sensor's IP address in the drop-down.

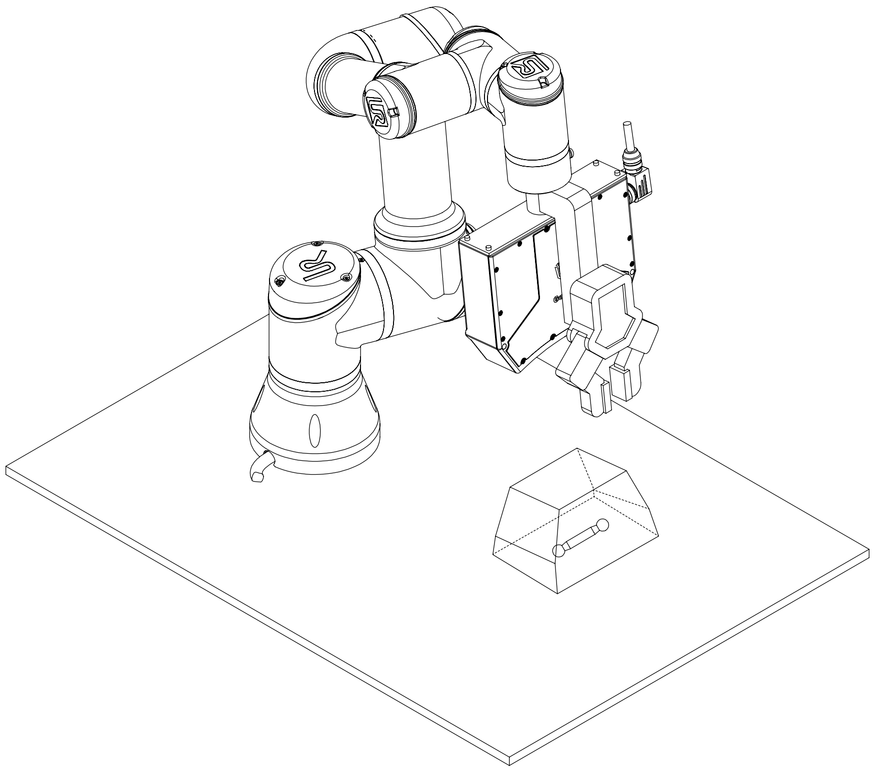
   The robot connects to the sensor.

2. Type the previously configured calibration job's name in the **Enter Job File Name** field.

   If necessary, you can get the job name from the job name field in the Gocator web interface.



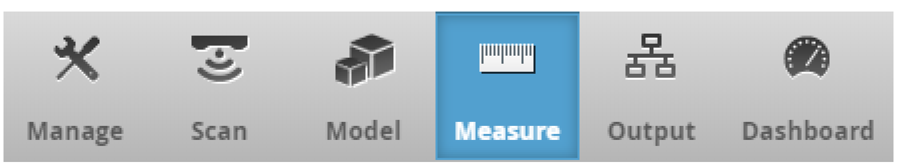
3. Manually position the robot arm over the calibration target.



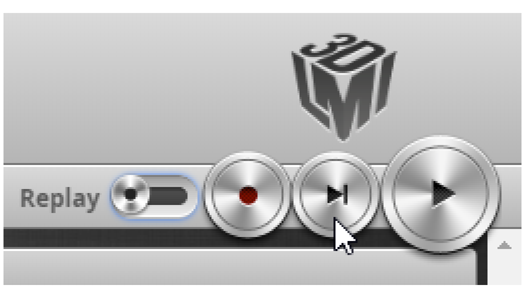
Use the field of view, clearance distance, and measurement range specifications of your sensor to correctly position the sensor.The target should be roughly in the middle of the sensor's scanning volume. For sensor specifications, see *Sensors* on page 880.

You will determine and adjust the position of the sensor in relation to the calibration target in the following steps.

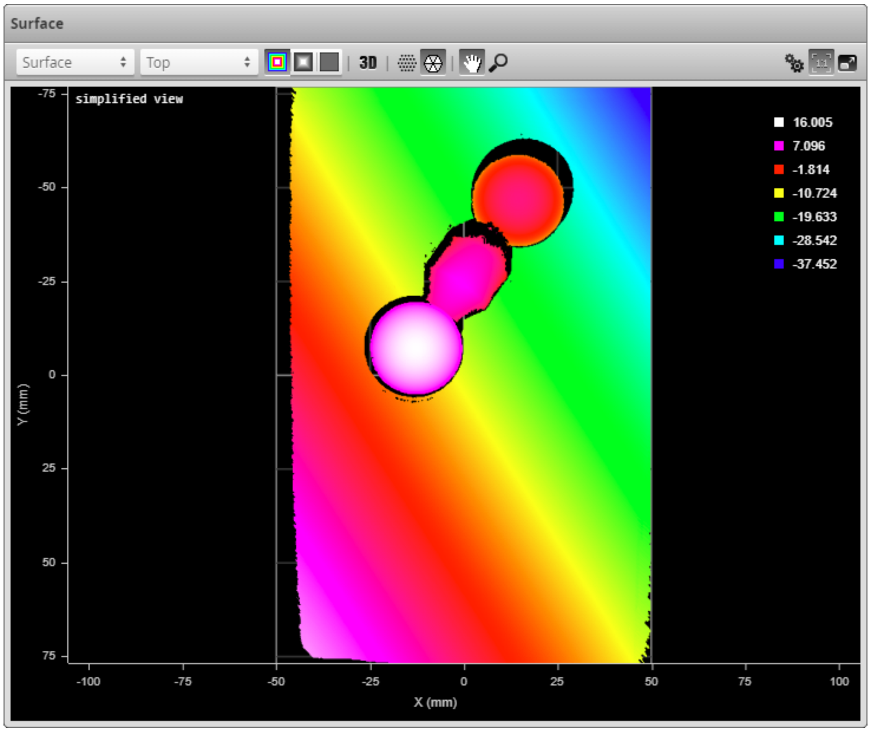
4. In the Gocator web interface, switch to the **Measure** page.



5. Click the Snapshot button to scan the target.



6. Examine the scan data in the sensor's data viewer to confirm that the calibration target is in the sensor’s field of view (FOV), and adjust the position of the robot if required.



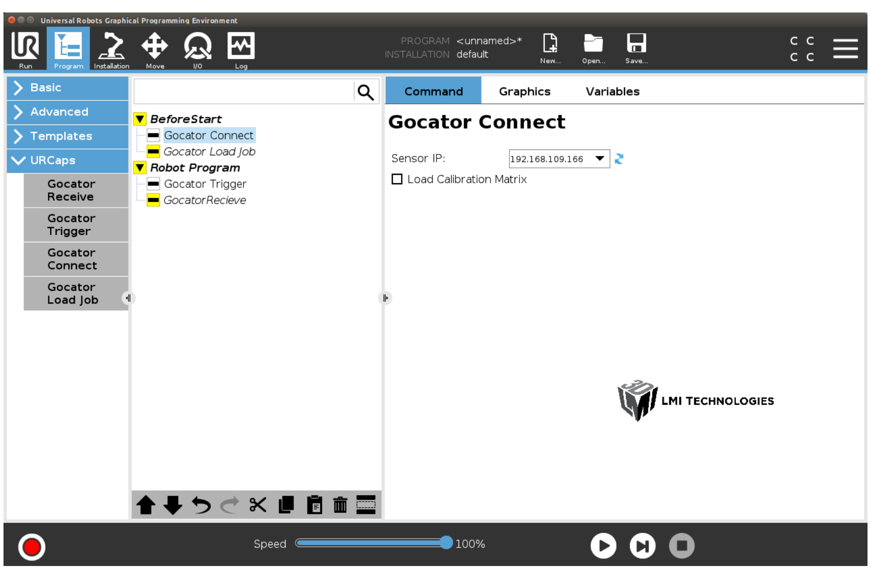
Typically, the entire calibration target must be in the sensor's field of view, and it should fill most of the field. There should be no other edges in the FOV.

Gocator Connect

This node connects the robot to the sensor and must appear before any of the other program nodes. We recommend that you place the *Gocator Connect* node in the *BeforeStart* sequence of the program tree.

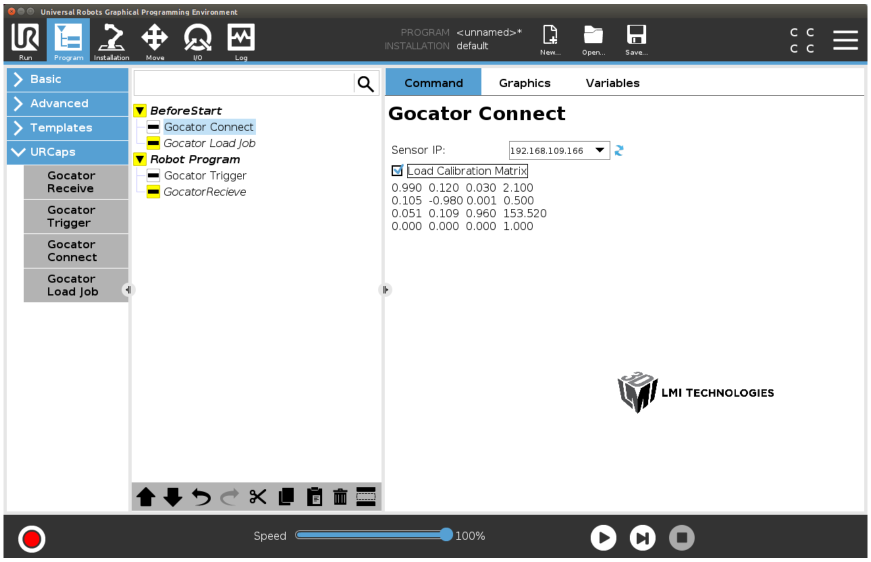
To configure the Gocator Connect node:

1. In the drop-down, choose the Gocator's IP address.



2. (Optional) If you wish to use calibrated matrix values in memory, check **Load Calibration Matrix**.

   The URCap loads the calibration matrix.



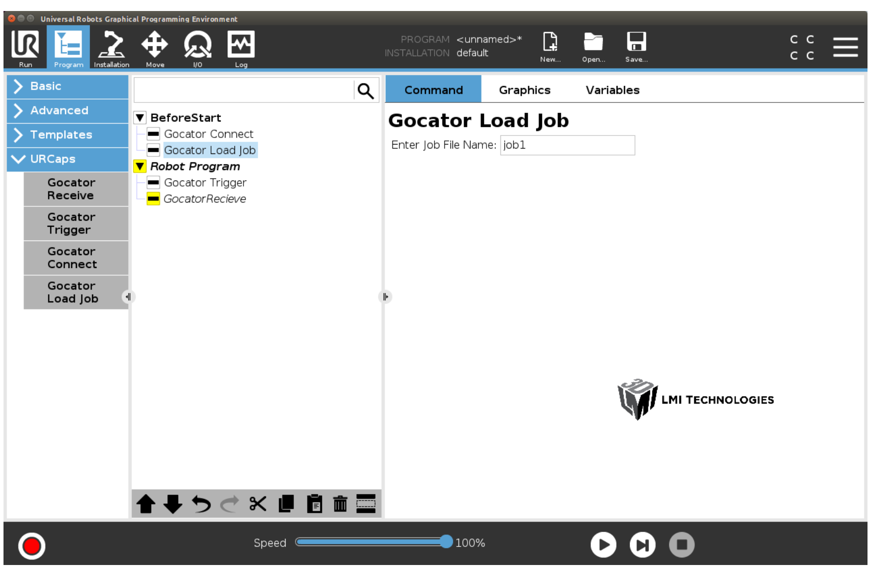
   If you do not load a calibration matrix, measurements from the Gocator will be in the sensor's coordinate system.

Gocator Load Job

This node loads a job file on the Gocator.

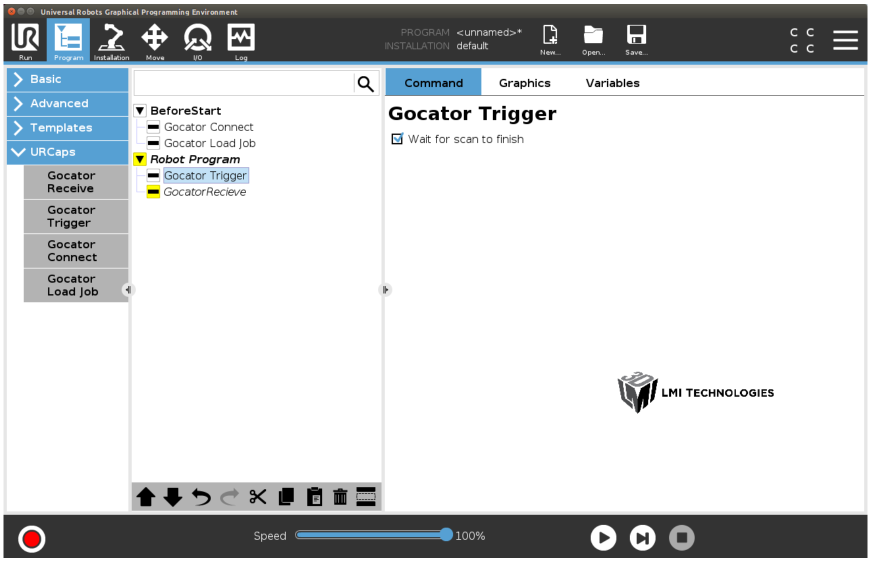
To configure the Gocator Load Job node:

- Type the job name in the field.



Gocator Trigger

The Gocator Trigger node causes the sensor to take a scan.



If you enable **Wait for scan to finish**, the node waits until the sensor has finished scanning and processing the results. If you disable the setting, the program will continue immediately after triggering the sensor.

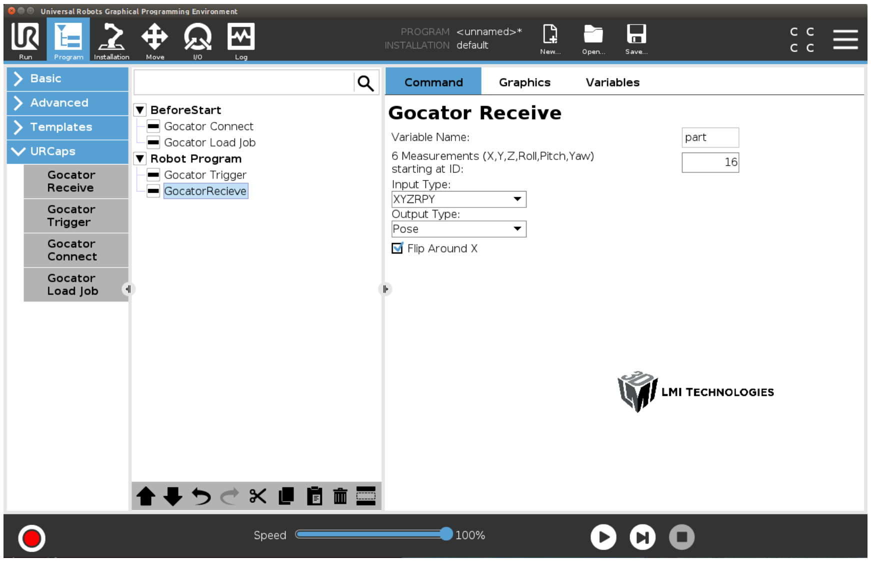
> You are responsible for making sure the sensor does not move for the duration of the exposure. You can configure the sensor's digital output to an Exposure End trigger event to have the sensor signal when the exposure has completed; for more information on digital output, see *Digital Output* on page 516. For information on configuring the robot's digital input, see the manual for your robot.

Gocator Scan

The Gocator Scan node is intended for use with G2 sensors only.

Gocator Receive

The Gocator Receive node receives measurements from the sensor and assigns them to a variable. You select the input type (the value or values received from the sensor) and the measurements on the sensor providing the values, and the output type (the value stored in the variable).



Gocator Receive node set to XYZRPY input type

You can use the Feature Robot Pose measurement tool on the Gocator sensor to easily generate X, Y, Z, raw, pitch, and yaw values, which the Gocator Receive node can then retrieve. For more information, see *Robot Pose* on page 502.

Gocator Receive Node Input Types

Type	Description
Single Measurement	Retrieves a single value using the provided measurement ID. The measurement will be provided as is, in the sensor's coordinate system (if relevant), and the calibration data will not be used. Typically used when measuring a feature on a part, rather than returning a part's location or orientation.
XYZRPY	Retrieves *six* consecutive measurements for X, Y, Z, roll, pitch, and yaw, *starting at* the specified measurement ID. The Gocator's web UI will still display the results in the sensor's coordinate system.
XYZ	Retrieves *three* consecutive measurements for X, Y, and Z, *starting at* the specified measurement ID.

Gocator Receive Node Output Types

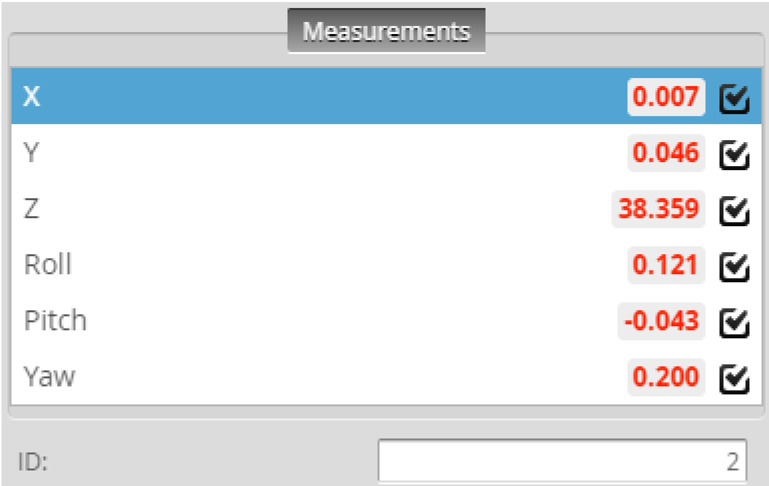
Type	Description
Double	Assigns the value to the variable. Only available when **Input Type** is set to **Single_ Measurement**.
Pose	Converts XYZRPY or XYZ input to the Universal Robots "XYZ and rotational vector" format. The variable can then be used directly in commands such as the Move command. If the input lacks angle information (XYZ input type), the rotational vector will be an approximation, especially the Z angle.
XYZ	Assigns the X, Y, and Z values to an array variable.

To configure the Receive node:

1. In the **Variable Name** text box field, type the name of the variable.

2. In the next field, enter the starting measurement ID from which you want to retrieve values.

   You can find the measurement ID in the **ID** field in the Gocator web interface, below the list of measurements. In the following, the selected measurement (X), has an ID of 2.



3. In the **Input Type** drop-down, choose the type you are retrieving from the sensor.

   For a list of input types, see *Gocator Receive Node Input Types* on the previous page.

4. In the **Output Type** drop-down, choose the type you are assigning to the variable.

   For a list of output types, see *Gocator Receive Node Output Types* on the previous page.

5. (Optional) If you intend to use the returned values to cause the robot to approach a part from above, check **Flip Around X**.

   Checking this option flips the part's coordinate system so that the Z axis is pointing down. This ensures that the coordinate systems of the part and the robot match so that the robot can correctly approach the part from above.

   Leave the option unchecked if you need the actual pose of the part.

Development Kits

These sections describe the following development kits:

- Software Development Kit (GoSDK)
- Gocator Development Kit (GDK)

GoSDK

The Gocator Software Development Kit (GoSDK) includes open-source software libraries and documentation that can be used to programmatically access and control Gocator sensors. To get the latest version of the *Gocator SDK* package, go to https://downloads.lmi3d.com/, choose your product from the Product Downloads section, and download it from the Download Center.

For information on the ports the SDK uses (for example, in order to ensure ports are not blocked over your network), see *Required Ports* on page 39.

	If you switch jobs or make changes to a job using the SDK or a protocol (from a PLC), the switch or changes are not automatically displayed in the web interface: you must refresh the browser to see these.

	The Gocator protocol is always on and its output is always available, regardless of the output you choose. This allows simultaneous connections via an SDK application and a PLC, letting you for example archive or display scan data on a PC while controlling equipment with a PLC.

You can download the Gocator SDK from within the Web interface.

To download the SDK:

1. Go to the **Manage** page and click on the **Support** category
2. Next to **Software Development Kit (SDK)**, click **Download**
3. Choose the location for the SDK on the client computer.

Applications compiled with previous versions of the SDK are compatible with sensor firmware if the major version numbers of the protocols match. For example, an application compiled with version 5.0 of the SDK (which uses protocol version 5.0) will be compatible with a sensor running firmware version 5.1 (which uses protocol version 5.1). However, any new features in firmware version 5.1 would not be available.

Applications compiled using SDK version 4.x are compatible with sensors running firware 5.x.

Applications compiled using SDK version 3.x are not compatible with sensors running firmware 4.x. In this case, you must rewrite the application with the SDK version corresponding to the sensor firmware in use.

For more information about programming with the SDK, refer to the class reference and sample programs included in the SDK.

Setup and Locations

Class Reference

The full GoSDK class reference is found by accessing the following file:

14400-x.x.x.xx_SOFTWARE_GO_SDK\GO_SDK\doc\GoSdk\Gocator_3x00\GoSdk.html

Examples

Examples showing how to perform various operations are provided, each one targeting a specific area. For Visual Studio, the examples can be found in solution files specific to different versions of Visual Studio. For example, *GoSdk-2017.sln* is for use with Visual Studio 2017. A make file for Linux systems is also provided.

🗋	To compile the examples in Visual Studio, you may be need to retarget the solution to the installed Windows SDK version. You can do this through the **Retarget solution** option in the solution context menu.

To run the GoSDK examples, make sure the required DLLs are copied beside the executable. In most cases only *GoSDK.dll* and *kApi.dll* are required, but with .NET and the accelerator additional DLLs are needed. Please refer to the SDK samples to see which DLLs are required.

Example Project Environment Variable

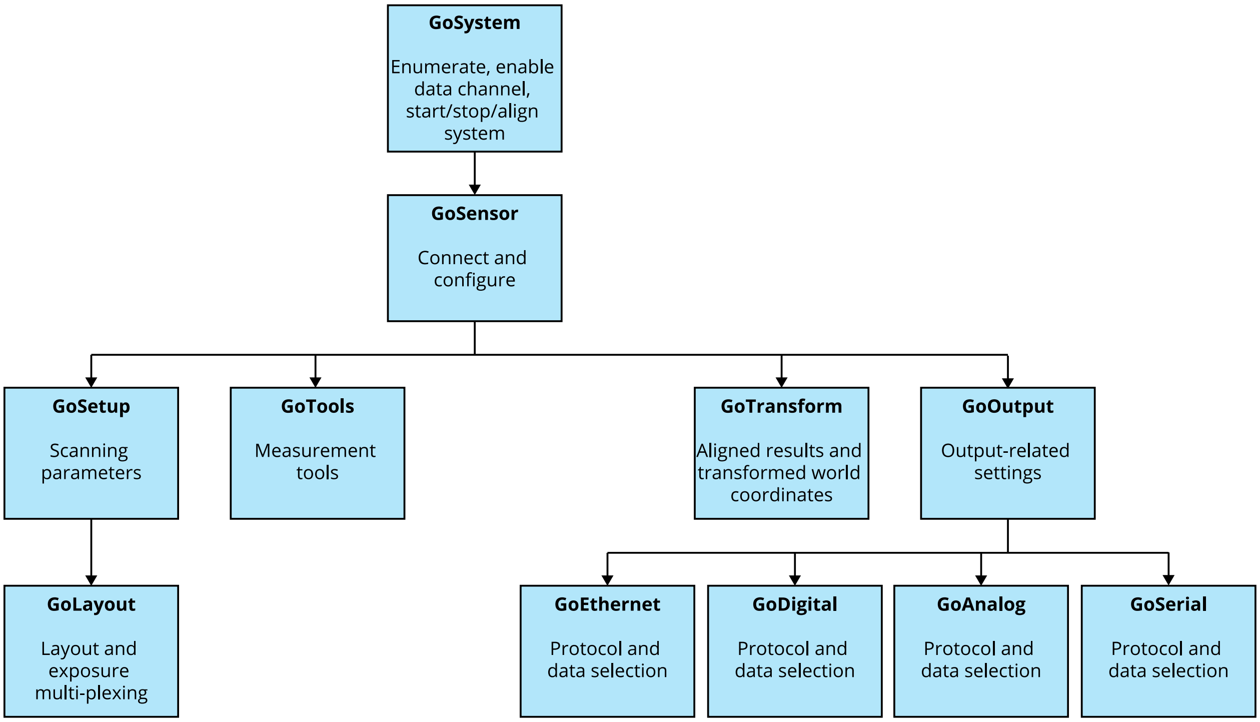
All GoSDK example projects use the environment variable *GO_SDK_4*. The environment variable should point to the *GO_SDK* directory, for example, *C:\14400-4.0.9.156_SOFTWARE_GO_SDK\GO_SDK*.

Header Files

Header files are referenced with GoSdk as the source directory, for example: #include <GoSdk/GoSdk.h>. The SDK header files also reference files from the *kApi* directory.

Functional Hierarchy of Classes

This section describes the functional hierarchy of the classes in the Gocator SDK ("GoSDK"). In the following diagram, classes higher in the hierarchy often provide resources for classes lower in the hierarchy, and for this reason should be instantiated earlier in a client application.



GoSystem

The *GoSystem* class is the top-level class in the SDK. Multiple sensors can be enabled and connected in one *GoSystem*. Only one *GoSystem* object is required for multi-sensor control.

Refer to the *How To Use The Open Source SDK To Fully Control A Gocator Multi-sensor System* how-to guide in http://lmi3d.com/sites/default/files/APPNOTE_Gocator_4.x_Multi_Sensor_Guide.zip for details on how to control and operate a multi-sensor system using the SDK.

> All objects that are explicitly created by the user or passed via callbacks should be destroyed by using the *GoDestroy* function.

GoSensor

GoSensor represents a physical sensor. If the physical sensor is the Main sensor in a dual-sensor setup, it can be used to configure settings that are common to both sensors.

GoSetup

The *GoSetup* class represents a device's configuration. The class provides functions to get or set all of the settings available in the web interface.

GoSetup is included inside *GoSensor*. It encapsulates scanning parameters, such as exposure, resolution, spacing interval, etc. For parameters that are independently controlled for Main and Buddy sensors, functions accept a role parameter.

GoLayout

The *GoLayout* class represents layout-related sensor configuration.

GoTools

The *GoTools* class is the base class of the measurement tools. The class provides functions for getting and setting names, retrieving measurement counts, etc.

GoTransform

The *GoTransform* class represents a sensor transformation and provides functions to get and set transformation information, as well as encoder-related information.

GoOutput

The *GoOutput* class represents output configuration and provides functions to get the specific types of output (Analog, Digital, Ethernet, and Serial). Classes corresponding to the specific types of output (*GoAnalog*, *GoDigital*, *GoEthernet*, and *GoSerial*) are available to configure these outputs.

Data Types

The following sections describe the types used by the SDK and the kApi library.

Value Types

GoSDK is built on a set of basic data structures, utilities, and functions, which are contained in the *kApi* library.

The following basic value types are used by the *kApi* library.

Value Data Types

Type	Description
k8u	8-bit unsigned integer
k16u	16-bit unsigned integer
k16s	16-bit signed integer
k32u	32-bit unsigned integer
k32s	32-bit signed integer
k64s	64-bit signed integer
k64u	64-bit unsigned integer
k64f	64-bit floating number
kBool	Boolean, value can be kTRUE or kFALSE
kStatus	Status, value can be kOK or kERROR
kIpAddress	IP address

Output Types

The following output types are available in the SDK.

Output Data Types

Data Type	Description
GoAlignMsg	Represents a message containing an alignment result.

Data Type	Description
GoBoundingBoxMatchMsg	Represents a message containing bounding box based part matching results.
GoDataMsg	Represents a base message sourced from the data channel. See *GoDataSet Type* below for more information.
GoEdgeMatchMsg	Represents a message containing edge based part matching results.
GoEllipseMatchMsg	Represents a message containing ellipse based part matching results.
GoExposureCalMsg	Represents a message containing exposure calibration results.
GoMeasurementMsg	Represents a message containing a set of GoMeasurementData objects.
GoProfileIntensityMsg	Represents a data message containing a set of profile intensity arrays.
GoProfileMsg	Represents a data message containing a set of profile arrays.
GoRangeIntensityMsg	Represents a data message containing a set of range intensity data.
GoRangeMsg	Represents a data message containing a set of range data.
GoResampledProfileMsg	Represents a data message containing a set of resampled profile arrays.
GoSectionMsg	Represents a data message containing a set of section arrays.
GoSectionIntensityMsg	Represents a data message containing a set of profile intensity arrays.
GoStampMsg	Represents a message containing a set of acquisition stamps.
GoSurfaceIntensityMsg	Represents a data message containing a surface intensity array.
GoSurfaceMsg	Represents a data message containing a surface array.
GoVideoMsg	Represents a data message containing a video image.

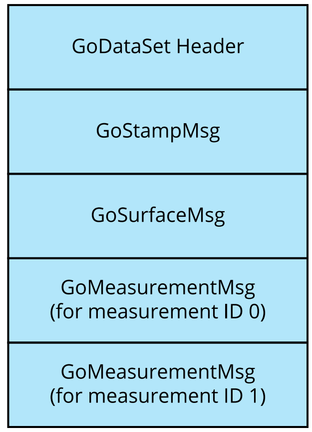
Refer to the *GoSdkSamples* sample code for examples of acquiring data using these data types.

See *Setup and Locations* on page 837 for more information on the code samples.

GoDataSet Type

Data are passed to the data handler in a *GoDataSet* object. The *GoDataSet* object is a container that can contain any type of data, including scan data (sections or surfaces), measurements, and results from various operations. Data inside the *GoDataSet* object are represented as messages.

The following illustrates the content of a *GoDataSet* object of a surface mode setup with two measurements.



After receiving the *GoDataSet* object, you should call *GoDestroy* to dispose the *GoDataSet* object. You do not need to dispose objects within the *GoDataSet* object individually.

> All objects that are explicitly created by the user or passed via callbacks should be destroyed by using the *GoDestroy* function.

Measurement Values and Decisions

Measurement values and decisions are 32-bit signed values (k32s). See *Value Types* on page 839 for more information on value types.

The following table lists the decisions that can be returned.

Measurement Decisions

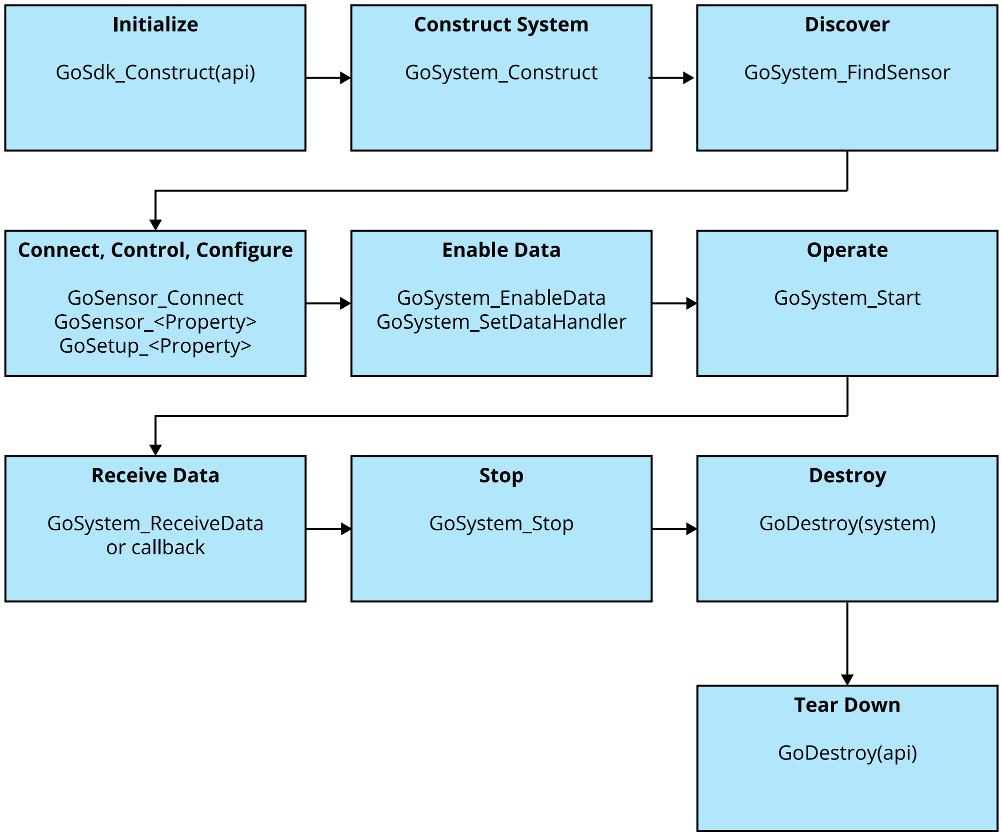
Decision	Description
1	The measurement value is between the maximum and minimum decision values. This is a pass decision.
0	The measurement value is outside the maximum and minimum. This is a fail decision.
-1	The measurement is invalid (for example, the target is not within range). Provides the reason for the failure.
-2	The tool containing the measurement is anchored and has received invalid measurement data from one of its anchors. Provides the reason for the failure.

Refer to the *SetupMeasurement* example for details on how to add and configure tools and measurements. Refer to the *ReceiveMeasurement* example for details on how to receive measurement decisions and values.

> You should check a decision against <=0 for failure or invalid measurement.

Operation Workflow

Applications created using the SDK typically use the following programming sequence



See *Setup and Locations* on page 837 for more information on the code samples referenced below.

Sensors must be connected before the system can enable the data channel.

All GoSDK data functions are named *Go<Object>_<Function>*, for example, *GoSensor_Connect*. For property access functions, the convention is *Go<Object>_<Property Name>* for reading the property and *Go<Object>_Set<Property Name>* for writing it, for example, *GoMeasurement_DecisionMax* and *GoMeasurement_SetDecisionMax*, respectively.

Initialize GoSdk API Object

Before the SDK can be used, the *GoSdk* API object must be initialized by calling *GoSdk_Construct(api)*:

```
kAssembly api = kNULL;
if ((status = GoSdk_Construct(&api)) != kOK)
{
    printf("Error: GoSdk_Construct:%d\n", status);
    return;
}
```

When the program finishes, call *GoDestroy(api)* to destroy the API object.

Discover Sensors

Sensors are discovered when *GoSystem* is created, using *GoSystem_Construct*. You can use *GoSystem_SensorCount* and *GoSystem_SensorAt* to iterate all the sensors that are on the network.

GoSystem_SensorCount returns the number of sensors physically in the network.

Alternatively, use *GoSystem_FindSensorById* or *GoSystem_FindSensorByIpAddress* to get the sensor by ID or by IP address.

Refer to the *Discover* example for details on iterating through all sensors. Refer to other examples for details on how to get a sensor handle directly from IP address.

Connect Sensors

Sensors are connected by calling *GoSensor_Connect*. You must first get the sensor object by using *GoSystem_SensorAt*, *GoSystem_FindSensorById*, or *GoSystem_FindSensorByIpAddress*.

Configure Sensors

Some configuration is performed using the *GoSensor* object, such as managing jobs, uploading and downloading files, scheduling outputs, setting alignment reference, etc. Most configuration is however performed through the *GoSetup* object, for example, setting scan mode, exposure, exposure mode, active area, speed, alignment, filtering, subsampling, etc. Surface generation is configured through the *GoSurfaceGeneration* object and part detection settings are configured through the *GoPartDetection* object.

See *Functional Hierarchy of Classes* on page 837 for information on the different objects used for configuring a sensor. Sensors must be connected before they can be configured.

Refer to the *Configure* example for details on how to change settings and to switch, save, or load jobs. Refer to the *BackupRestore* example for details on how to back up and restore settings.

Enable Data Channels

Use *GoSystem_EnableData* to enable the data channels of all connected sensors. *GoSystem_EnableData* should only be used when you also receive and discard the data in your application.

Perform Operations

Operations are started by calling *GoSystem_Start*, *GoSystem_StartAlignment*, and *GoSystem_StartExposureAutoSet*.

Refer to the *StationaryAlignment* and *MovingAlignment* examples for details on how to perform alignment operations. Refer to the *ReceiveRange*, *ReceiveProfile*, and *ReceiveWholePart* examples for details on how to acquire data.

Example: Configuring and starting a sensor with the API

```
#include <GoSdk/GoSdk.h>

void main()
{
```

```
kIpAddress ipAddress;
GoSystem system = kNULL;
GoSensor sensor = kNULL;
GoSetup setup = kNULL;

//Construct the GoSdk library.
GoSdk_Construct(&api);

//Construct a sensor system object.
GoSystem_Construct(&system, kNULL);

//Parse IP address into address data structure
kIpAddress_Parse(&ipAddress, SENSOR_IP);

//Obtain GoSensor object by sensor IP address
GoSystem_FindSensorByIpAddress(system, &ipAddress, &sensor)

//Connect sensor object and enable control channel
GoSensor_Connect(sensor);

//Enable data channel
GoSensor_EnableData(system, kTRUE)

//[Optional] Setup callback function to receive data asynchronously
//GoSystem_SetDataHandler(system, onData, &contextPointer)
//Retrieve setup handle
setup = GoSensor_Setup(sensor);

//Reconfigure system to use time-based triggering.
GoSetup_SetTriggerSource(setup, GO_TRIGGER_TIME);

//Send the system a "Start" command.
GoSystem_Start(system);

//Data will now be streaming into the application
//Data can be received and processed asynchronously if a callback function has been
//set (recommended)
//Data can also be received and processed synchronously with the blocking call
//GoSystem_ReceiveData(system, &dataset, RECEIVE_TIMEOUT)
//Send the system a "Stop" command.
GoSystem_Stop(system);

//Free the system object.
GoDestroy(system);

//Free the GoSdk library
```

```
    GoDestroy(api);
}
```

Limiting Flash Memory Write Operations

Several operations and Gocator SDK functions write to the sensor's flash memory. The lifetime of the flash memory is limited by the number of write cycles. Therefore it is important to avoid frequent write operation to the sensor's flash memory when you design your system with the SDK.

	Power loss during flash memory write operation will also cause sensors to enter rescue mode.

	This topic applies to all Gocator sensors.

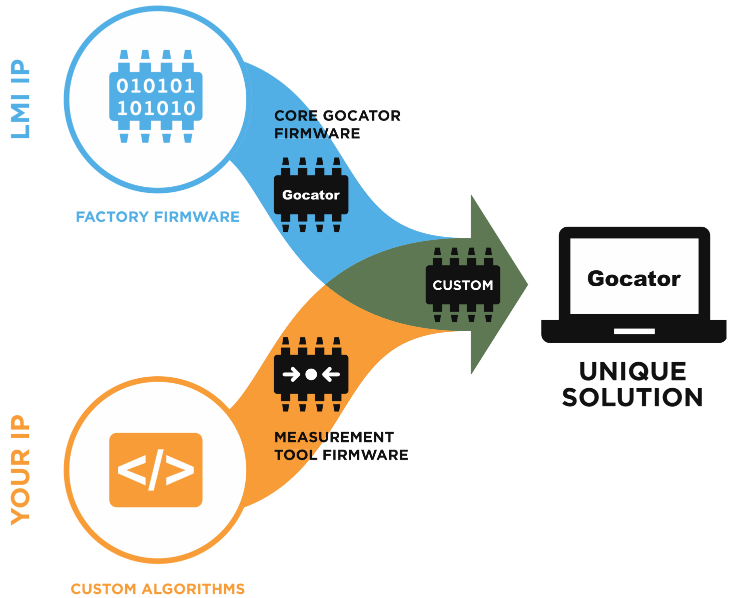
SDK Write-Operation Functions

Name	Description
GoSensor_Restore	Restores a backup of sensor files.
GoSensor_RestoreDefaults	Restores factory default settings.
GoSensor_CopyFile	Copies a file within the connected sensor. The flash write operation does not occur if GoSensor_CopyFile function is used to load an existing job file. This is accomplished by specifying “_live” as the destination file name.
GoSensor_DeleteFile	Deletes a file in the connected sensor.
GoSensor_SetDefaultJob	Sets a default job file to be loaded on boot.
GoSensor_UploadFile	Uploads a file to the connected sensor.
GoSensor_Upgrade	Upgrades sensor firmware.
GoSystem_StartAlignment	When alignment is performed with alignment reference set to fixed, flash memory is written immediately after alignment. GoSensor_ SetAlignmentReference() is used to configure alignment reference.
GoSensor_SetAddress	Configures a sensor's network address settings.
GoSensor_ChangePassword	Changes the password associated with the specified user account.
GoTransform_SetEncoderResolution	Sets the encoder resolution.
GoTransform_SetSpeed	Sets the travel speed.
GoTransform_SetX	Sets the transformation X component.
GoTransform_SetY	Sets the transformation Y component.
GoTransform_SetZ	Sets the transformation Z component.
GoTransform_SetXAngle	Sets the transformation X-angle.
GoTransform_SetYAngle	Sets the transformation Y-angle.
GoTransform_SetZAngle	Sets the transformation Z-angle.

System created using the SDK should be designed in a way that parameters are set up to be appropriate for various application scenarios. Parameter changes not listed above will not invoke flash memory write operations when the changes are not saved to a file using the GoSensor_CopyFile function. Fixed alignment should be used as a means to attach previously conducted alignment results to a job file, eliminating the need to perform a new alignment.

GDK

The Gocator Development Kit (GDK) is a framework for developing and testing custom Gocator tools containing your own algorithms, and then deploying them to Gocator sensors.



Custom tools created with the GDK act much like native Gocator data output tools (providing measurements, geometric features, data and generic outputs) with support for multiple input parameters), running at native speeds and taking advantage of features such as anchoring. The GDK supports all data types, and tools created with the GDK use the same data visualization as native tools.

Benefits

When you use the GDK to create custom measurement tools, you have complete control over how and where your custom measurement tools can be used, which protects your intellectual property.

You can also easily troubleshoot and modify your tools on-site, letting you respond quickly to your customers' urgent issues.

Supported Sensors

The GDK is available for free for the following Gocator sensors:

- Gocator 1300 series
- Gocator 2100 series
- Gocator 2300 series
- Gocator 2400 series
- Gocator 2500 series
- Gocator 2880
- Gocator 3210 and Gocator 3500 series

Typical Workflow

The following is the typical workflow for creating and deploying custom measurement tools:

- Develop and build tools using the GDK project files and libraries in Microsoft Visual Studio, targeting Win32.
- Debug the tools using the emulator on a PC.
- Build the tools into a custom firmware binary.
- Upload the custom firmware to a sensor.

Installation and Class Reference

The GDK project and library files are in the *GDK* package (`14524-x.x.x.xx_SOFTWARE_GDK.zip`). To download the package, go to https://downloads.lmi3d.com/, choose your product from the Product Downloads section, and download it from the Download Center.

After downloading the package, extract the package to a directory.

You can access full installation and setup instructions, as well as the complete class reference documentation, by double-clicking the *Guide* shortcut under the root directory.

bin	8/4/2016 2:08 AM	File folder
doc	8/4/2016 2:10 AM	File folder
Gocator	8/4/2016 2:14 AM	File folder
lib	8/4/2016 2:15 AM	File folder
pkg	8/4/2016 2:16 AM	File folder
Platform	8/4/2016 2:16 AM	File folder
res	8/4/2016 2:16 AM	File folder
Guide	8/3/2016 1:39 PM	Shortcut

Required Tools

The GDK requires Microsoft Visual Studio 2017, as well as various other tools provided in the *GDK Prerequisites* package (`14525_x.x.x.x_SOFTWARE_GDK_Prerequisites.zip`). This package is available in LMI's Downloads Center (see above for download location).

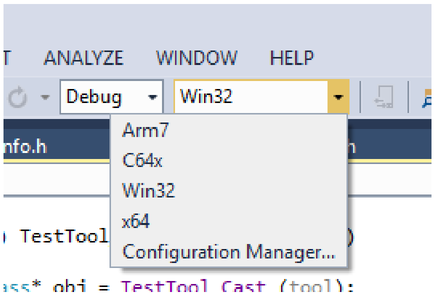
Getting Started with the Example Code

The best way to get started is with the GDK sample code. You can find the sample projects under `Gocator\GDKSampleApp`. This project is ready for you to build and use as a template for new projects.

Start by opening `GDK.sln` in Visual Studio 2017.

Building the Sample Code

You can build the sample code for working with either the emulator or a sensor. To do this, choose the target and then build the solution.



The following targets are available:

- Win32/x64 for debugging code and emulating a sensor to test tools (on a PC)
- Arm7 for building for Gocator 2300C and 2400 series sensors
- C64x for Gocator 1300, 2300A, 2300B, 3210, and 3506 series sensors

The Win32 target supports Debug and Release builds. The Arm7 and C64x targets (sensors) only the support Release builds.

Tool Registration

For a tool to be available to a user in the sensor web interface, you must add it to the project assembly in `Asm.c`.

```
#include <GdkSampleApp/Asm.h>
#include <GdkSampleApp/TestProfileSelect.h>
#include <GdkSampleApp/TestSurfaceSelect.h>
#include <GdkSampleApp/TestSurfaceConfiguration.h>
#include <GdkSampleApp/TestSurfaceGraphics.h>
#include <Gdk/GdkLib.h>
#include <GoSensor/Version.h>
#include <GoSensorAppLib/GsaDef.h>
#include <GoSensorAppLib/GsaAsm.h>

kBeginAssembly(Tool, ToolAsm, TOOL_VERSION, GOCATOR_VERSION)
    kAddDependency(GdkLib)
    kAddType(TestProfileSelect)
       kAddType(TestSurfaceSelect)
    kAddType(TestSurfaceConfiguration)
       kAddType(TestSurfaceGraphics)
kEndAssembly()
```

You can add multiple tools in a GDK project. As seen above, `TestProfileSelect`, `TestSurfaceSelect`, `TestSurfaceConfiguration`, etc. will be available for users from the drop-down menu in the **Tools** panel in sensor's web interface.

Tool Definitions

You must add standard entry functions (methods) for each tool. The class table declares the entry functions:

```
kBeginClass(Tool, TestTool, GdkTool)
    kAddVMethod(TestTool, kObject, VRelease)
    kAddVMethod(TestTool, GdkTool, VInit)
    kAddVMethod(TestTool, GdkTool, VName)
    kAddVMethod(TestTool, GdkTool, VDescribe)
    kAddVMethod(TestTool, GdkTool, VNewToolConfigInstanced)
    kAddVMethod(TestTool, GdkTool, VNewMeasurementConfigInstanced)
    kAddVMethod(TestTool, GdkTool, VUpdateConfigInstanced)
    kAddVMethod(TestTool, GdkTool, VNewFeatureConfigInstanced)
    kAddVMethod(TestTool, GdkTool, VNewToolDataOutputConfigInstanced)
    kAddVMethod(TestTool, GdkTool, VIsVisible)
    kAddVMethod(TestTool, GdkTool, VCalcDataOutputRegionInstanced)
    kAddVMethod(TestTool, GdkTool, VStart)
    kAddVMethod(TestTool, GdkTool, VStop)
    kAddVMethod(TestTool, GdkTool, VProcess)
kEndClass()

ToolFx (kStatus) TestTool_VDescribe(GdkToolInfo toolInfo)
{
    GdkMeasurementInfo mmt;
    GdkParamsInfo params;
    GdkParamInfo paramInfo;

    kCheck(GdkToolInfo_SetTypeName(toolInfo, TEST_PROFILE_SELECT_TOOL_NAME));
    kCheck(GdkToolInfo_SetLabel(toolInfo, TEST_PROFILE_SELECT_TOOL_LABEL));

    kCheck(GdkToolInfo_SetSourceType(toolInfo, GDK_DATA_TYPE_UNIFORM_PROFILE));

    ...
```

The function `<Tool Name>_VDescribe` describes the tool and its basic configuration. This function is called during sensor start-up. For more information on entry functions, see *Entry Functions* below.

	Make sure the `VDescribe` function for each tool is properly formed. Significant issues with this function (for example, overwriting memory) could prevent the sensor from starting.

	You should use the emulator to debug tools *before* deploying tools to sensors.

Entry Functions

The following table describes the main entry functions.

Function	Description
VDescribe	Defines the tool's name, data types, acceptable source options, configuration parameters, and at least one measurement.
VStart	Called when the sensor starts running (that is, the user clicks the Run button). The function gets parameters from `GtTool`. You typically allocate memory in this function.
VProcess	Called every time data is received while the sensor is running.

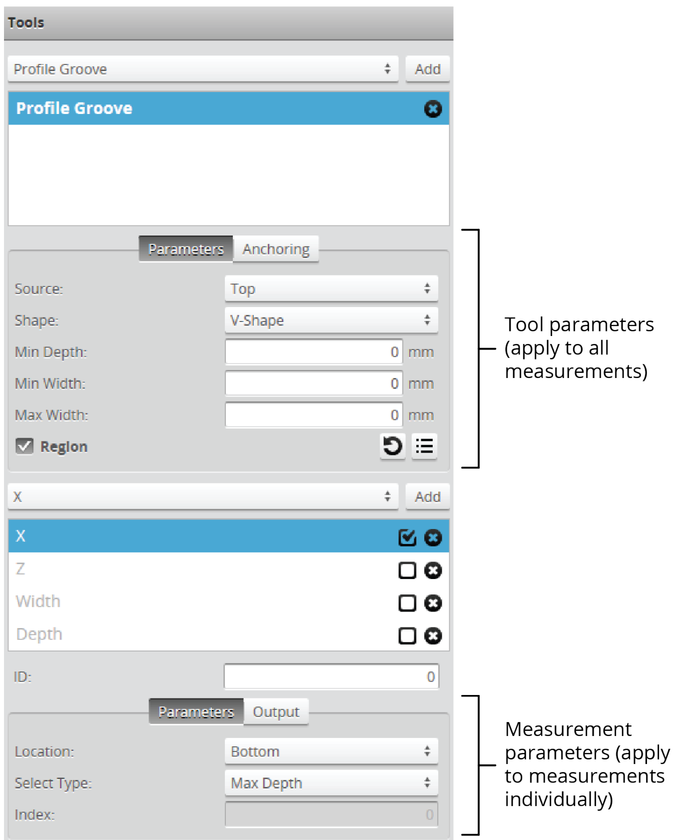
Function	Description
VStop	Called when the user clicks the Stop button.

The `TestSurfaceConfiguration` example shows how to create and modify parameters based on other user settings.

For full descriptions of these functions, see the GDK class reference documentation (see *Installation and Class Reference* on page 848 for information on installing the documentation).

Parameter Configurations

Each tool has two levels of parameters: tool parameters and measurement parameters.



A tool can contain multiple measurements. In the image above, the Groove tool contains four measurements: X, Z, Width, and Depth. Each tool has one set of tool parameters and each measurement in a tool has one set of measurement parameters.

The following table lists the functions that provide advanced or interactive control for setting up tool and measurement parameters:

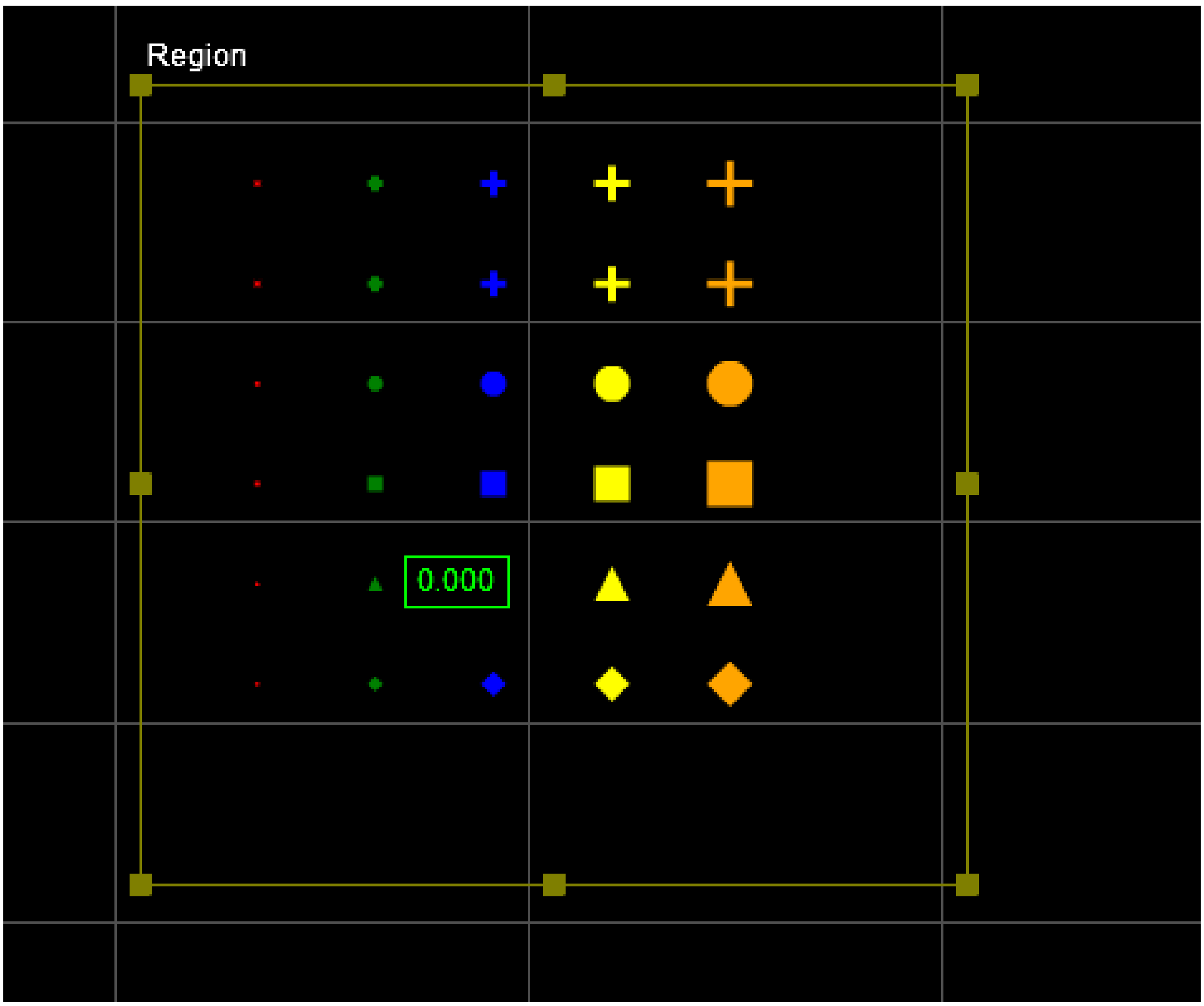
Function	Description
VNewToolConfig	Advanced method for setting default values of tool parameters based on the current sensor configuration (for example, active area). Called when a new tool is added in the interface.

Function	Description
VNewMeasurementConfig	Advanced method for setting default values of measurement parameters based on the current sensor configurations (for example, active area). Called when measurements in a tool is are added in the interface.
VUpdateConfig	Advanced method for updating the configuration based on parameters set by users.

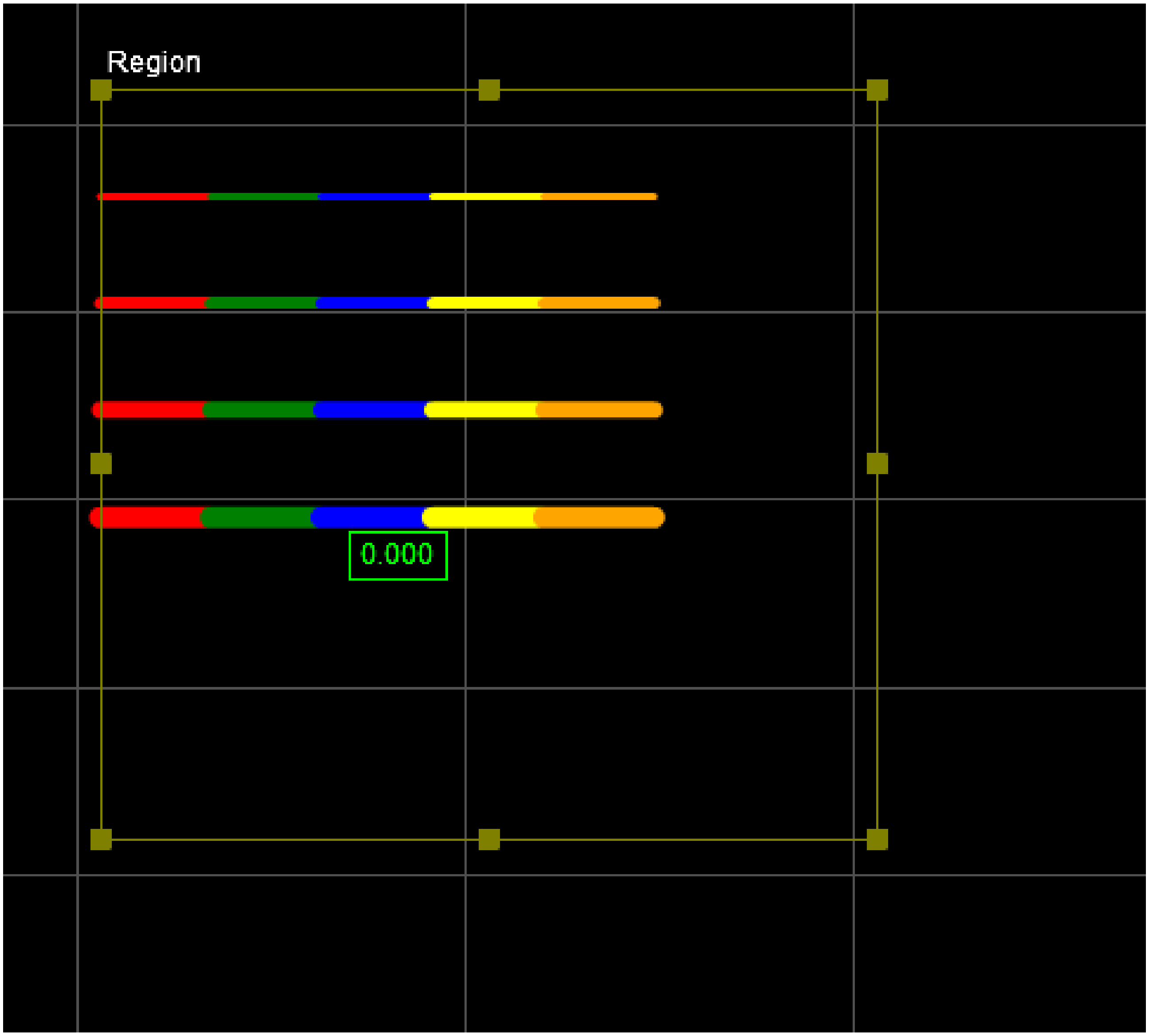
For full descriptions of these functions, see the GDK class reference documentation (see *Installation and Class Reference* on page 848 for information on installing the documentation).

Graphics Visualization

The `GdkGraphic` function supports points and lines.



Point graphics



Line graphics

To create graphics:

1. Use `GdkGraphic_Construct` to create a graphic object.
2. Use `GdkGraphicPointSet_Construct` to create points or `GdkGraphicLineSet_Construct` to create lines.
3. Add the points and lines to the graphic object using `GdkGraphic_AddPointSet` and `GdkGraphic_AddLineSet`.
4. Output using GdkToolOutput_`SetRendering`.

The following illustrates the process:

```
kTest(GdkGraphic_Construct(&graphic, kObject_Alloc(tool)));
```

```
kTest(GdkGraphicPointSet_Construct(&pointSet, 4.0, kMARKER_SHAPE_CROSS, kCOLOR_LIME,
&point32f, 1, kObject_Alloc(tool)));

kTest(GdkGraphic_AddPointSet(graphic, pointSet));

kTest(GdkToolOutput_SetRendering(output, measurementIndex, graphic));
```

The GDK example `TestSurfaceGraphics` shows how to use the graphics functions.

Graphic functions take an array of `kPoint3d32f`. It does NOT accept `kPoint3d64f`.

Debugging Your Tools

We highly recommend using the emulator to debug tools you create with the GDK. By using a sensor support file and previously recorded scan data, downloaded from a physical sensor, you can completely simulate standalone and multi-sensor configurations on a PC to test your tools.

To debug your tools in the emulator:

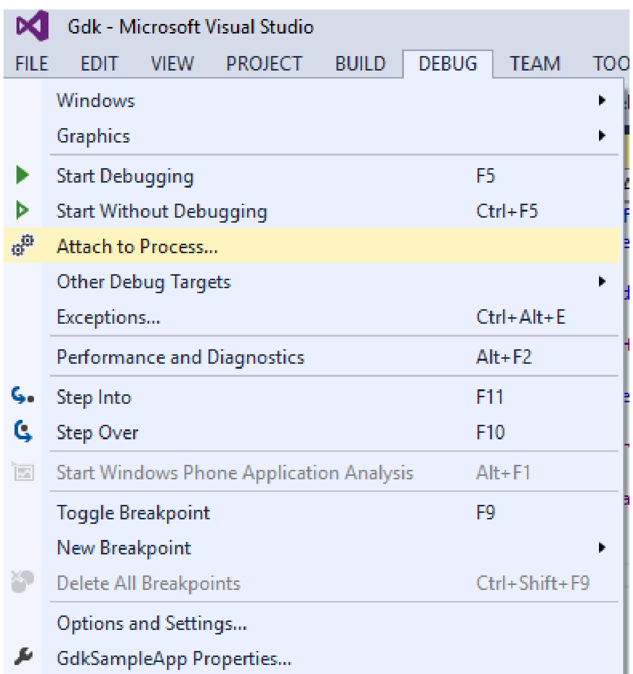
1. Compile your code using the Win32 target (Debug or Release).
2. In the output directory, rename the DLL with the same name as your project to GdkApp.dll.

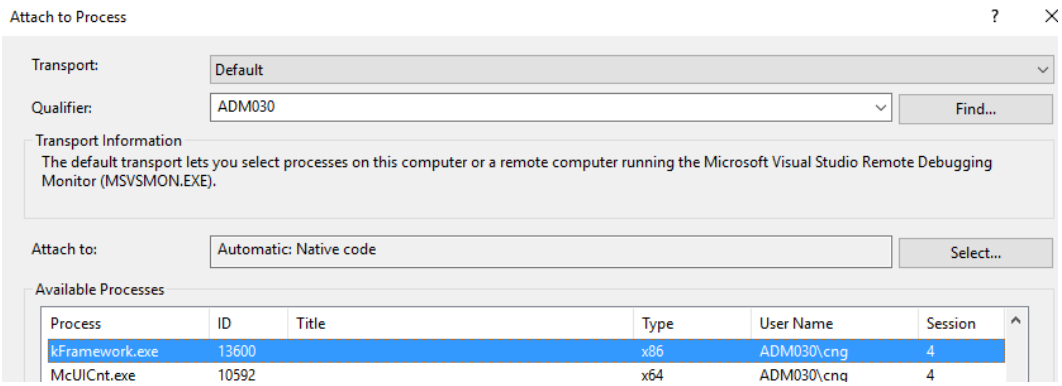
   For example, if your project is called `MyGDKTools`, the resulting DLL should be called `MyGDKTools.dll`. You rename this DLL to GdkApp.dll.

   The output directories are as follows:

   Release: `win32`

   Debug: `win32d`
3. Launch the emulator from same output directory as in step 2.
4. In the emulator, choose a scenario and start it.
5. In Visual Studio, attach the debugger to the `kFramework.exe` process.





> `kFramework.exe` is only loaded after a user selects a scenario and starts the emulator session.

Debugging Entry Functions

VStart, *VProcess*, and *VStop* are called whenever a data record is played back in the emulator (that is, when a user clicks on the Next button or types the frame number in the frame field) with at least one tool instance. For more information on playback controls, see *Recording, Playback, and Measurement Simulation* in the Gocator user manual.

VDescribe however is called when the DLL loads, before the debugger can attach to the `kFramework.exe` process. To debug *VDescribe*, we recommend testing the function calls by putting them in *VInit*.

> For information on building targets for testing in the emulator, see the GDK class reference documentation.

Tips

The following sections provide useful information for creating custom measurement tools.

Backward Compatibility with Older Versions of Tools

When loading a recording or job file that contains a custom measurement tool, the parameters in the loaded recording or job file must match those in the firmware.

By default, if declared parameters are missing from the configuration, a job file or a recording will fail to load.

There are two ways to provide backward compatibility with older parameter sets.

Define new parameters as optional

Mark a parameter as optional with the function `GdkParamInfo_SetIsOptional`. When a parameter is marked as optional, parameter parsing functions succeed even if the parameter is missing from the configuration. The missing parameter is initialized with default value.

Configuration Versioning

Over the lifetime of a tool, you may need to make changes to its interface (for example, changing or removing parameters). The user-defined aspects of a tool interface—its parameters and measurements—are captured by `GDKToolVersionInfo` objects.

By default, a tool has just one version (`GdkToolInfo_FirstVersion`), but more versions may be added using `GdkToolInfo_AddVersion`. Whenever the interface of a tool has changed, a new version can be registered so that the new interface can be correctly parsed by the framework.

When the configuration of a tool instance is saved, the version used at the time is also saved. This saved version is used by the framework to parse the configuration. If a version is not defined by the firmware implementation, then that tool instance will not be active.

During run-time, you can query the version of the configuration of a tool instance by using `GdkToolCfg_Version`. You can then interpret the parameters depending on the version the configuration is saved in.

```
GdkFx(kStatus) GdkExampleTool_VDescribe(GdkToolInfo info)
{
       kCheck(GdkToolInfo_SetLabel(info, "Example"));

       kCheck(GdkToolInfo_SetSourceType(info, GDK_DATA_TYPE_UNIFORM_PROFILE));
       kCheck(GdkToolInfo_AddSourceOption(info, GDK_DATA_SOURCE_TOP));

       kCheck(GdkExampleTool_DescribeV0(info));
       kCheck(GdkExampleTool_DescribeV1(info));

       kCheck(GdkToolInfo_SetDefaultVersion(info, GdkToolInfo_VersionAt(info, 1)));

       return kOK;
}

GdkFx(kStatus) GdkExampleTool_DescribeV0(GdkToolInfo info)
{
       kCheck(GdkParamsInfo_Add(GdkToolInfo_Params(info), "RefRegion", GDK_PARAM_TYPE_PROFILE_
REGION, "Ref Region", kNULL));
       kCheck(GdkParamsInfo_Add(GdkToolInfo_Params(info), "Region", GDK_PARAM_TYPE_PROFILE_
REGION, "Region", kNULL));
       kCheck(GdkToolInfo_SetFirstVersionName(info, ""));

       return kOK;
}

GdkFx(kStatus) GdkExampleTool_DescribeV1(GdkToolInfo info)
{
       GdkToolVersionInfo versionInfo;

       // Auto-version

       kCheck(GdkToolInfo_AddVersion(info, kNULL, &versionInfo));
       kCheck(GdkToolVersionInfo_UseBase(versionInfo, GdkToolInfo_FirstVersion(info)));
       kCheck(GdkParamsInfo_AddFloat(GdkToolVersionInfo_Params(versionInfo), "BaseScale",
kNULL, 2.0, kNULL));

       return kOK;
}
```

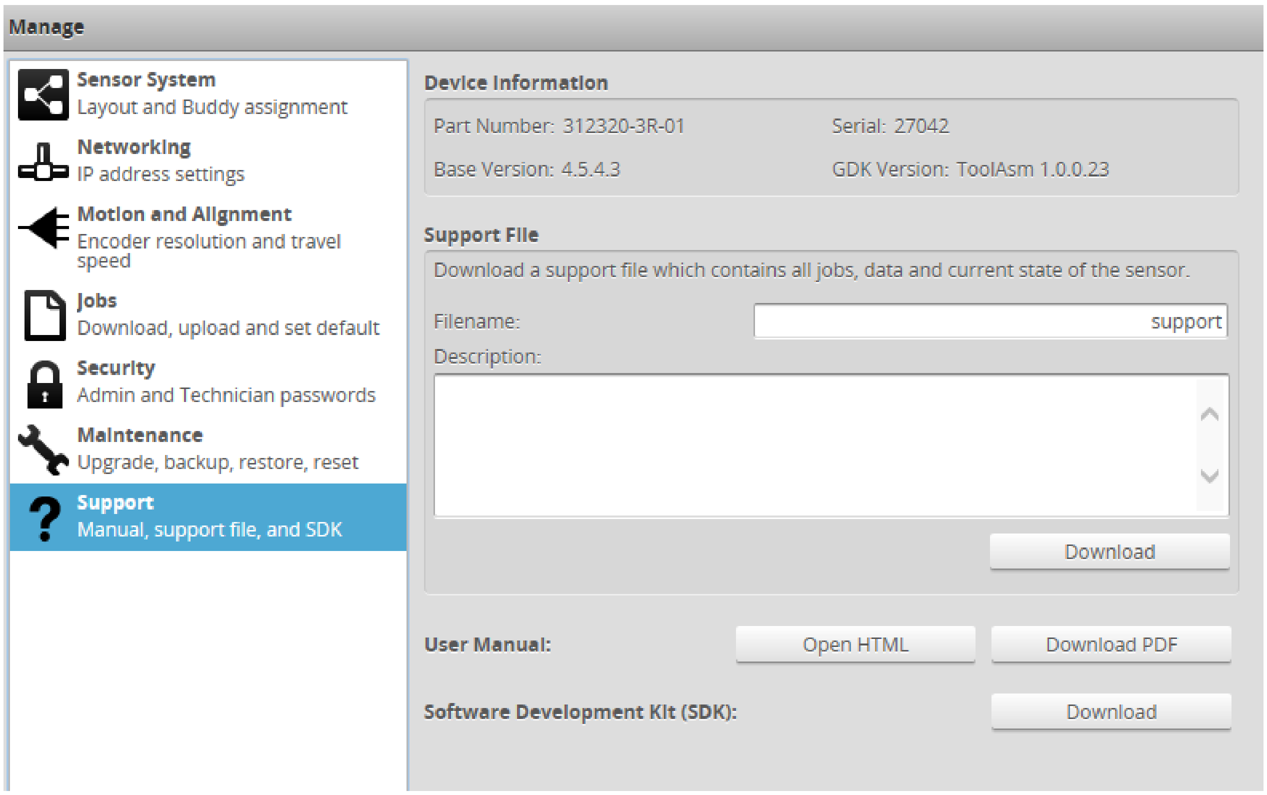
Adding a new measurement does not require special handling. The new measurement is just not instantiated in a previous configuration.

Version

You can define the version number of your tools in `Asm.x.h`.

```
#define TOOL_VERSION     kVersion_Stringify_(1, 0, 0, 23)
```

The version is displayed on the **Manage** page, in the **Support** category.



Common Programming Operations

The following sections describe common programming operations.

Input Data Objects

The `VProcess` function receives a `GdkToolInput` object as input. This object is a container where the information and actual data of the received input is stored.

```
GdkInputItem item = GdkToolInput_Find(input, obj->dataSource);
GdkDataInfo itemInfo = GdkInputItem_Info(item);
```

The `GdkToolInput_Find` and `GdkInputItem_Info` functions are used to extract the item and info objects. These objects can then be used to retrieve the input data and information (for example, offset and resolution) associated to the input. The following are some examples:

Computing actual height information using offset and scale

```
k64f height = rangeSrc[index] * scale->z + offset->z;
```

Extracting height information from profiles and surfaces.

The `TestProfileSelect` and `TestSurfaceSelect` examples show how to perform these operations.

Setup and Region Info during Tool Initialization

Memory allocation is often done in the `VInit` or `VStart` function. To retrieve sensor and data information such as active area settings and data scale outside of `VProcess`, you can use the following function:

```
GdkDataInfo info = GdkSensorInfo_DataSource(GdkTool_SensorInfo(tool), GDK_DATA_SOURCE_
TOP);
```

Computing Region Based on the Offset from an Anchor Source

Just like built-in measurement tools, custom tools created with the GDK can be anchored to another tool (GDK-based tools or built-in tools).

To compute the offset region:

```
TestToolClass* obj = TestTool_Cast_(tool);
GdkParams params = GdkToolCfg_Parameters(config);
const kPoint3d64f* anchor = GdkToolInput_AnchorPosition(input);
GdkRegionXZ64f offsetRegion = { k64F_NULL, k64F_NULL, k64F_NULL, k64F_NULL };

param = GdkParams_Find(params, "Region");
obj->region = *GdkParam_AsProfileRegion(param);

offsetRegion = obj->region;
offsetRegion.x += anchor->x;
offsetRegion.z += anchor->z;
```

In the code above, we first retrieve the tool's region settings (before anchoring is applied), and then adjust the region based on the results from the anchored source in *VProcess*. If the anchored source fails, the tools will not be invoked.

The `TestProfileSelect` and `TestSurfaceSelect` examples show how to extract height information from anchored regions.

For more information on anchoring, see *Measurement Anchoring* in the Gocator user manual.

Part Matching

When part matching is enabled, the tool receives translated and corrected surface data. If part matching fails for the current scan (for example, the quality score is too low), the tools will not be invoked.

For more information on part matching, see *Part Matching* in the user manual.

Accessing Sensor Local Storage

You can access a sensor's local storage by using the kFile API.

For example, to read and write a file to a sensor's storage, you could use the following:

```
#include <kApi/Io/kFile.h>

…

ToolFx(kStatus) TestTool_VStart(TestTool tool)
{
        …

        kFile_Save("test.txt", stringBuf, (kSize) 1024);
        kFile_Load("test.txt", stringBuf, &bufLen, kNULL);
```

Print Output

In the emulator, you can send output to Visual Studio or to programs such as DebugView by using the `OutputDebugString` function.

```
GtsFx(kStatus) TestTool_Trace(const kChar* format, ...)
{
    kStatus status = kOK;
    kChar debugLine[256];

    kVarArgList argList;
    kVarArgList_Start_(argList, format);
    {
        status = kStrPrintvf(debugLine, 256, format, argList);
    }
    kVarArgList_End_(argList);
    OutputDebugStringA(debugLine);
    return status;
}
```

OutputDebugString is NOT supported on sensor targets. Use #ifdef to comment out the code when compiling against sensor targets.

GoRobot

GoRobot is an SDK that provides an abstract interface for developers to write calibration applications and other applications that combine sensor measurements with robot movement. The library encapsulates primitives such as robot poses and calibration matrices, and simplifies access to Gocator functionality.

The GoRobot SDK sits on top of the standard Gocator SDK. Currently, the SDK supports Universal Robots, Kuka, and Yaskawa. The sample code for a Kuka GoRobot driver is provided as a template and can be used to integrate robot models from other vendors.

Because the GoRobot SDK depends on the Gocator SDK, you must first install and configure the Gocator SDK as documented in *GoSDK* on page 836.

Installation

The GoRobot SDK is available in the Gocator Utilities package (14405-x.x.x.x_SOFTWARE_GO_Utilities.zip) in LMI's Download Center. To get the package, go to https://downloads.lmi3d.com/, choose your product from the Product Downloads section, and download it from the *Software* section of the appropriate release version.

Class Reference and Sample Code

The GoRobot SDK class reference is available within the utilities package you downloaded, in `\Integration\GoRobot\doc\GoRobot.html`.

You can find sample code in `\Integration\GoRobot\GoRobotSamples`.

Tools

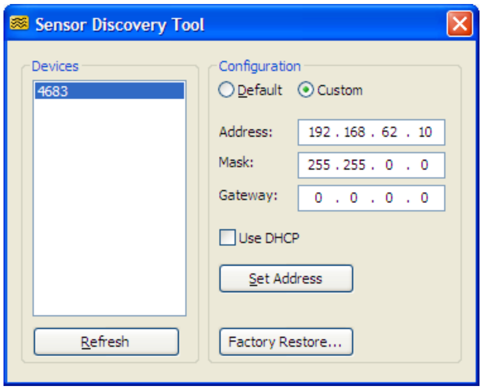
The following sections describe some of the tools provided with a Gocator sensor, as well as the CSV format that a sensor can export. For information on the integrations available with a sensor, see *Integrations* on page 650.

- Bandwidth Tool: Use this tool to diagnose bandwidth-related issues.
- CSV Converter Tool: Used to convert CSV data exported from a sensor to several formats. See *CSV Converter Tool* on the next page.
- Discovery Tool: Used to find sensors on a network. See *Sensor Discovery Tool* below.
- Track Editor: Used with the Surface Track tool. For more information, see *Track* on page 449.

Sensor Discovery Tool

If a sensor's network address or administrator password is forgotten, the sensor can be discovered on the network and/or restored to factory defaults by using the Sensor Discovery software tool. This tool can be obtained from the downloads area of the LMI Technologies website: http://www.lmi3D.com.

After downloading the utility package [14405-x.x.x.x_SOFTWARE_GO_Utilities.zip], unzip the file and run the Sensor Discovery Tool [Tools > Discovery > kDiscovery.exe].



Any sensors that are discovered on the network will be displayed in the Devices list.

To change the network address of a sensor:

1. Select the **Custom** option.
2. Enter the new network address information.
3. Click **Set Address**.

To restore a sensor to factory defaults:

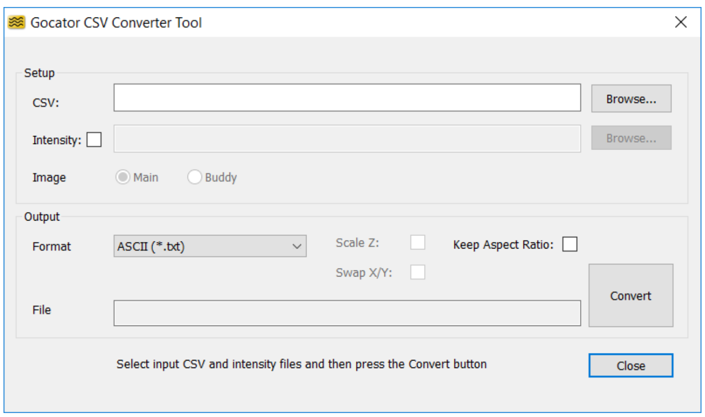
1. Select the sensor serial number in the **Devices** list.
2. Press the **Factory Restore...** button.
   Confirm when prompted.

	The Sensor Discovery tool uses UDP broadcast messages to reach sensors on different subnets. This enables the Sensor Discovery tool to locate and re-configure sensors even when the sensor IP address or subnet configuration is unknown.

CSV Converter Tool

The CSV Converter tool lets you convert data exported from a Gocator sensor in the CSV format to several formats (see table below). For more information on exporting recorded data, see *Downloading, Uploading, and Exporting Replay Data* on page 69.

For information on the CSV file format that the sensor exports, see the next section.



	The tool supports data exported from Profile or Surface mode.

To get the utility package (), go to https://downloads.lmi3d.com/, choose your product from the Product Downloads section, and download it from the Download Center.

After downloading the tool package, unzip the file and run the Gocator CSV Converter tool [Tools > CSV Converter > kCsvConverter.exe].

The tool supports the following output formats:

Output formats

Format	Description
ASCII (XYZI)	Comma-separated points in X, Y, Z, Intensity (if available) format.
16-bit BMP	Heightmap with 16bit height values in a 5-5-5 RGB image. Not intended for visualization.

Format	Description
16-bit TIFF	Heightmap as grayscale image.
16-bit PNG	Heightmap as grayscale image.
GenTL RGB	For more information, see *16-bit RGB Image* on page 798
GenTL Mono	For more information, see *16-bit Grey Scale Image* on page 799.
Raw CSV	LMI Gocator CSV format for a single frame.
HexSight HIG	LMI HexSight heightmap.
STL ASCII	Mesh in standard STL text format (can become very large).
STL Binary	Mesh in binary STL format.
Wavefront OBJ	Mesh with comma-separated vertices and facets in text format.
ODSCAD OMC	ODSCAD heightmap.
MountainsMap SUR	DigitalSurf MountainsMap heightmap.
24-bit Spectrum	Color spectrum bitmap for visualization of heightmap. Does not contain height values.

With some formats, one or more of the following options are available:

Output options

Option	Description
Scale Z	Resamples the Z values to use the full value range.
Swap X/Y	Swaps the X and Y axes to obtain a right-handed coordinate system.
Keep Aspect Ratio	Resamples the X and Y axes to obtain the proper aspect ratio.

> The GenTL format is a 48-bit RGB or grey scale PNG. Height map, intensity and stamp information are stored as defined in the GenTL Driver section (*GenICam GenTL Driver* on page 794). You can load the exported data into image processing software to provide simulation data for developing applications using the GenTL driver.

To convert exported CSV into different formats:

1. Select the CSV file to convert in the **CSV** field.
2. (Optional) If intensity information is required, check the **Intensity** box and select the intensity bitmap. Intensity information is only used when converting to ASCII or GenTL format. If intensity is not selected, the ASCII format will only contain the point coordinates (XYZ).
3. If a dual-sensor system was used, choose the source sensor next to **Image**.
4. Select the output format.
   For more information on output formats, see *Output formats* on the previous page.

5. (Optional) Set the **Scale Z**, **Swap X/Y**, and **Keep Aspect Ratio** options.

   Availability of these options depends on the output format you have chosen. For more information, see *Output options* on the previous page.

6. Click **Convert**.

   The converter converts the input files.

   The converted file will be in the same directory as the input file. It will also have the same name as the input file but with a different file extension. The converted file name is displayed in the **Output File** field.

CSV File Format

The CSV Converter tool can convert from the CSV format that a sensor can export to several other formats. If you want to work with the exported file directly, use the following information.

An exported CSV file contains a series of "sections." Each section begins with a row containing the name of the section, and ends with a row containing the string "End." An empty line separates each section.

Each section usually contains one or more subsections. Each subsection has a header row containing a list of field names, followed by one or more rows of data. There is usually no empty line between the subsections.

Other structures within sections are possible.

Example:

```
Info
CSV Version,Sensor Count,Trigger Mode,...
2,1,0,32000.00000,...
End

DeviceInfo
ID,Model,Version,...
13434,311320-2M-01,4.8.2.29,...
End

Ranges
...
End
```

Usually all available data in the recording buffer is exported. The exceptions are Surface and SurfacePointCloud. For these sections, only the currently selected frame is exported.

Info

This section contains basic system information. It has one header row and one value row. The fields are described below:

Info Fields

Field	Description
CSV Version	Version of the CSV file format.
Sensor Count	Number of sensors in the system.
Trigger Mode	Trigger source: 0 – Time 1 – Encoder 2 – Digital input 3 – Software
Trigger Rate	Frame rate for time trigger (Hz).
Trigger Delay Domain	Output delay domain: 0 – Time (µs) 1 – Encoder (mm)
Trigger Delay	Output delay (µs or mm, depending on delay domain defined above).
Operation Mode	The scan mode.
XResolution	System X resolution (mm).
YResolution	System Y resolution (mm).
ZResolution	System Z resolution (mm).
Yspeed	Y Speed (mm/s).
Layout	Sensor orientation: 0 – Normal (single-sensor system) / Wide (dual-sensor system) 1 – Opposite 2 – Reverse 3 – Grid

DeviceInfo

This section contains information about each device in the system. There is one header row, and one value row per device.

DeviceInfo Fields

Field	Description
ID	Device serial number
Model	Device part number
Version	Firmware version
Exposure Mode	Exposure mode: 0 – Single exposure 1 – Multiple exposures 2 – Dynamic exposure
Exposure 0 through Exposure 4	Multiple exposures

Field	Description
Exposure Min	Dynamic exposure min
Exposure Max	Dynamic exposure max
FOV X	Active area X
FOV Y	Active area Y
FOV Z	Active area Z
FOV Width	Active area width
FOV Height	Active area length (Y). (Note difference in terminology.)
FOV Depth	Active area height (Z). (Note difference in terminology.)
Transform X	Transform X offset (mm)
Transform Y	Transform Y offset (mm)
Transform Z	Transform Z offset (mm)
Transform X Angle	Transform X Angle (degrees)
Transform Y Angle	Transform Y angle (degrees)
Transform Z Angle	Transform Z angle (degrees)

RecordingFilter

This section lists the filters used during recording. Unlike the other sections, it contains multiple sub-sections within, separated by spaces (but not the "End" keyword).

Example:

```
RecordingFilter
Section1 Param 1, Section1 Param2
value, value
Section2 Param 1
value
Section3 Param1, Section3 Param2
value
End
```

Each section will be described by a separate table below. They appear in the same order as documented.

RecordingFilter Fields

Field	Description
Condition Combination Type	Any or All

"Any Measurement" Filter Fields

Field	Description
Type	Any Measurement
Enabled	Whether or not is enabled. Yes/No
Result	Accepted result type: Pass/Fail/Invalid/Valid

"Any Data" Filter Fields

Field	Description
Type	Any Data
Enabled	Whether or not is enabled: Yes/No
Threshold Case	How to threshold: At or Above, or Below
Range Count Threshold	Threshold value (point count)

"Measurement" Filter Fields

Field	Description
Type	Measurement
Enabled	Whether or not is enabled: Yes/No
Result	Accepted result type: Pass/Fail/Invalid/Valid
Selection ID	First measurement ID

Ranges

This section describes single-point range data. It has two sub-sections: attributes and data.

The attribute section has only one row of data

Attribute Section Fields

Field	Description
Frame Count	Total number of frames
X Offset	X offset (mm)
Y Offset	Y offset (mm)
Z Offset	Z offset (mm)

The data section has one or more rows of data per frame (for example, range and intensity).

Data Section Fields

Field	Description
Frame	Frame index
Source	Source (for example, 0 for Top)
Time	Stamp time
Encoder	Stamp encoder
Z Encoder	Stamp encoder Z
Inputs	Stamp inputs
Exposure	Stamp exposure (us)
Y	Y value (mm)
Axis	Axis: Z (range) or I (Intensity)
Value	Range value (mm) or intensity (count)

Profile

This section describes uniform (or resampled) profile data, which is produced when the sensor is in

Profile mode and uniform spacing is enabled. It has two sub-sections: attributes and data.

The attribute section has only one row of data.

Attribute Section Fields

Field	Description
Frame Count	Total number of frames
Column Count	Number of columns
X Offset	X offset (mm)
Y Offset	Y offset (mm)
Z Offset	Z offset (mm)

The data section has one or more rows of data per frame (for example, range and intensity).

Data Section Fields

Field	Description
Frame	Frame index
Source	Source (for example, 0 for Top)
Time	Stamp time
Encoder	Stamp encoder
Z Encoder	Stamp encoder Z
Inputs	Stamp inputs
Exposure	Stamp exposure (µs)
Y	Y value (mm)
Axis	Axis: Z (range) or I (Intensity)
(x values)	Each column in header is a resampled X position Each column in data is the range (mm) or intensity (count)

RawProfile

This section describes point cloud profile data (or unresampled / raw data), which is produced when the sensor is in Profile mode and uniform spacing is disabled. It has two sub-sections: attributes and data.

The attribute section has only one row of data.

Attribute Section Fields

Field	Description
Frame Count	Total number of frames
Column Count	Number of columns
X Offset	X offset (mm)
Y Offset	Y offset (mm)
Z Offset	Z offset (mm)

The data section has one or more rows of data per frame (for example, range and intensity).

Data Section Fields

Field	Description
Frame	Frame index
Source	Source (for example, 0 for Top)
Time	Stamp time
Encoder	Stamp encoder
Z Encoder	Stamp encoder Z
Inputs	Stamp inputs
Exposure	Stamp exposure (µs)
Y	Y value (mm)
Axis	Axis: X, Z, or I (Intensity)
(x values)	Each column in header is an index. Each column in data is the X/Z value (mm) or intensity (count)

Part

This section describes uniform (or resampled) surface data, which is produced when the sensor is in Surface mode and uniform spacing is enabled.

> Only the data for the frame currently selected in the UI is exported when you export part data to a CSV file.

The section has two sub-sections: attributes and data.

The attribute section has only one row of data.

Attribute Section Fields

Field	Description
Frame	Frame index
Source	Source (for example, 0 for Top)
Time	Stamp time
Encoder	Stamp encoder
Z Encoder	Stamp encoder Z
Inputs	Stamp inputs
Row Count	Number of rows
Column Count	Number of columns
X Offset	X offset (mm)
Y Offset	Y offset (mm)
Z Offset	Z offset (mm)

The data section contains the data of a single surface scan. Each data row corresponds to one Y position. The first row contains the X values, and the first column contains the Y values. The region inside contains the range values (mm) for the corresponding row and column.

SurfacePointCloud

This section describes point cloud data (unresampled surface data), which is produced when the sensor is in Surface mode and uniform spacing is disabled. It has two sub-sections: attributes and data.

The attribute section has only one row of data.

Attribute Section Fields

Field	Description
Frame	Total number of frames
Source	Source (for example, 0 for Top)
Time	Stamp time
Encoder	Stamp encoder
Z Encoder	Stamp encoder Z
Inputs	Stamp inputs
Row Count	Number of rows
Column Count	Number of columns
X Offset	X offset (mm)
Y Offset	Y offset (mm)
Z Offset	Z offset (mm)

The data section contains the data of a single surface scan. The first row (header) can be ignored.

The data are (x,y,z) tuples expanded into a flat list of values, for example:

```
p0x, p0y, p0z, p1x, p1y, p1z, ..., pnx, pny, pnz
p(n+1)x, p(n+1)y, p(n+1)z, ...
...
```

Because the data has multiple rows and columns, it forms a rectangular grid of (x,y,z) tuples. The rows and columns do not correspond exactly to X and Y values, but suggest adjacency. i.e. positions with consecutive row indices or column indices are generally adjacent to each other in (x,y,z) coordinates.

The values are provided in mm, with the resolution and offset already calculated in the values.

Surface Section

This section describes surface section data, which is produced when a section is added to uniform surface data. A surface section is similar to a uniform profile.

The data section contains the following fields.

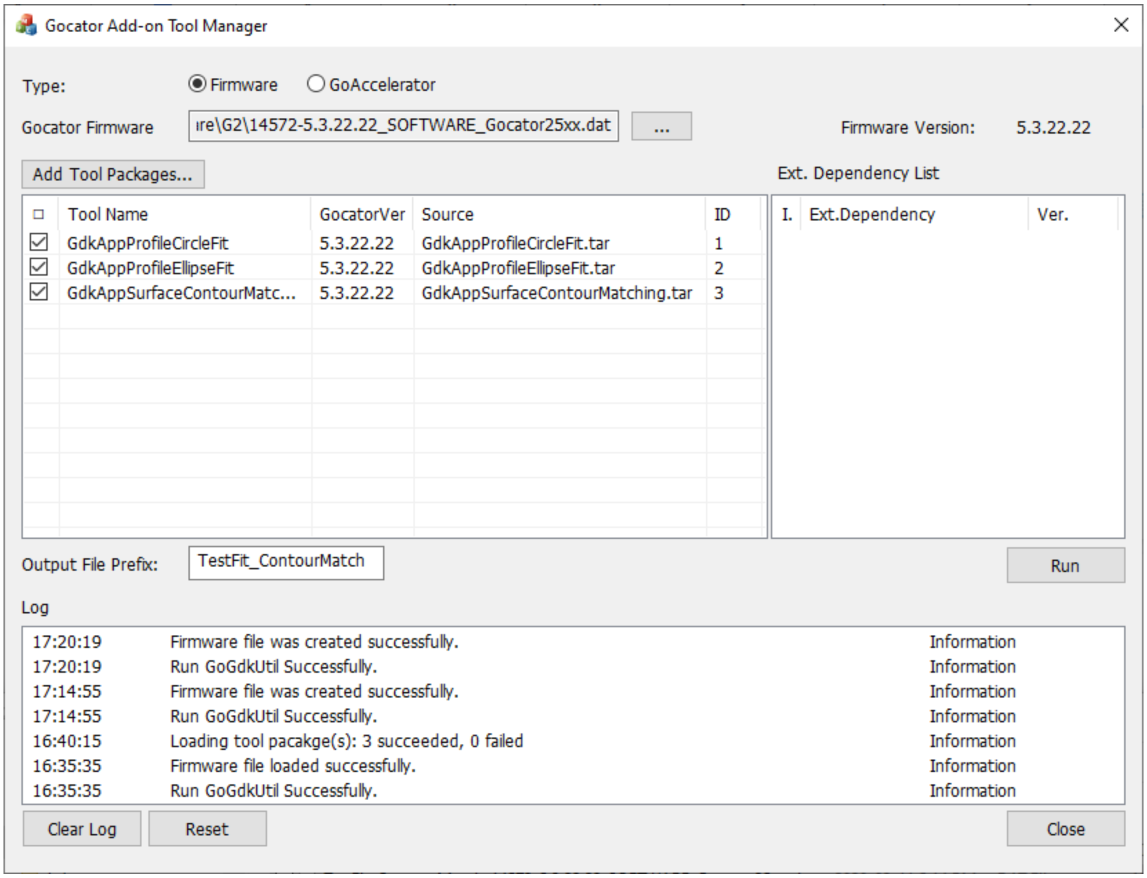
Data Section Fields

Field	Description
Frame	Frame Index
Source	Source (e.g. 0 for Top)
Time	Stamp time

Field	Description
Encoder	Stamp encoder
Z Encoder	Stamp encoder Z
Inputs	Stamp inputs
Exposure	Exposure
Column Count	Number of columns
Start X	X Start
Start Y	Y Start
End X	X End
End Y	Y End
Pose Angle	Pose Angle
Pose X	Pose X Offset
Pose Y	Pose Y Offset
X Offset	X Offset
Y Offset	Y Offset
Z Offset	Z Offset
XResolution	X Resolution
ZResolution	Y Resolution
Axis	Axis: Z (range) or I (Intensity)
(x values)	Each column in header is a resampled X position Each column in data is the range (mm) or intensity (count)

Gocator Add-on Tool Manager

The Gocator Add-on Tool Manager lets you quickly and easily add LMI-provided beta tools to an existing firmware file. This lets you try out upcoming tools and features before their public release. After updating the firmware, upload it to a compatible sensor; for more information, see *Firmware Upgrade* on page 83.



The tool manager is available in the 14577-x.x.xx.xx_SOFTWARE_AddOn_Beta_Tools.zip package, available in the Download Center (https://downloads.lmi3d.com/), under the *Beta Software Releases* section.

To get the package:

1. Go to https://downloads.lmi3d.com/ and log in to your account.
2. Select Gocator in the brand drop-down.
3. Choose the product family of your sensor in the next drop-down.
4. Click **Go**.
5. Scroll down to the *Beta Software Releases* section and expand it.
6. Choose the package that corresponds to the firmware or accelerator you wish to run the tool manager on and download it.

Remember that the tools available in the package are *beta* tools. LMI does not recommend using them in production settings.

The general workflow with the Gocator Add-on Tool Manager is as follows:

1. Load a .dat Gocator firmware file.
2. Load one or more beta tool packages (in .tar archives).
3. Enable the tools you want to add to the loaded firmware.

4. Run the tool manager on the firmware and tool packages.

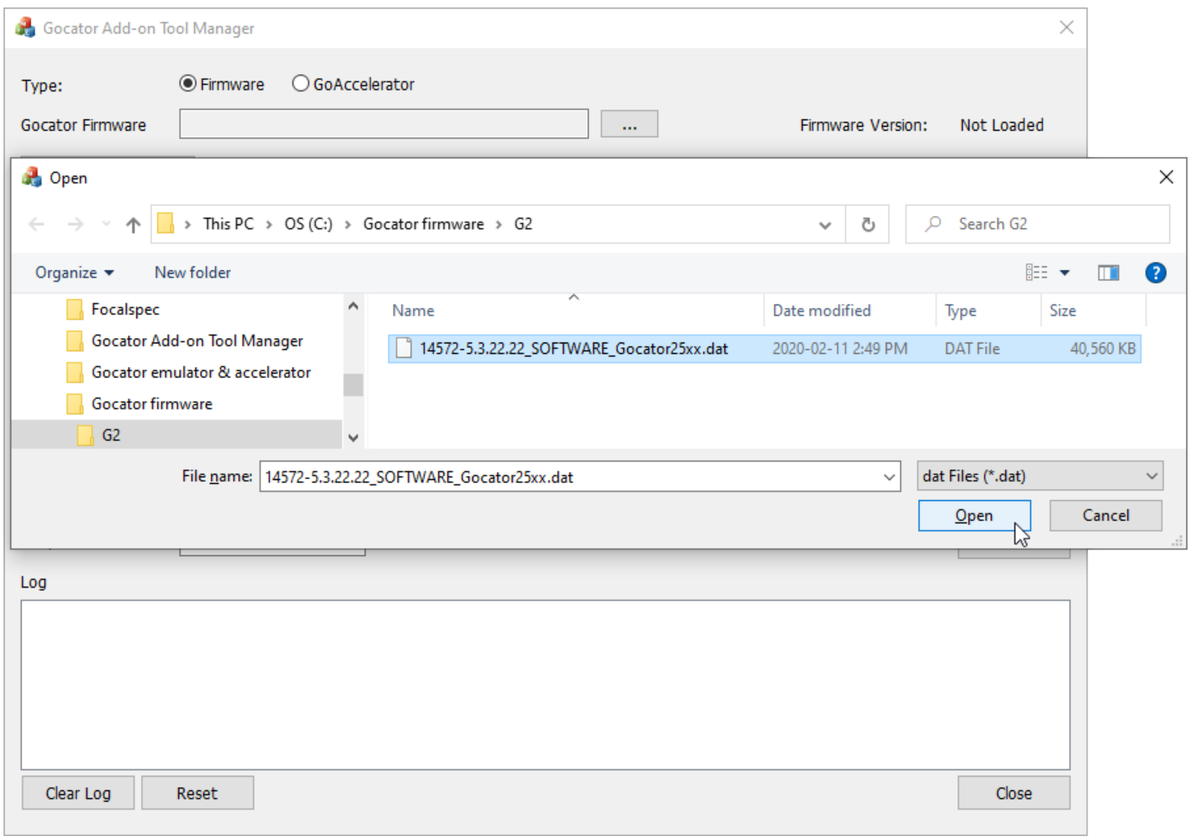
The result is a modified firmware file that contains the tools you selected.

At any point before creating the new firmware with the tool manager, you can click **Reset** to remove the loaded firmware and tool packages, and start over.

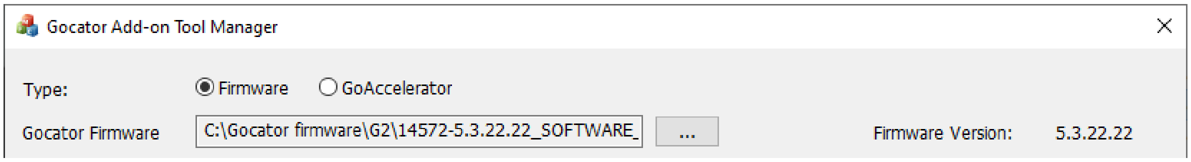
Adding Beta Tools to a Firmware

To add beta tools to a firmware:

1. If you haven't already done so, download and unzip the package containing the Gocator Add-on Tool Manager to a convenient location on your computer.
2. Launch the tool manager (GoAddOn_x.x.x.x.exe) from the \GoAddOn subfolder.
3. In the tool manager, choose **Firmware** next to **Type**.
4. Click the ... button next to the **Gocator Firmware** field.
5. In the Open dialog that displays, navigate to the location of the .dat firmware file you want to use and click **Open**.

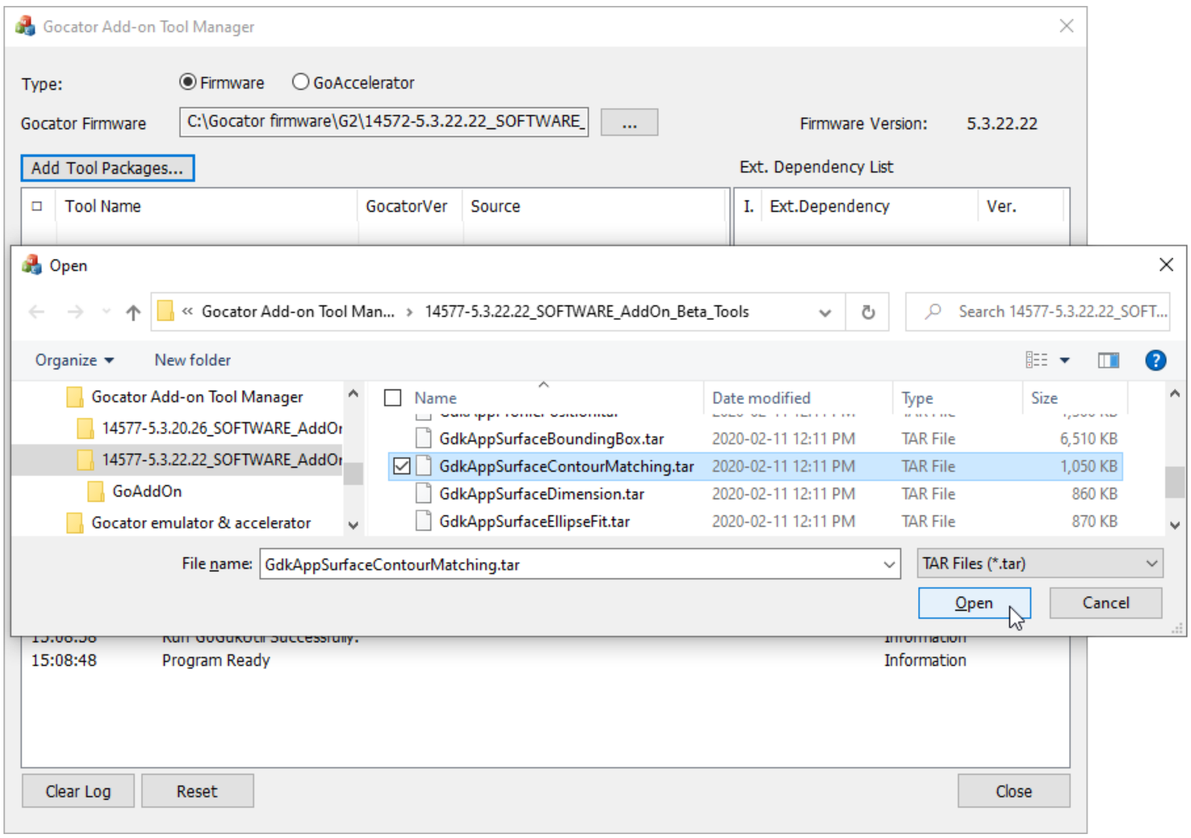


The tool manager loads the firmware. The firmware version is displayed in the upper right of the application.



If the firmware was previously created using the tool manager, the tools will be listed in the tool window. For information on removing tools, see *Removing Beta Tools from a Firmware* on page 876.

6. Click the **Add Tool Packages...** button above the tool list.
7. In the Open dialog, navigate to the folder containing the \GoAddOn folder, select a .tar tool package, and click **Open**.

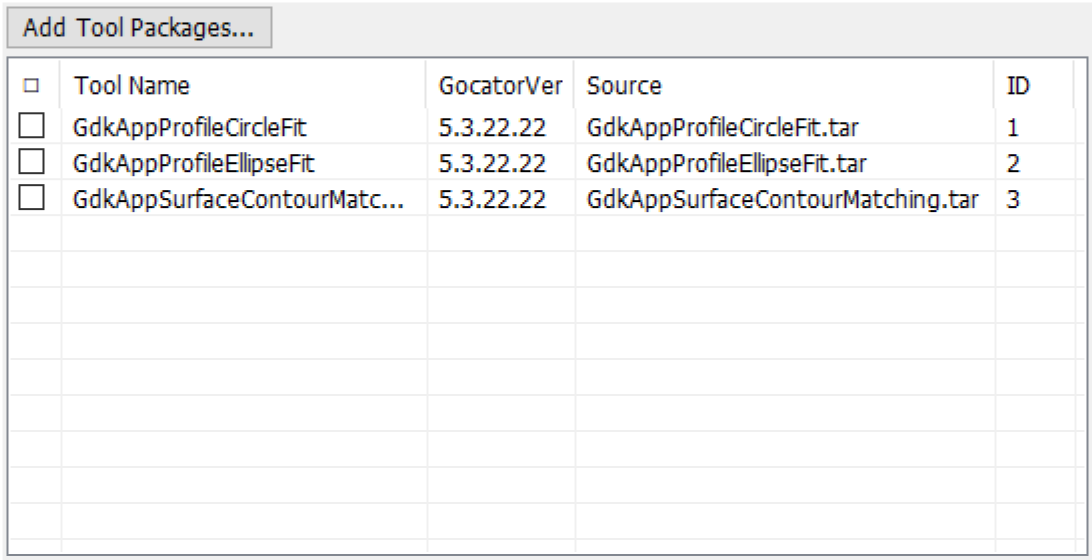


The tool manager adds the package to the tool list.

You can add multiple packages at once from the Open dialog.

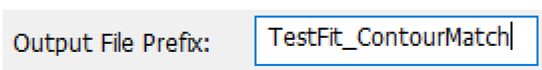
The version of a tool package must match the version of the loaded firmware. If a package's version does not match the firmware's, it is highlighted with grey in the tool list and can't be selected and included in the firmware. (The version of a tool package comes from the version of the GDK used to create it. It is not currently possible for users to create their own tool packages.).

8. When you have finished adding the packages, in the tool list, check the checkbox next the tools you want to add to the loaded firmware.

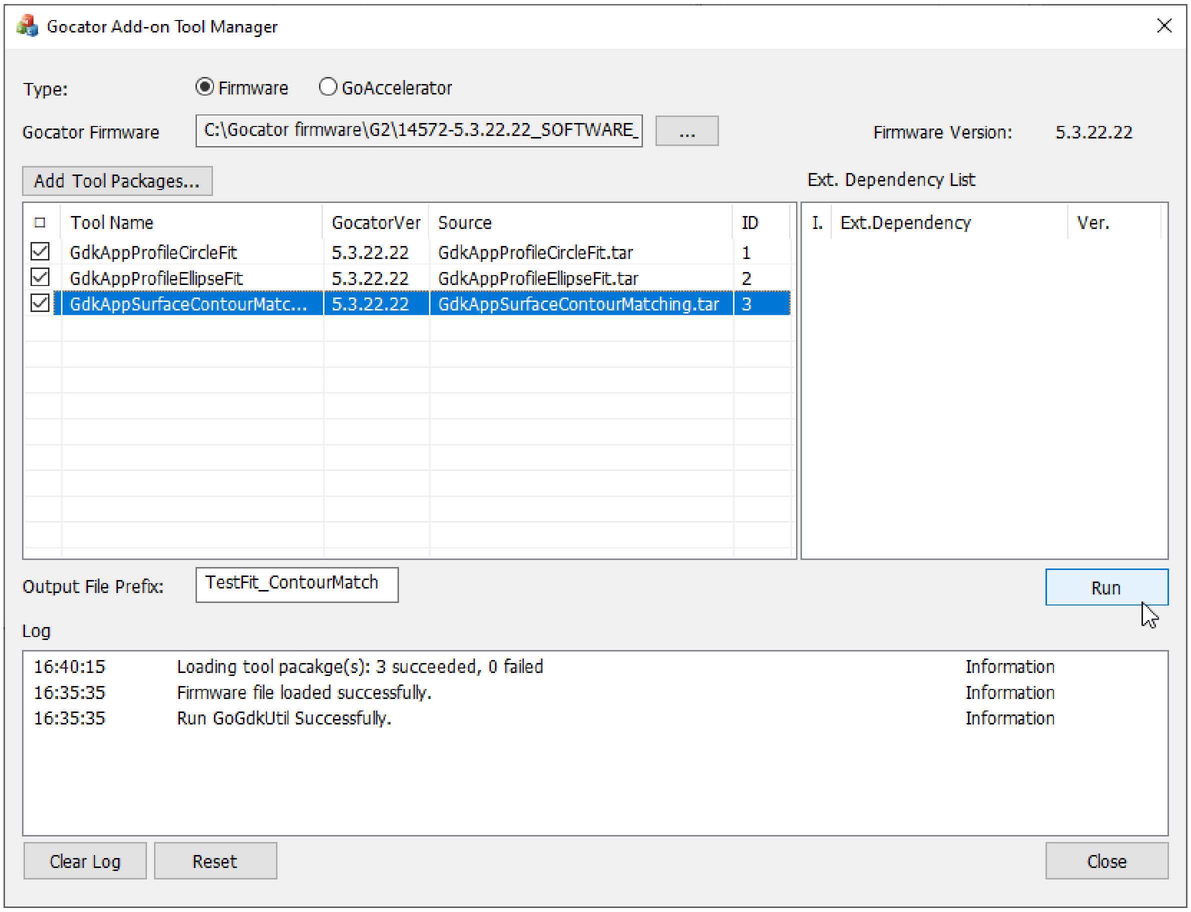


Be sure to select at least one tool.

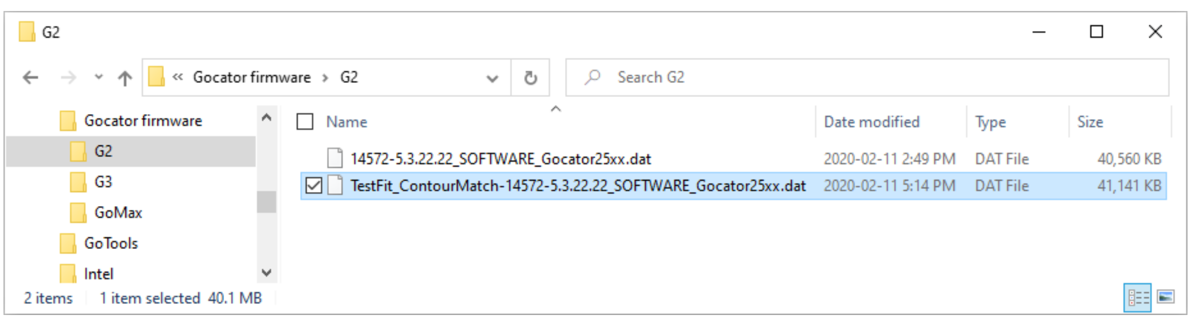
9. (Optional) In the **Output File Prefix** field, change the default "NewCustom" to something that will help you remember what the new firmware is for.



10. Click **Run**.



The tool manager creates a new firmware that contains the beta tools you selected, using the prefix you provided for the filename. The new firmware is created in the same location as the original firmware you loaded.



The tool manager will overwrite an existing firmware without warning.

After you have successfully created the new firmware, you can upload it to any compatible sensor; for more information, see *Firmware Upgrade* on page 83.

Removing Beta Tools from a Firmware

To remove a tool previously added to a firmware:

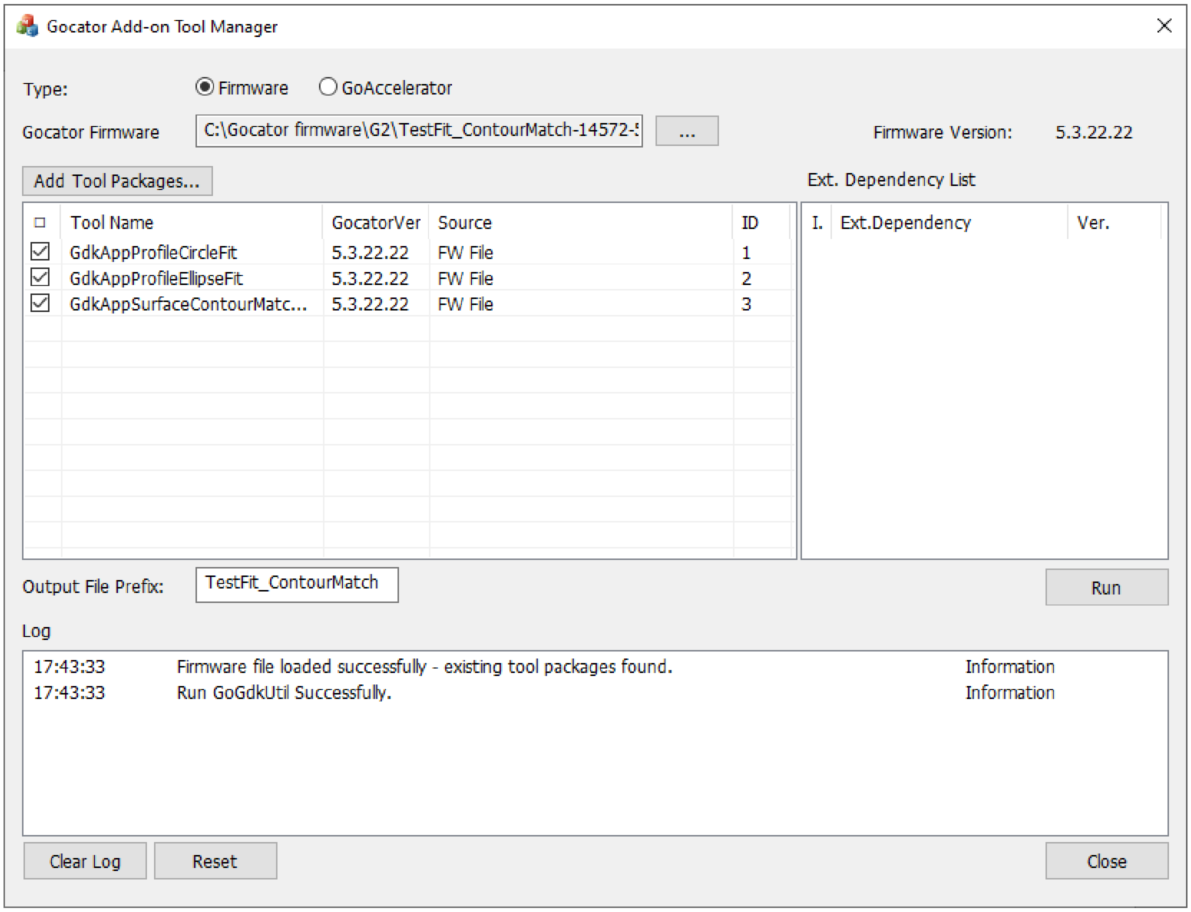
1. Launch the Gocator Add-on Tool Manager (GoAddOn_x.x.x.x.exe) from the \GoAddOn subfolder.

2. In the tool manager, choose **Firmware** next to **Type**.

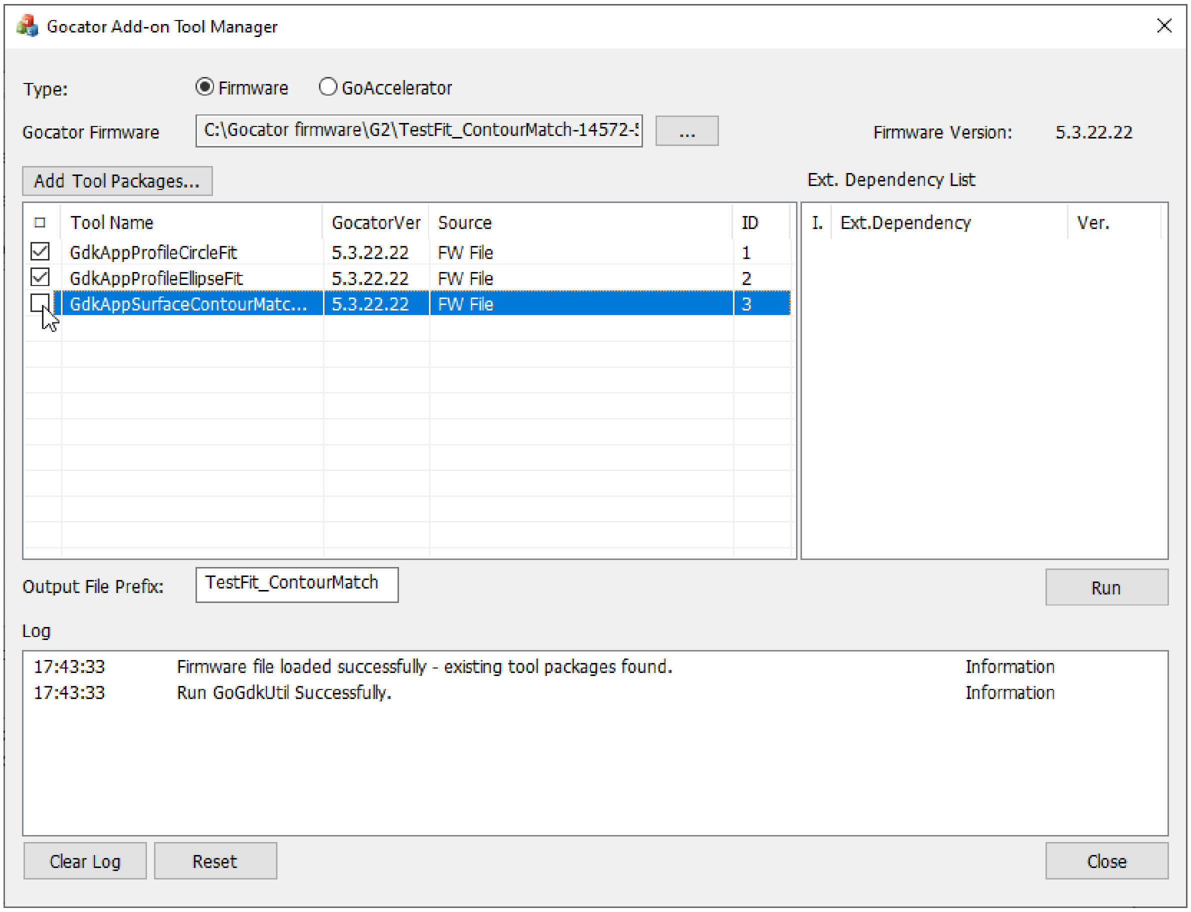
3. Click the  button next to the **Gocator Firmware** field.

4. In the Open dialog that displays, navigate to the location of the .dat firmware file from which you want to remove a tool.

   The tool manager loads the firmware and lists the previously added tools.



5. In the tool list, deselect the tools you want to remove from the firmware.



6. Click **Run**.

   The Gocator Add-on Tool Manager saves a new version of the firmware with the deselected tools removed.

> The tool manager overwrites existing firmware files with no warning.

After you have successfully created the new firmware, you can upload it to any compatible sensor; for more information, see *Firmware Upgrade* on page 83.

Troubleshooting

Review the guidance in this chapter if you are experiencing difficulty with a sensor system.

If the problem that you are experiencing is not described in this section, see *Return Policy* on page 921.

Mechanical/Environmental

The sensor is warm.

- It is normal for a sensor to be warm when powered on. A sensor is typically 15° C warmer than the ambient temperature.

Connection

When attempting to connect to the sensor with a web browser, the sensor is not found (page does not load).

- Verify that the sensor is powered on and connected to the client computer network. The Power Indicator LED should illuminate when the sensor is powered.
- Check that the client computer's network settings are properly configured.
- Use the Sensor Recovery tool to verify that the sensor has the correct network settings. See *Sensor Discovery Tool* on page 861 for more information.

When attempting to log in, the password is not accepted.

- Use the Sensor Recovery tool. See *Sensor Discovery Tool* on page 861 for steps to reset the password.

Performance

The sensor CPU level is near 100%.

- Consider reducing the speed. If you are using a time or encoder trigger source, see *Triggers* on page 89 for information on reducing the speed. If you are using an external input or software trigger, consider reducing the rate at which you apply triggers.
- Consider reducing the resolution.
  See *Spacing* on page 103 for more information on configuring resolution.
- Review the measurements that you have programmed and eliminate any unnecessary measurements.

Specifications

The following sections describe the specifications of Gocator sensors and connectors, as well as Master hubs.

Sensors

The following sections provide the specifications of Gocator sensors.

Gocator 3210 Sensor

The Gocator 3210 is defined below:

MODEL	3210
Scan Rate (Hz)	6
Imagers (megapixels)	2
Clearance Distance (CD) (mm)	164.0
Measurement Range (MR) (mm)	110.0
Field of View (mm)	71.0 x 98.0 - 100.0 x 154.0
Repeatability Z (µm)	4.7
Resolution XY (mm)	0.060 - 0.090
VDI/VDE Accuracy (mm)*	0.035
Dimensions (mm)	49 x 146 x 190
Weight (kg)	1.7
Light Source	Blue LED (465 nm)
Interface	Gigabit Ethernet
Inputs	Differential Encoder, Trigger
Outputs	2x Digital output, RS-485 Serial (115 kBaud) Analog Output (4 – 20 mA)
Input Voltage (Power)	+24 to +48 VDC (50 Watts); Ripple +/- 10%
Housing	Gasketed aluminum enclosure, IP67
Operating Temp.	0 to 45° C
Storage Temp.	-30 to 70° C
Vibration Resistance	10 to 55 Hz, 1.5 mm double amplitude in X, Y and Z directions, 2 hours per direction
Shock Resistance	15 g, half sine wave, 11 ms, positive and negative for X, Y and Z directions

* Based on 2634, Part 2.

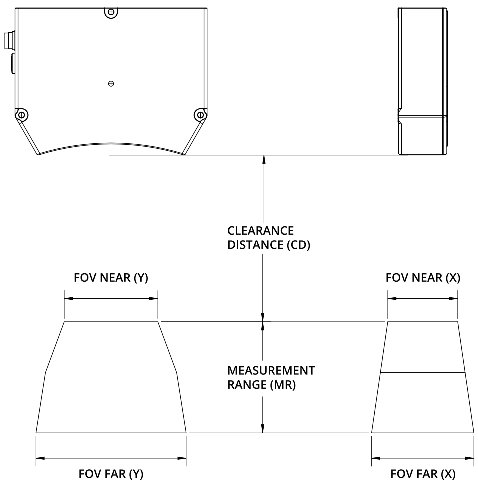
Linearity Z values and Resolution Z are typical values.

Field of View and Resolution XY are specified as [X] x [Y], near to far.

Differential encoder requires the use of Master 400/800/810/1200/2400/2410.

For details on scan rates, see *Estimated Performance and Scan Rates* on page 535.

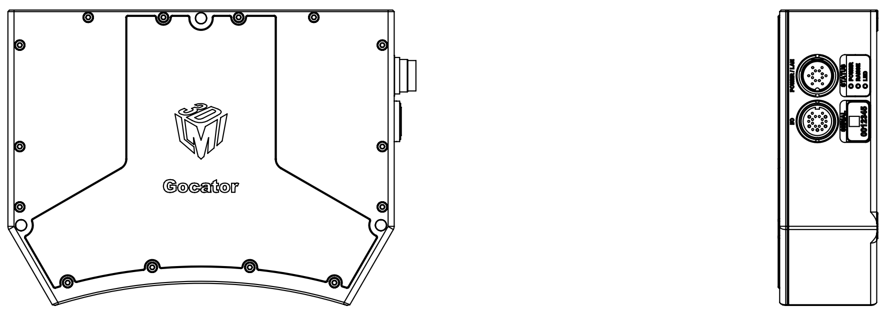
The following diagram illustrates some of the terms used in the table above.

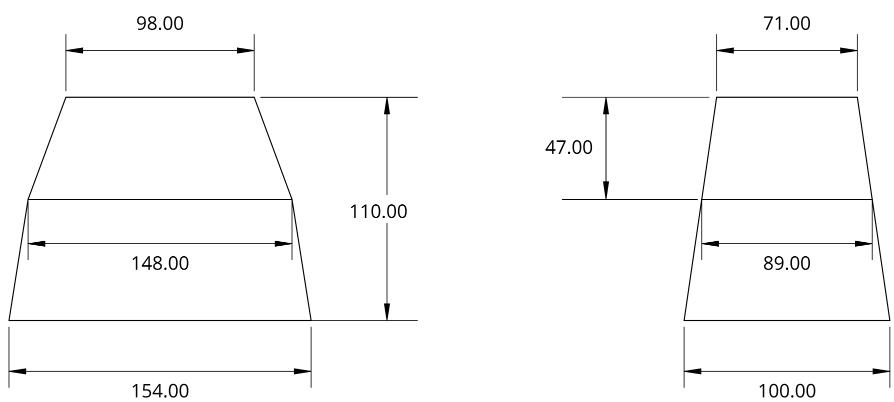


Mechanical dimensions, CD/FOV/MR, and the envelope for each sensor model are illustrated on the following pages.

Gocator 3210

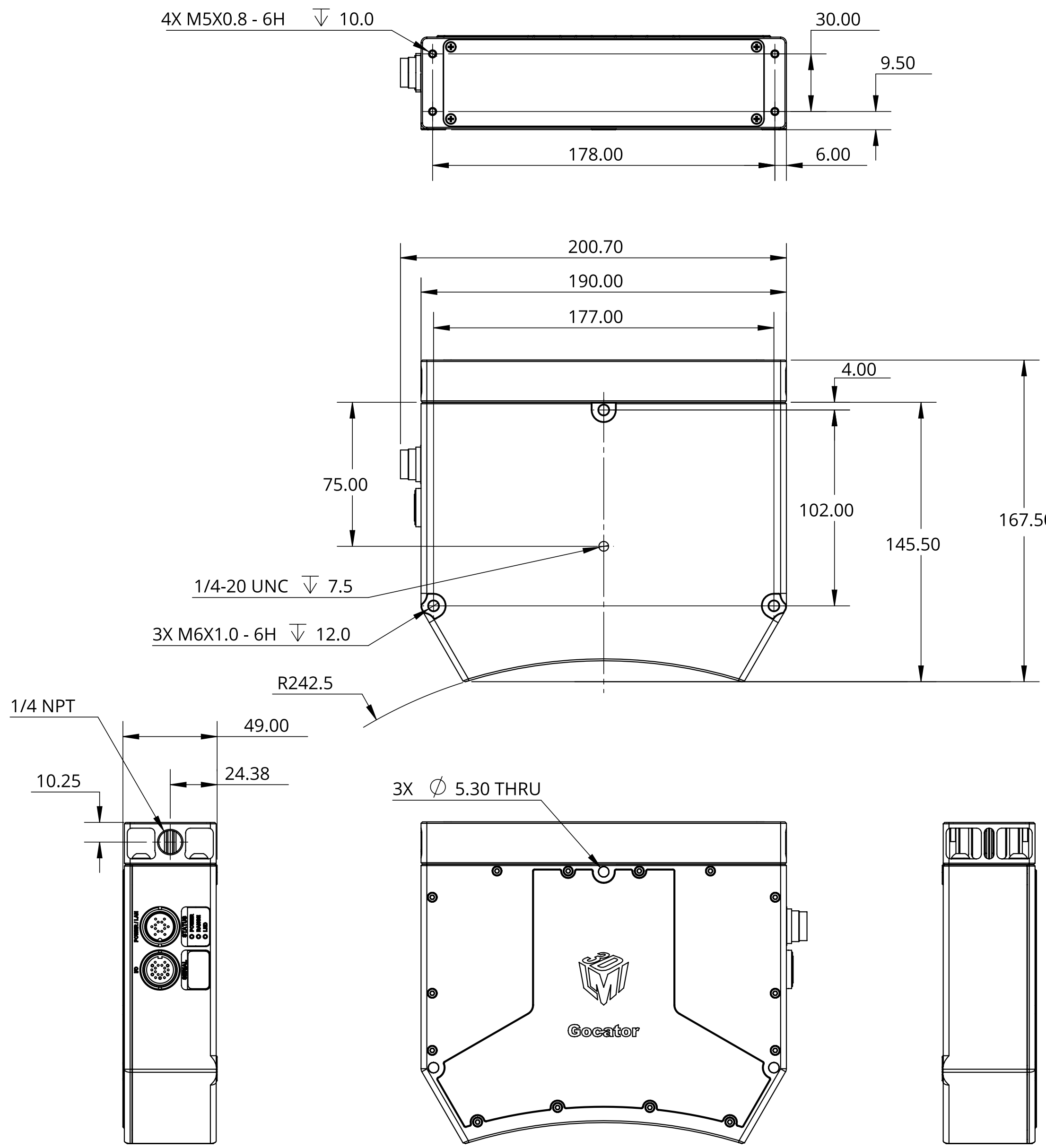
Field of View / Measurement Range / Coordinate System Orientation



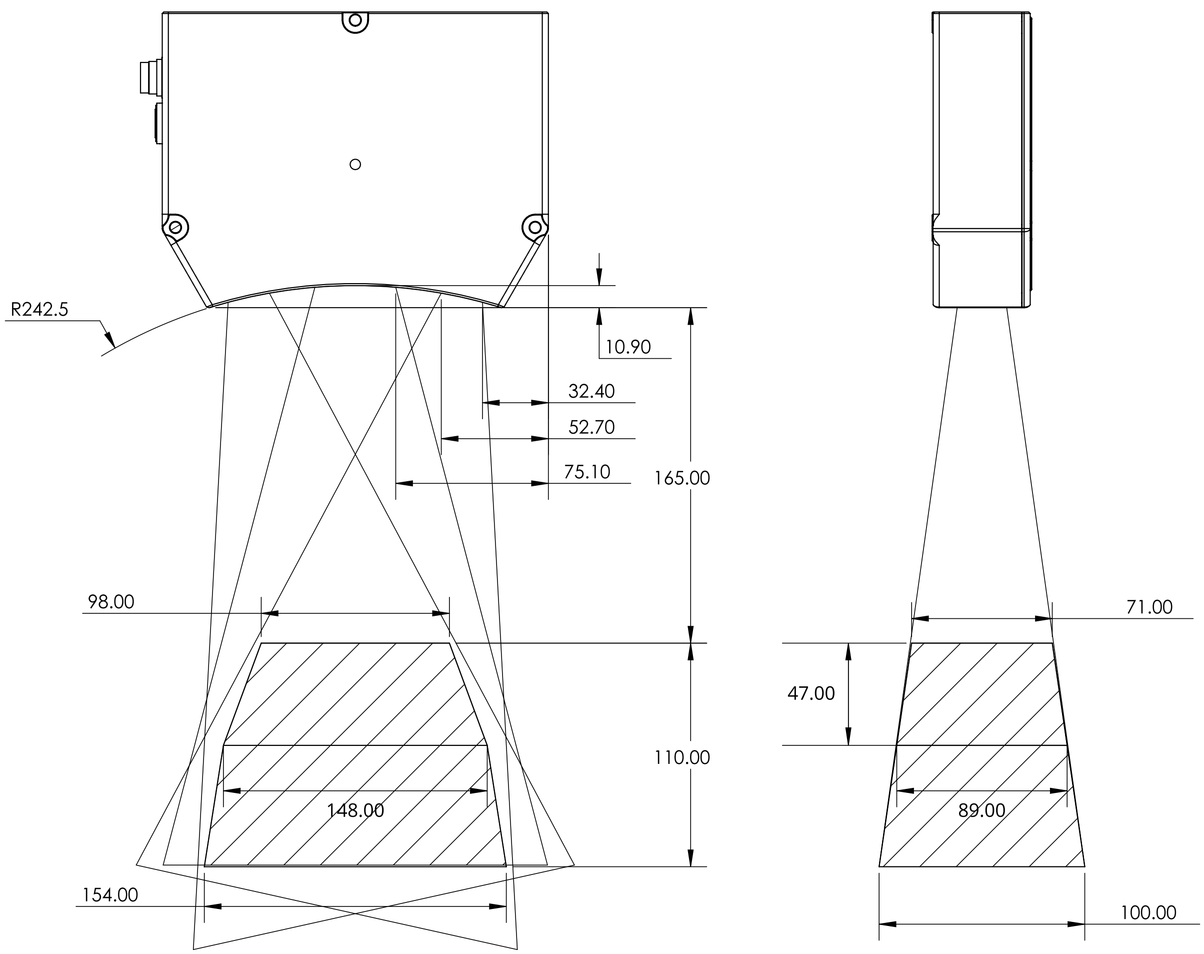


Dimensions

The following drawings include the optional shop air heat sink. For more information, see *APPNOTE_Gocator_3210_Heatsink.pdf*, available on LMI's website.


4X M5X0.8 - 6H ↧ 10.0
30.00
9.50
178.00
6.00
200.70
190.00
177.00
4.00
75.00
102.00
145.50
167.50
1/4-20 UNC ↧ 7.5
3X M6X1.0 - 6H ↧ 12.0
R242.5
1/4 NPT
49.00
24.38
10.25
3X ∅ 5.30 THRU
Gocator

Envelope



R242.5
10.90
32.40
52.70
75.10
165.00
98.00
110.00
148.00
154.00
71.00
47.00
89.00
100.00

Gocator 3500 Series

The Gocator 3500 series consists of the following models:

MODEL	3504	3506
Scan Rate (Hz)	6	6
Imagers (megapixels)	5	5
Clearance Distance (CD) (mm)	52.5	87.0
Measurement Range (MR) (mm)	7	25.0
Field of View (mm)	12.1 x 13.2 (near) 12.7 x 16.4 (maxY) 13.0 x 15.0 (far)	27.0 x 45.0 (near) 30.0 x 45.0 (far)
Repeatability Z (µm)	0.2	2.0
Resolution XY (µm)	6.7 - 7.1	20 - 25
VDI/VDE Accuracy (mm)*	n/a	0.012
Accuracy XYZ (µm)**	6	n/a
Dimensions (mm)	49x152x177.5	49 x 136 x 170
Weight (kg)	1.77	1.52

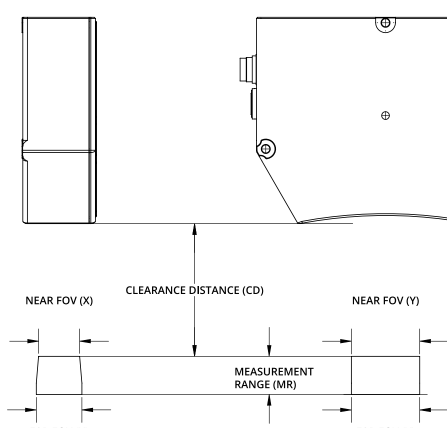
* Based on 2634, Part 2.

** Based on sphere-fitting at various positions in the scan volume.

Field of View and Resolution XY are specified as [X] x [Y], near to far.

For details on scan rates, see *Estimated Performance and Scan Rates* on page 535.

The following diagram illustrates some of the terms used in the table above.

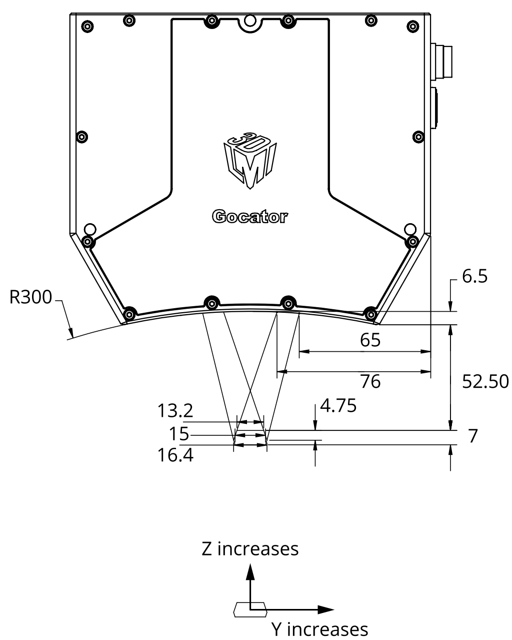


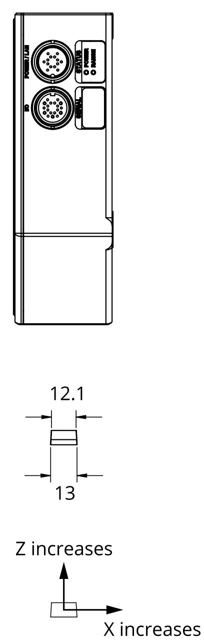
ALL 3500 SERIES MODELS	
Light Source	Blue LED (465 nm)
Interface	Gigabit Ethernet
Inputs	Differential Encoder, Trigger
Outputs	2x Digital output, RS-485 Serial (115 kBaud) Analog Output (4 – 20 mA)
Input Voltage (Power)	+24 to +48 VDC (25 Watts); Ripple +/- 10%
Housing	Gasketed aluminum enclosure, IP67
Operating Temp.	0 to 50° C
Storage Temp.	-30 to 70° C
Vibration Resistance	10 to 55 Hz, 1.5 mm double amplitude in X, Y and Z directions, 2 hours per direction
Shock Resistance	15 g, half sine wave, 11 ms, positive and negative for X, Y and Z directions

Mechanical dimensions, CD/FOV/MR, and the envelope for each sensor model are illustrated on the following pages.

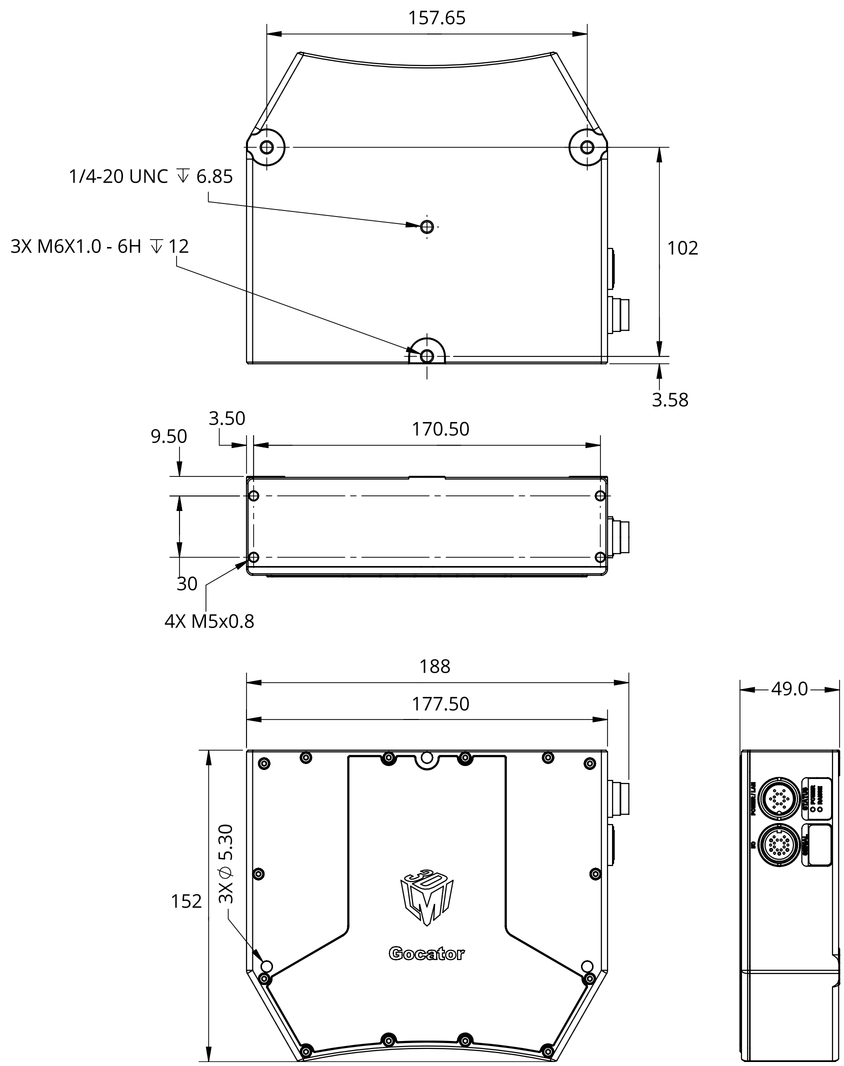
Gocator 3504

Field of View / Envelope / Measurement Range / Coordinate System Orientation





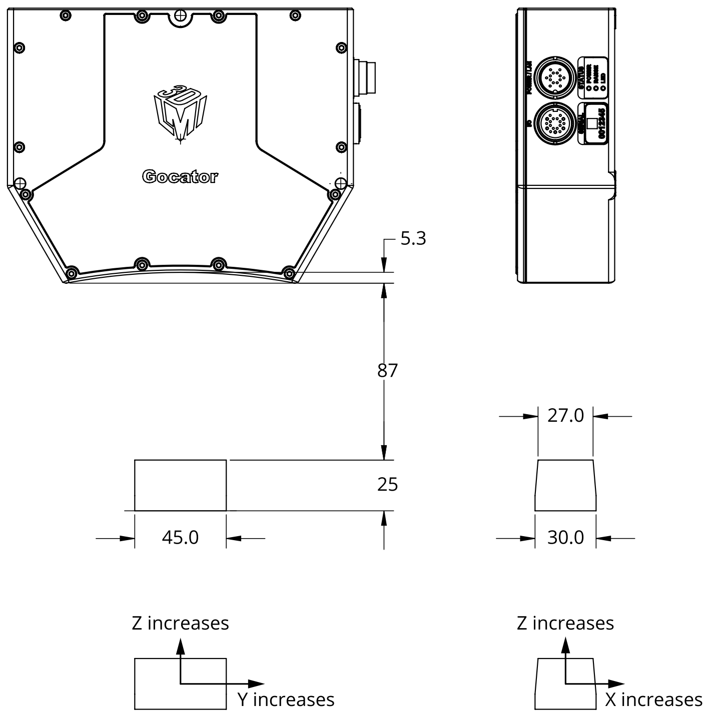
Dimensions



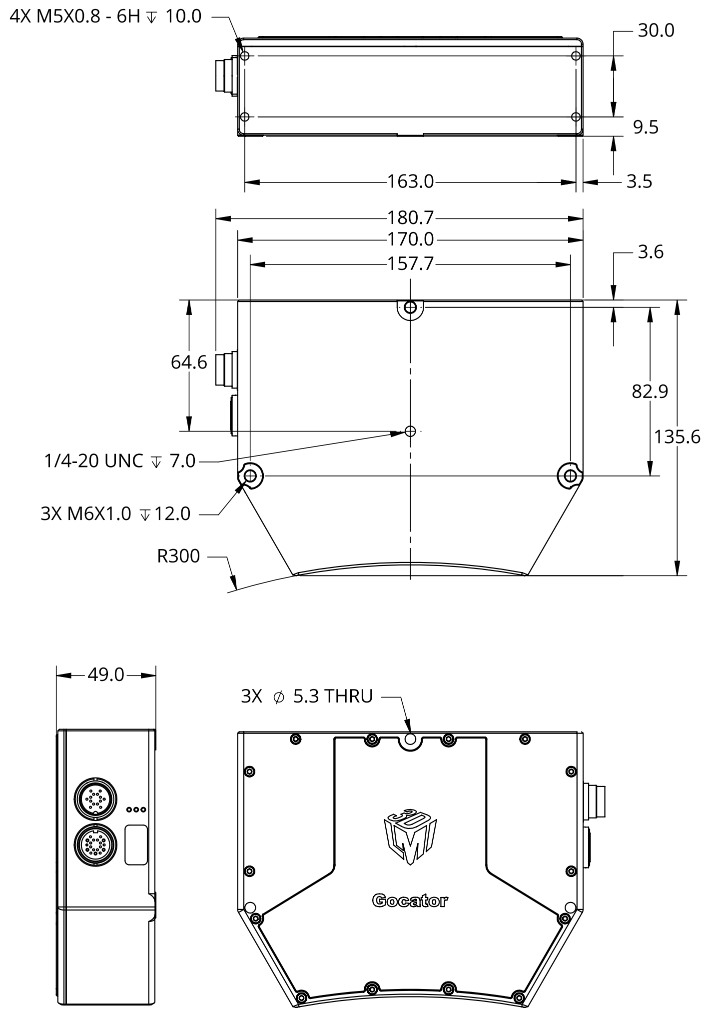
157.65
1/4-20 UNC ↧ 6.85
3X M6X1.0 - 6H ↧ 12
102
3.58
3.50
9.50
170.50
30
4X M5x0.8
188
177.50
49.0
152
3X ⌀ 5.30
Gocator

Gocator 3506

Field of View / Measurement Range / Coordinate System Orientation

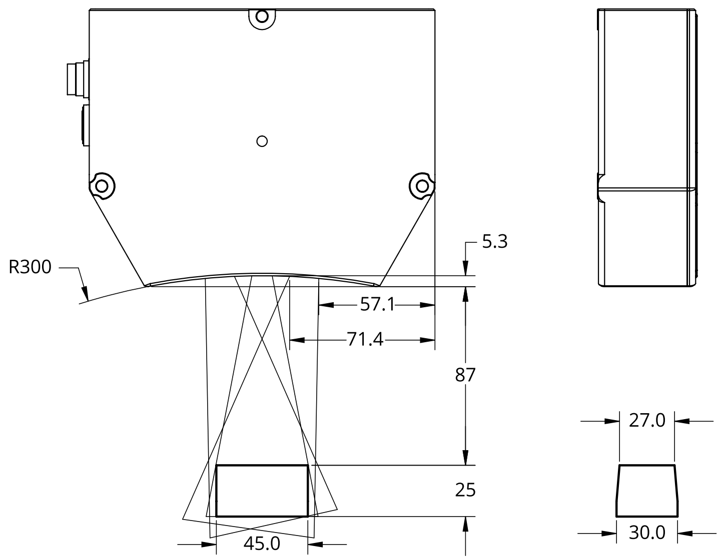


Dimensions



4X M5X0.8 - 6H ⌴ 10.0
30.0
9.5
163.0
3.5
180.7
170.0
157.7
3.6
64.6
82.9
135.6
1/4-20 UNC ⌴ 7.0
3X M6X1.0 ⌴12.0
R300
49.0
3X ⌀ 5.3 THRU
Gocator

Envelope



Sensor Connectors

The following sections provide the specifications of the connectors on Gocator sensors.

Gocator Power/LAN Connector

The Gocator Power/LAN connector is a 14 pin, M16 style connector that provides power input and Ethernet.

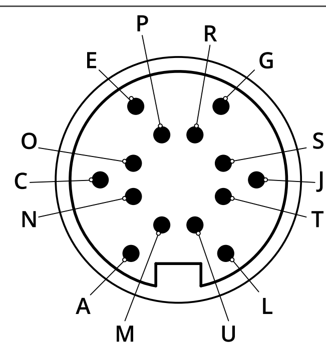
This connector is rated IP67 only when a cable is connected or when a protective cap is used.

Some sensors require a minimum input voltage of 48 VDC. Verify the accepted input voltage for your sensor in the sensor's specifications; for specifications, see *Sensors* on page 880.

This section defines the electrical specifications for Power/LAN Connector pins, organized by function.

Gocator Power/LAN Connector Pins

Function	Pin	Lead Color on Standard Cordsets	Lead Color on High Flex Cordsets
GND_24-48V	L	White/Orange & Black	Orange/Red
GND_24-48V	L	Orange/Black	Orange/Black
DC_24-48V	A	White/Green & Black	Green/Red
DC_24-48V	A	Green/Black	Green/Black
Reserved	G	White/Blue & Black	Blue/Black
Reserved	J	Blue/Black	Blue/Red
Sync+*	E	White/Brown & Black	Brown/Red
Sync-*	C	Brown/Black	Brown/Black
Ethernet MX1+	M	White/Orange	White/Orange
Ethernet MX1-	N	Orange	Orange
Ethernet MX2+	O	White/Green	White/Green
Ethernet MX2-	P	Green	Green
Ethernet MX3-	S	White/Blue	White/Blue
Ethernet MX3+	R	Blue	Blue
Ethernet MX4+	T	White/Brown	White/Brown



*View: Looking into the connector **on** the sensor*

Function	Pin	Lead Color on Standard Cordsets	Lead Color on High Flex Cordsets
Ethernet MX4-	U	Brown	Brown

Two wires are connected to the ground and power pins.

* The Sync leads are not connected in the open wire versions of the Power/LAN cordsets.

Grounding Shield

The grounding shield should be mounted to the earth ground.

Power

Apply positive voltage to DC_24-48V.

> Some sensors require a minimum input voltage of 48 VDC. Verify the accepted input voltage for your sensor in the sensor's specifications; for specifications, see *Sensors* on page 880.

Power requirements

Function	Pins	Min	Max
DC_24-48V	A	24 V (Some models require a minimum of 48 V.)	48 V
GND_24-48VDC	L	0 V	0 V

Safety Input

With snapshot sensors, the Safety_in+ and Safety_in- signals do not need to be connected (to a voltage source) in order to scan with these models.

Gocator I/O Connector

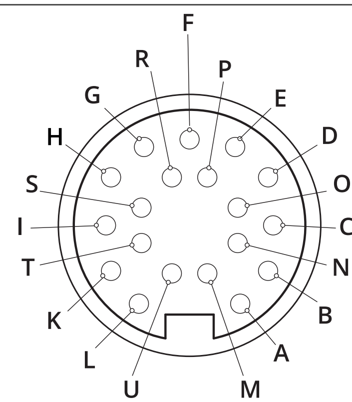
The Gocator I/O connector is a 19 pin, M16 style connector that provides encoder, digital input, digital outputs, serial output, and analog output signals.

This connector is rated IP67 only when a cable is connected or when a protective cap is used.

This section defines the electrical specifications for I/O connector pins, organized by function.

Gocator I/O Connector Pins

Function	Pin	Lead Color on Standard Cordset	Lead Color on High Flex Cordset
Trigger_in+	D	Grey	Blue / Red
Trigger_in-	H	Pink	Blue / Black
Out_1+ (Digital Output 0)	N	Red	Brown / Red
Out_1- (Digital Output 0)	O	Blue	Brown / Black
Out_2+ (Digital Output 1)	S	Tan	Green / Red
Out_2- (Digital Output 1)	T	Orange	Green / Black
Encoder_A+	M	White / Brown & Black	Pink / Red
Encoder_A-	U	Brown / Black	Pink / Black
Encoder_B+	I	Black	Yellow / Red
Encoder_B-	K	Violet	Yellow / Black
Encoder_Z+	A	White / Green & Black	White / Red
Encoder_Z-	L	Green / Black	White / Black
Serial_out+	B	White	Purple / Red
Serial_out-	C	Brown	Purple / Black
Reserved	E	Blue / Black	Red
Reserved	G	White / Blue & Black	Black
Analog_out+ (Reserved on Gocator 2500 series sensors)	P	Green	Gray / Red
Analog_out- (Reserved on Gocator 2500 series sensors)	F	Yellow & Maroon / White	Gray / Black & Orange / Black
Reserved	R	Maroon (not connected)	Orange / Red (not connected)



*View: Looking into the connector **on** the sensor*

Grounding Shield

The grounding shield should be mounted to the earth ground.

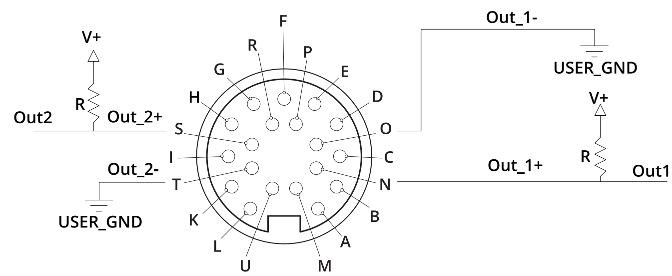
Digital Outputs

Each sensor has two optically isolated outputs. Both outputs are open collector and open emitter, which allows a variety of power sources to be connected and a variety of signal configurations.

> Digital outputs cannot be used when taking scans using the Snapshot button, which takes a single scan and is typically used to test measurement tool settings. Digital outputs can only be used when a sensor is running, taking a continuous series of scans.

Out_1 (Collector – Pin N and Emitter – Pin O) and Out_2 (Collector – Pin S and Emitter – Pin T) are independent and therefore V+ and GND are not required to be the same.

Function	Pins	Max Collector Current	Max Collector–Emitter Voltage	Min Pulse Width
Out_1	N, O	40 mA	70 V	20 µs
Out_2	S, T	40 mA	70 V	20 µs

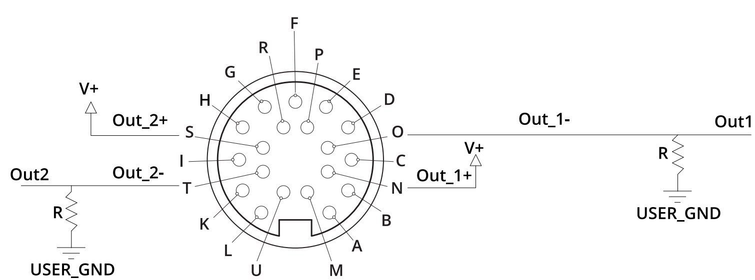


The resistors shown above are calculated by R = (V+) / 2.5 mA.

The size of the resistors is determined by power = (V+)^2 / R.

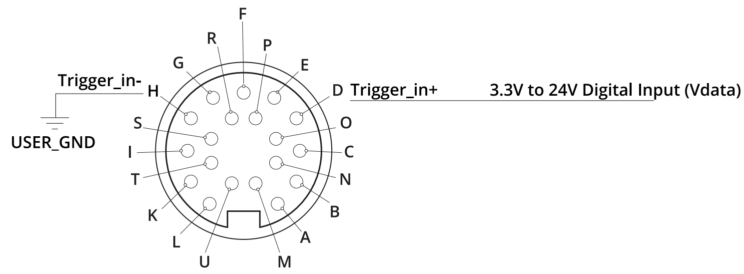
Inverting Outputs

To invert an output, connect a resistor between ground and Out_1- or Out_2- and connect Out_1+ or Out_2+ to the supply voltage. Take the output at Out_1- or Out_2-. For resistor selection, see above.



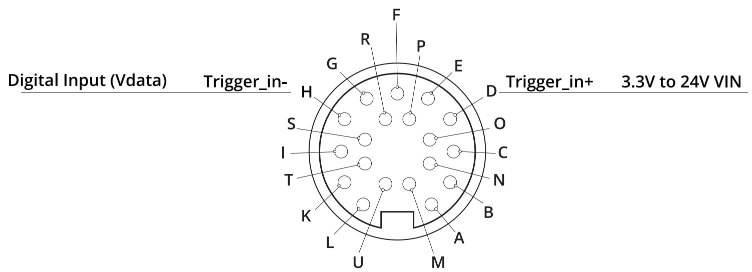
Digital Input

Every sensor has a single optically isolated input. To use this input without an external resistor, supply 3.3 - 24 V to the positive pin and GND to the negative.



Active High

If the supplied voltage is greater than 24 V, connect an external resistor in series to the positive. The resistor value should be R = [(Vin-1.2V)/10mA]-680.



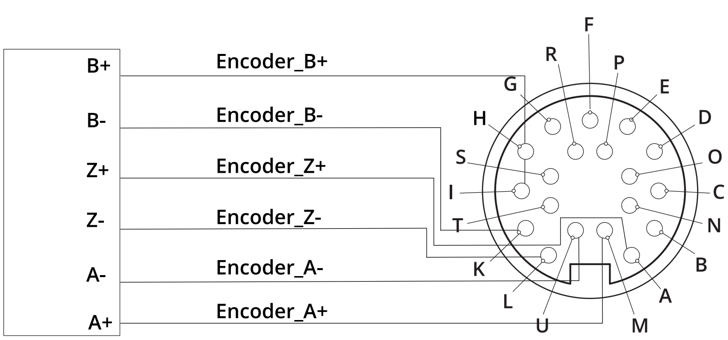
Active Low

To assert the signal, the digital input voltage should be set to draw a current of 3 mA to 40 mA from the positive pin. The current that passes through the positive pin is I = (Vin – 1.2 – Vdata) / 680. To reduce noise sensitivity, we recommend leaving a 20% margin for current variation (i.e., uses a digital input voltage that draws 4mA to 25mA).

Function	Pins	Min Voltage	Max Voltage	Min Current	Max Current	Min Pulse Width
Trigger_in	D, H	3.3 V	24 V	3 mA	40 mA	20 µs

Encoder Input

Encoder input is provided by an external encoder and consists of three RS-485 signals. These signals are connected to Encoder_A, Encoder_B, and Encoder_Z.



Function	Pins	Common Mode Voltage		Differential Threshold Voltage			Max Data Rate
		Min	Max	Min	Typ	Max	
Encoder_A	M, U	-7 V	12 V	-200 mV	-125 mV	-50 mV	1 MHz
Encoder_B	I, K	-7 V	12 V	-200 mV	-125 mV	-50 mV	1 MHz
Encoder_Z	A, L	-7 V	12 V	-200 mV	-125 mV	-50 mV	1 MHz

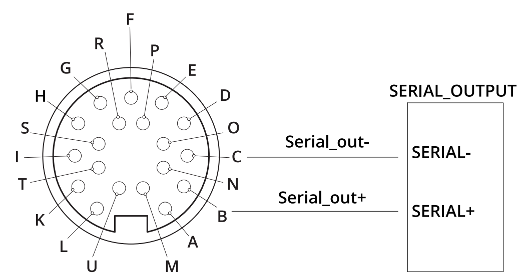
Gocator only supports differential RS485 signalling. Both + and - signals must be connected.

Encoders are normally specified in *pulses* per revolution, where each pulse is made up of the four quadrature *signals* (A+ / A- / B+ / B-). Because the sensor reads each of the four quadrature signals, you should choose an encoder accordingly, given the resolution required for your application.

Serial Output

Serial RS-485 output is connected to Serial_out as shown below.

Function	Pins
Serial_out	B, C

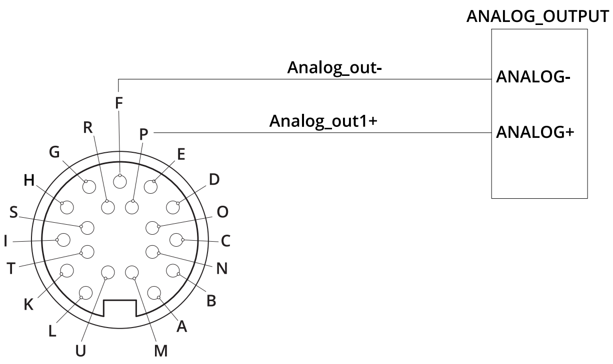


Analog Output

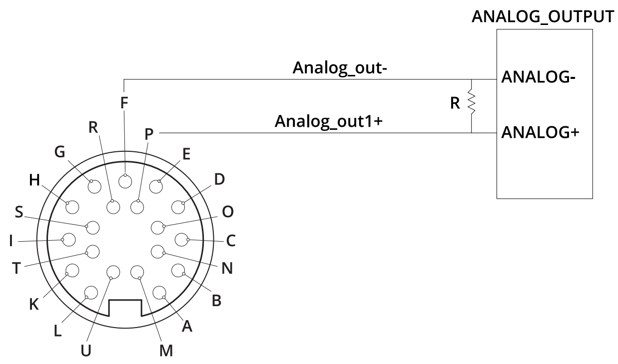
The Sensor I/O Connector defines one analog output interface: Analog_out.

You do not need to supply an external power source.

Function	Pins	Current Range
Analog_out	P, F	4 – 20 mA

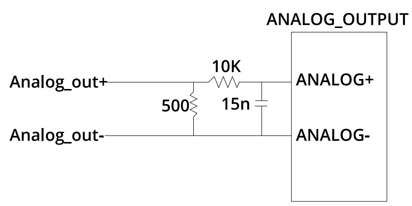


Current Mode



Voltage Mode

To configure for voltage output, connect a 500 Ohm ¼ Watt resistor between Analog_out+ and Analog_out- and measure the voltage across the resistor. To reduce the noise in the output, we recommend using an RC filter as shown below.



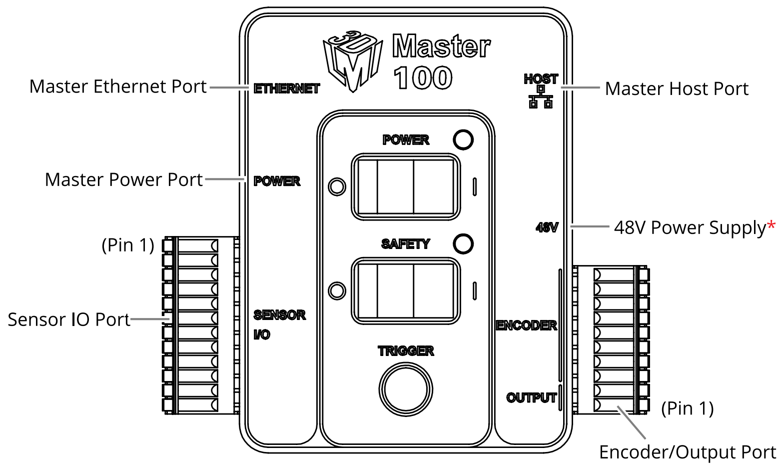
Master Network Controllers

The following sections provide the specifications of Master network controllers.

For information on maximum external input trigger rates, see *Maximum Input Trigger Rate* on page 92.

Master 100

The Master 100 accepts connections for power, safety (not supported by snapshot sensors), and encoder, and provides digital output.



*Contact LMI for information regarding this type of power supply.

Connect the Master Power port to the Gocator's Power/LAN connector using the Gocator Power/LAN to Master cordset. Connect power RJ45 end of the cordset to the Master Power port. The Ethernet RJ45 end of the cordset can be connected directly to the Ethernet switch, or connect to the Master Ethernet port. If the Master Ethernet port is used, connect the Master Host port to the Ethernet switch with a CAT5e Ethernet cable.

To use encoder and digital output, wire the Master's Gocator Sensor I/O port to the Gocator IO connector using the Gocator I/O cordset.

Sensor I/O Port Pins

I/O Pin	Master Pin	Conductor Color
Encoder_A+	1	White/Brown & Black
Encoder_A-	2	Brown/Black
Encoder_Z+	3	White/Green & Black
Encoder_Z-	4	Green/Black
Trigger_in+	5	Grey
Trigger_in-	6	Pink
Out_1-	7	Blue
Out_1+	8	Red

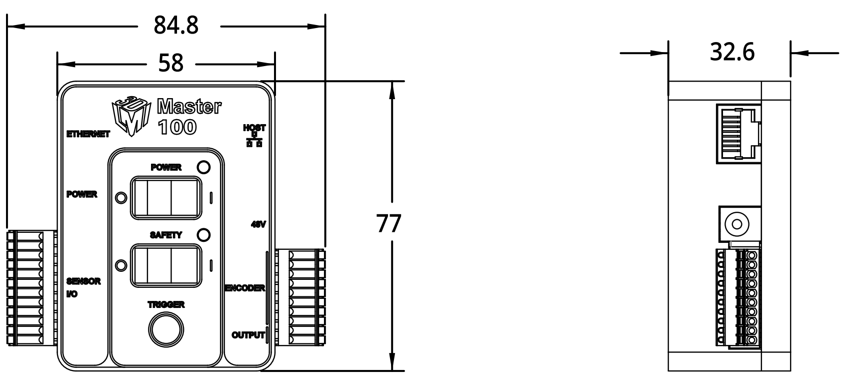
I/O Pin	Master Pin	Conductor Color
Encoder_B+	11	Black
Encoder_B-	12	Violet

The rest of the wires in the I/O cordset are not used.

Encoder/Output Port Pins

Function	Pin
Output_1+ (Digital Output 0)	1
Output_1- (Digital Output 0)	2
Encoder_Z+	3
Encoder_Z-	4
Encoder_A+	5
Encoder_A-	6
Encoder_B+	7
Encoder_B-	8
Encoder_GND	9
Encoder_5V	10

Master 100 Dimensions

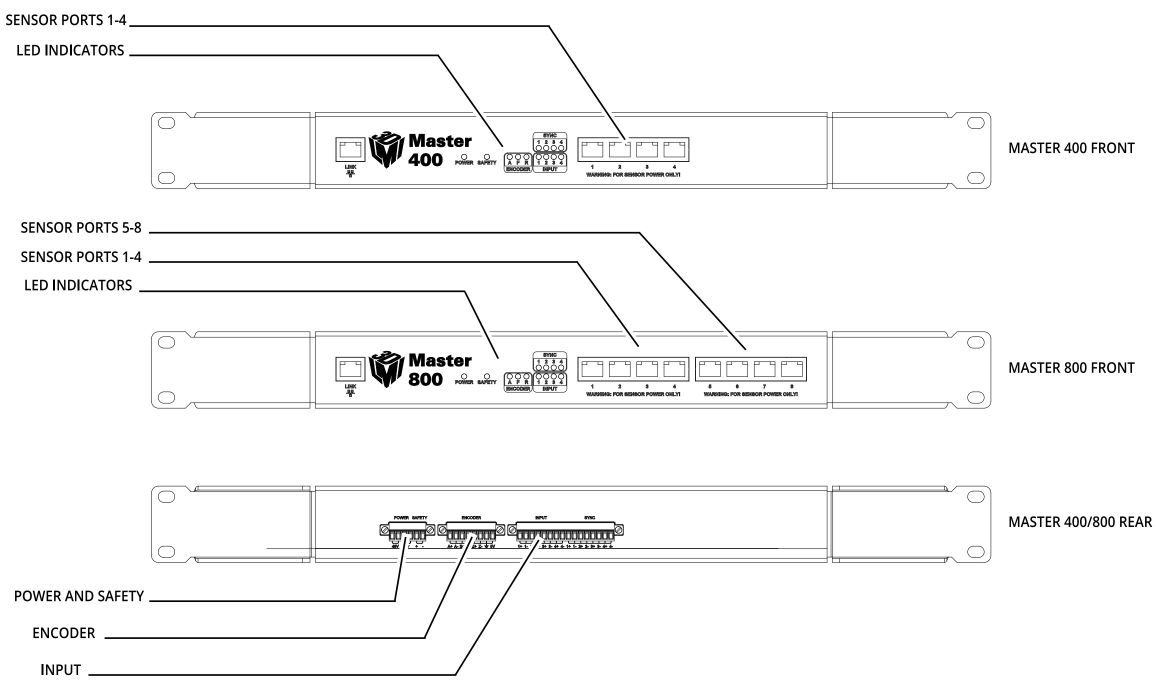


Master 400/800

Master network controllers provide sensor power and safety interlock, and broadcast system-wide synchronization information (i.e., time, encoder count, encoder index, and digital I/O states) to all devices on a sensor network.

	Safety interlock is not supported by snapshot sensors.

	The Phoenix connectors on Master 400/800/1200/2400 are not compatible with the connectors on Master 810/2410. For this reason, if you are switching models in your network, you must rewire the connections to the Master.



Power and Safety (6 pin connector)

Function	Pin
+48VDC	1
+48VDC	2
GND (24-48VDC)	3
GND (24-48VDC)	4
Reserved	5
Reserved	6

	The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

Input (16 pin connector)

Function	Pin
Input 1	1
Input 1 GND	2
Reserved	3
Reserved	4
Reserved	5
Reserved	6
Reserved	7
Reserved	8
Reserved	9
Reserved	10
Reserved	11
Reserved	12
Reserved	13
Reserved	14
Reserved	15
Reserved	16

The Input connector does not need to be wired up for proper operation.

Encoder (8 pin connector)

Function	Pin
Encoder_A+	1
Encoder_A-	2
Encoder_B+	3
Encoder_B-	4
Encoder_Z+	5
Encoder_Z-	6
GND	7
+5VDC	8

Master 400/800 Electrical Specifications

Electrical Specifications

Specification	Value
Power Supply Voltage	+48 VDC
Power Supply Current (Max.)	10 A

Specification	Value
Power Draw (Min.)	5.76 W
Encoder Signal Voltage	Differential (5 VDC)
Digital Input Voltage Range	Logical LOW: 0 to +0.1 VDC Logical HIGH: +3.3 to +24 VDC

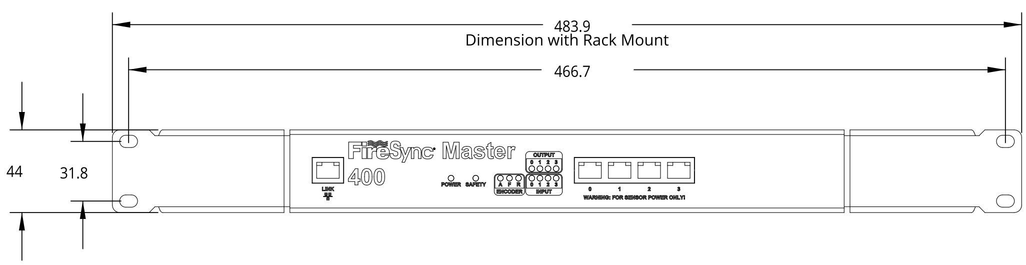
⚠	When using a Master hub, the chassis must be well grounded.

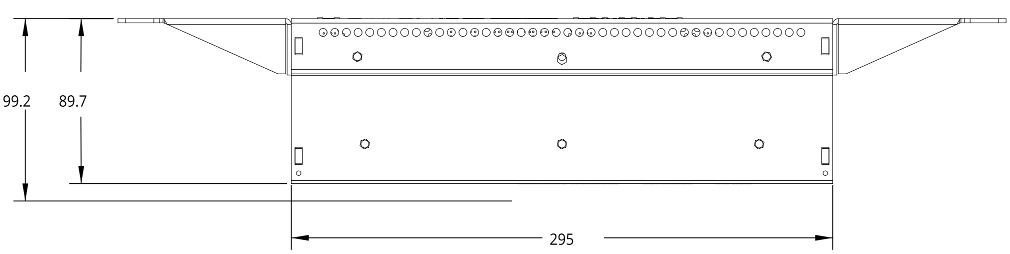
🗋	The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

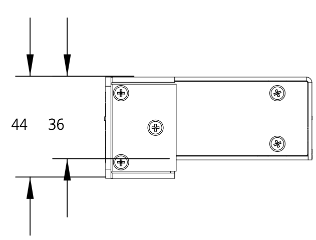
🗋	The Power Draw specification is based on a Master with no sensors attached. Every sensor has its own power requirements that need to be considered when calculating total system power requirements..

Master 400/800 Dimensions

The dimensions of Master 400 and Master 800 are the same.







Master 810/2410

Master network controllers provide sensor power and safety interlock, and broadcast system-wide synchronization information (i.e., time, encoder count, encoder index, and digital I/O states) to all devices on a sensor network.

> Safety interlock is not supported by snapshot sensors.

> Some sensors require a minimum input voltage of 48 VDC. Verify the accepted input voltage for your sensor in the sensor's specifications; for specifications, see *Sensors* on page 880.

The following table summarizes Master 810 and 2410:

Master 810 and 2410	
Input Voltage (Power)	+24-48 VDC (2 Watts)[1]
Total Power	Master 810 / 2410 input power + (sensor input power x number of sensors)
Capacity	Master 810: up to 8 sensors Master 2410: up to 24 sensors
I/O	4 digital inputs[2] Single-Ended Active LOW: 0 to +0.8 VDC Single-Ended Active HIGH: +3.3 to +24 VDC Differential LOW: 0.8 to -24 VDC Differential HIGH: +3.3 to +24 VDC 10-pin Phoenix For more information, see *Electrical Specifications* on page 908.
Encoder	Differential (5 VDC, 12 VDC) Single-ended (5 VDC, 12 VDC)[3] For more information, see *Electrical Specifications* on page 908.
LED Indicators	Safety, power, encoder, input. For more information, see *LED Indicators* on the next page.
Cable	Dual CAT5e cable for power / safety / synchronization / data
Weight (kg)	Master 810: 0.6 Master 2410: 0.9

Notes

1. Refer to sensor datasheets for additional power required by sensors.
2. Gocator only supports one digital input.
3. Supports open collector, pull-up resistor, line driver, push-pull, and TTL.

The following table describes the meanings of the encoder and sensor port LED indicators:

LED Indicators

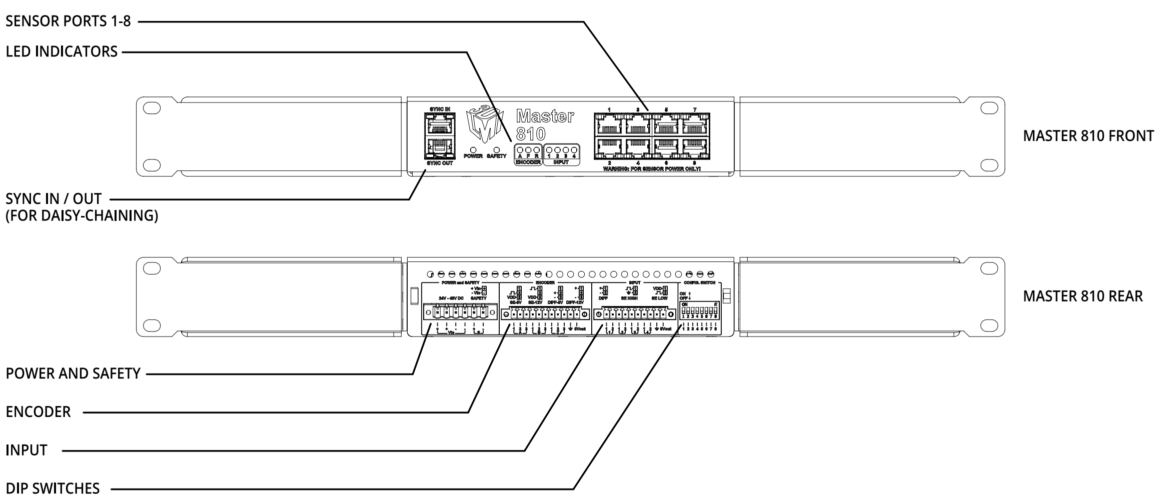
Indicator	Description
Power	Device is on.
Safety	Indicates the status of the Safety Interlock circuitry. The "On" state indicates that all sensor light sources are active.
Encoder A	Reserved
Encoder F	**On continuously**: Forward motion with no indexing is detected. **Blinking**: Forward motion with indexing is detected.
Encoder R	**On continuously**: Forward motion with no indexing is detected. **Blinking**: Forward motion with indexing is detected.
Input 1	Digital input port 1 active.
SYNC IN and SYNC OUT Ports (Green and Orange LEDs)	Reserved.
Sensor Port Green LED	Indicates that a sensor is connected to the port and is powered up.
Sensor Port Orange LED	Not used.

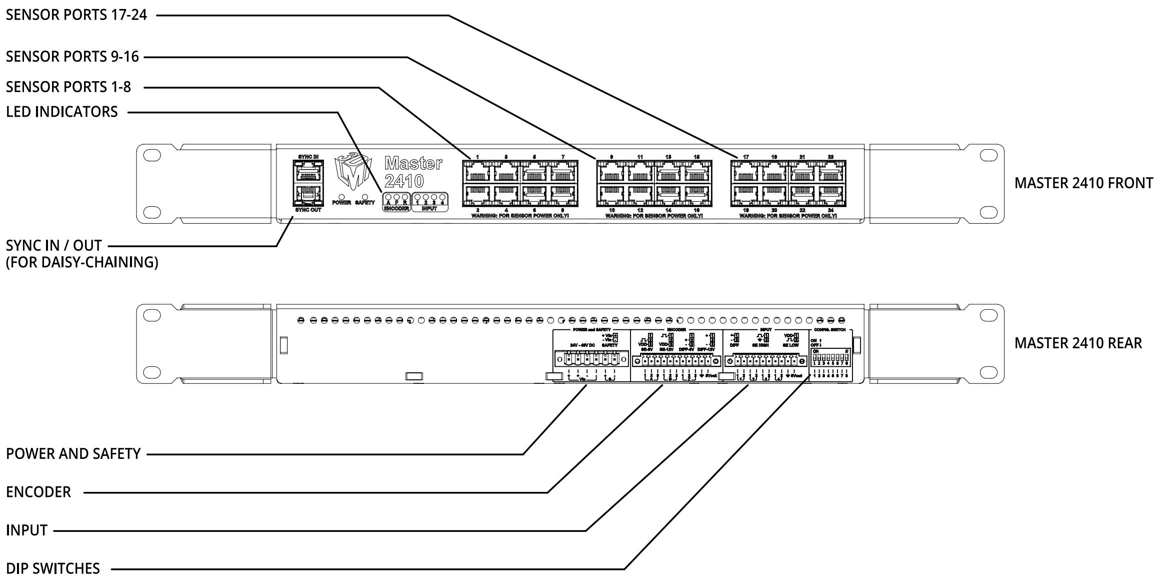
Master 810 and 2410 can be mounted to DIN rails using the provided adapters (for more information, see *Installing DIN Rail Clips: Master 810 or 2410* on page 30). The units are also provided with removable adapters for 1U rack mounting; the mounting holes for this option are compatible with older Master models (400/800/1200/2400).

> The Phoenix connectors on Master 400/800/1200/2400 are not compatible with the connectors on Master 810/2410. For this reason, if you are switching models in your network, you must rewire the connections to the Master.

Master 2410 can currently be used with encoders with a maximum quadrature frequency of 300 kHz.

Master 810 can be configured to work with a maximum encoder quadrature frequency of 6.5 MHz. For more information, see *Configuring Master 810* on page 32.





Power and Safety (6 pin connector)

Function	Pin
Power In+	1
Power In+	2
Power In-	3
Power In-	4
Reserved	5
Reserved	6

The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

On earlier revisions of Master 810 and Master 2410, the inputs are labeled 0-3.

Input (10 pin connector)

Function	Pin
Input 1 Pin 1	1
Input 1 Pin 2	2
Reserved	3
Reserved	4
Reserved	5
Reserved	6
Reserved	7
Reserved	8
GND (output for powering other devices)	9
+5VDC (output for powering other devices)	10

The Input connector does not need to be wired up for proper operation.

For Input connection wiring options, see *Input* on page 911.

Encoder (11 pin connector)

Function	Pin
Encoder_A_Pin_1	1
Encoder_A_Pin_2	2
Encoder_A_Pin_3	3
Encoder_B_Pin_1	4
Encoder_B_Pin_2	5
Encoder_B_Pin_3	6
Encoder_Z_Pin_1	7
Encoder_Z_Pin_2	8
Encoder_Z_Pin_3	9
GND (output for powering external devices)	10
+5VDC (output for powering external devices)	11

For Encoder connection wiring options, see *Encoder* on the next page.

Electrical Specifications

Some sensors require a minimum input voltage of 48 VDC. Verify the accepted input voltage for your sensor in the sensor's specifications; for specifications, see*Sensors* on page 880

Electrical Specifications

Specification	Value
Power Supply Voltage	+24 VDC to +48 VDC
Power Supply Current (Max.)*	Master 810: 9 A Master 2410: 25 A * Fully loaded with 1 A per sensor port.
Power Draw (Min.)	Master 810: 1.7 W Master 2410: 4.8 W
Encoder Signal Voltage	Single-Ended Active LOW: 0 to +0.8 VDC Single-Ended Active HIGH: +3.3 to +24 VDC Differential LOW: 0.8 to -24 VDC Differential HIGH: +3.3 to +24 VDC For more information, see *Encoder* on the next page.
Digital Input Voltage Range	Single-Ended Active LOW: 0 to +0.8 VDC Single-Ended Active HIGH: +3.3 to +24 VDC

Specification	Value
	Differential LOW: 0.8 to -24 VDC Differential HIGH: +3.3 to +24 VDC For more information, see *Input* on page 911. If the input voltage is above 24 V, use an external resistor, using the following formula: R = [(Vin - 1.2V) / 10mA] - 680

When using a Master hub, the chassis must be well grounded.

The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

24 VDC power supply is only supported if all connected sensors support an input voltage of 24 VDC.

The Power Draw specification is based on a Master with no sensors attached. Every sensor has its own power requirements that need to be considered when calculating total system power requirements..

Encoder

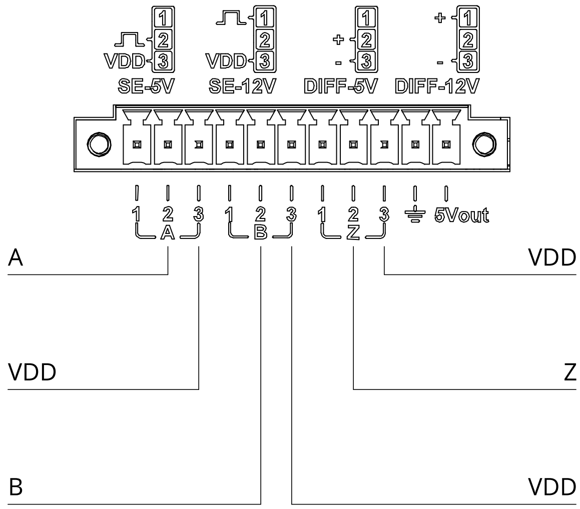
Master 810 and 2410 support the following types of encoder signals: Single-Ended (5 VDC, 12 VDC) and Differential (5 VDC, 12 VDC).

For 5 VDC operation, pins 2 and 3 of each channel are used.

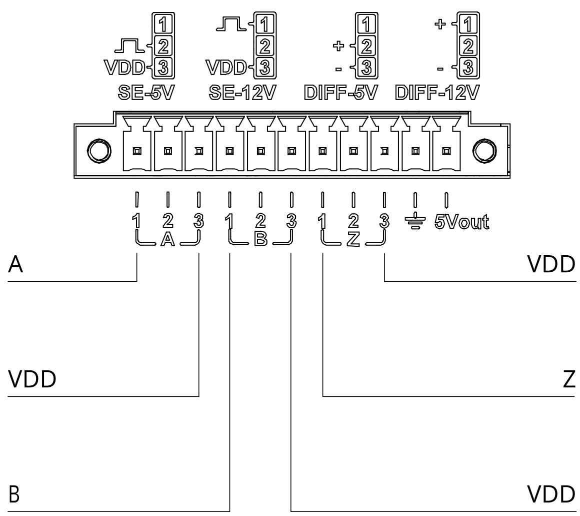
For 12 VDC operation, pins 1 and 3 of each channel are used.

The 5-volt encoder input supports up to 12 volts for compatibility with earlier Master network controllers. However, we strongly recommend connecting 12-volt output encoders to the appropriate 12-volt input to attain maximum tolerance.

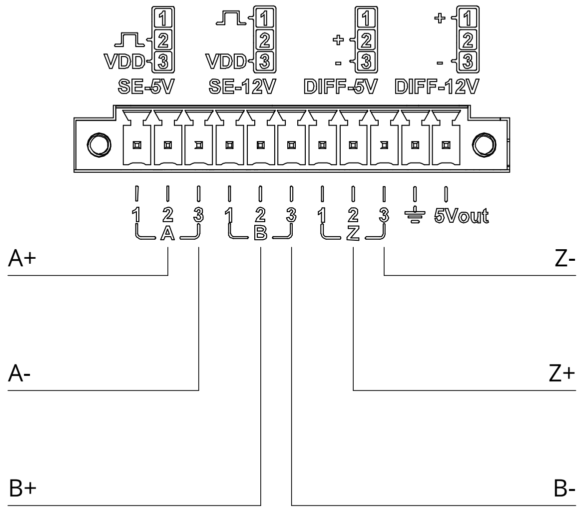
To determine how to wire a Master to an encoder, see the illustrations below.



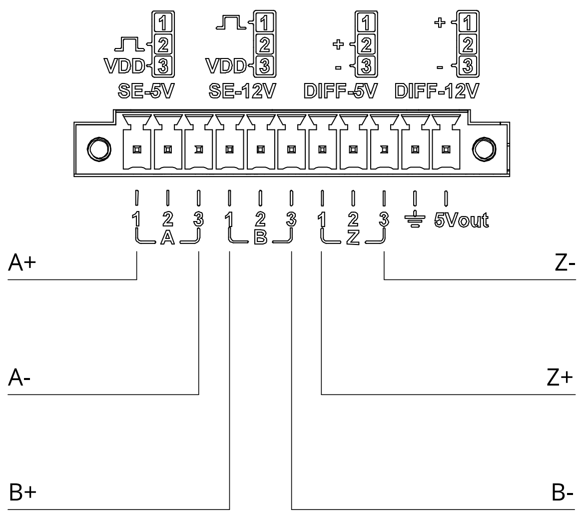
Single-Ended 5 VDC



Single-Ended 12 VDC



Differential 5 VDC



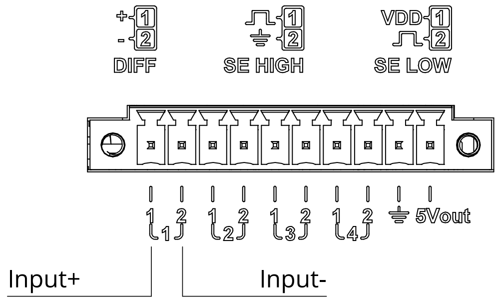
Differential 12 VDC

Input

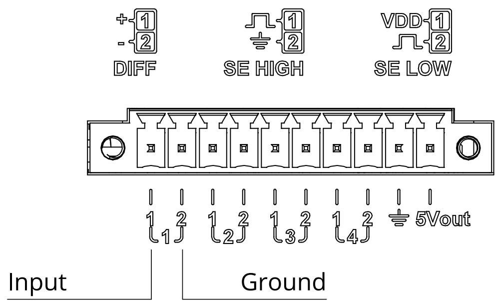
Master 810 and 2410 support the following types of input: Differential, Single-Ended High, and Single-Ended Low.

Currently, Gocator only supports Input 0.

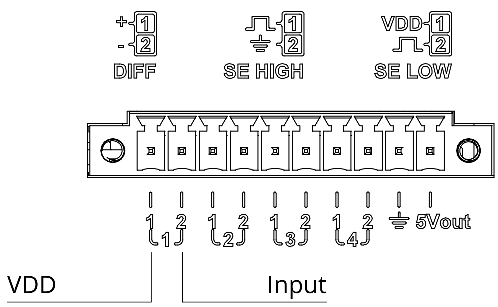
For digital input voltage ranges, see the table below.



Differential



Single-Ended Active High



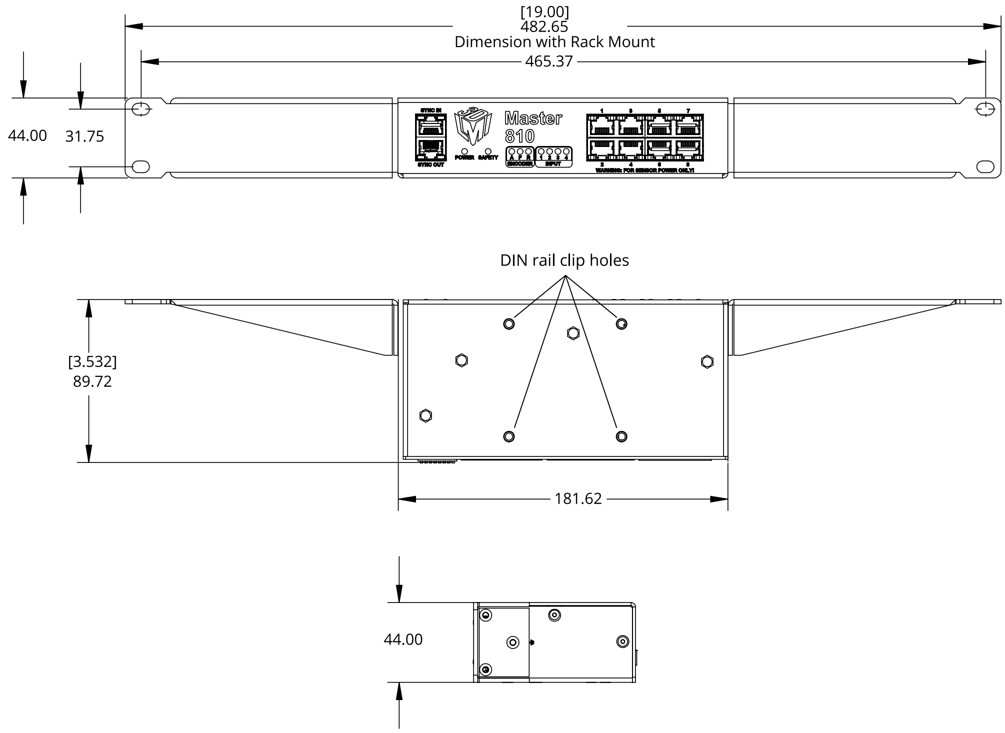
Single-Ended Active Low

Digital Input Voltage Ranges

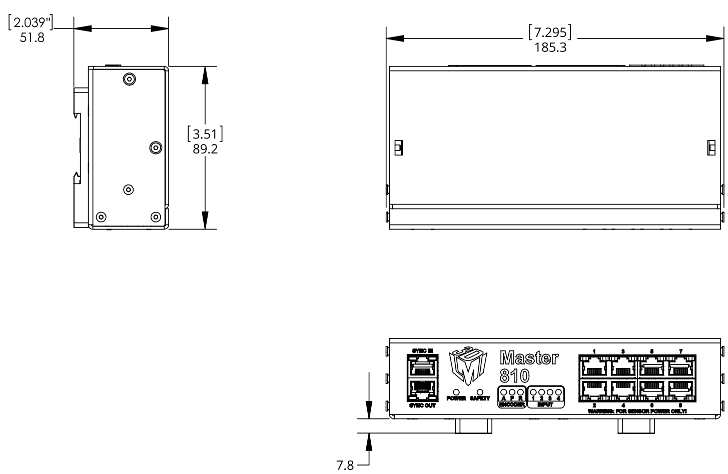
	Input Status	Min (VDC)	Max (VDC)
Single-ended Active High	Off	0	+0.8
	On	+3.3	+24
Single-ended Active Low	Off	(V_{DD} - 0.8)	V_{DD}
	On	0	(V_{DD} - 3.3)
Differential	Off	-24	+0.8
	On	+3.3	+24

Master 810 Dimensions

With 1U rack mount brackets:



With DIN rail mount clips:



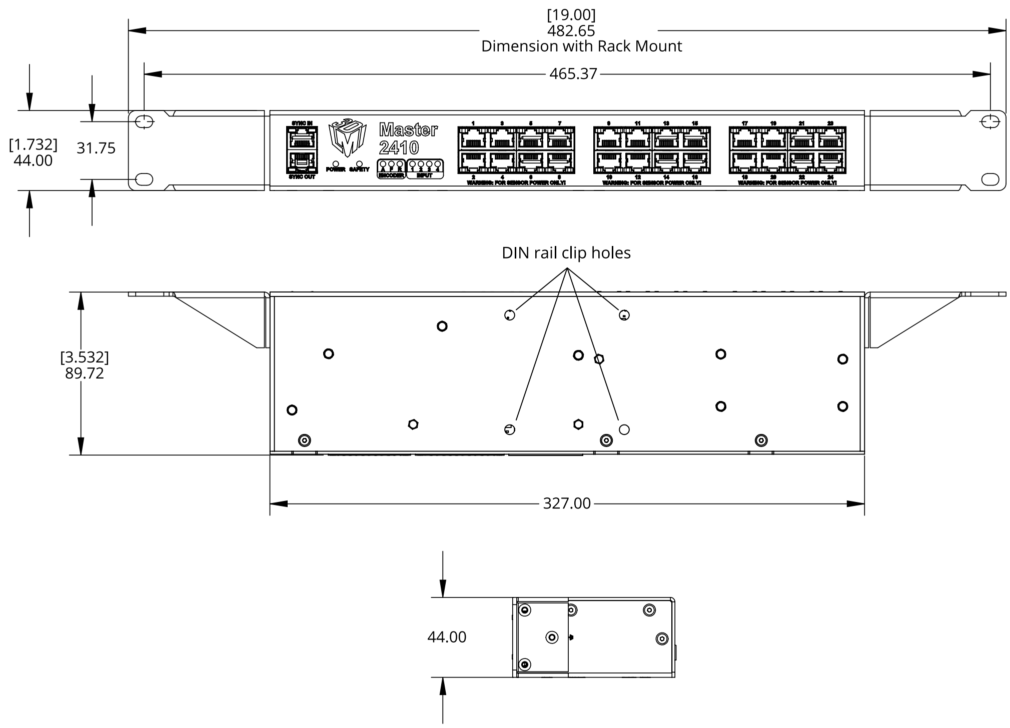
Older revisions of Master 810 and 2410 network controllers use a different configuration for the DIN rail clip holes.

For information on installing DIN rail clips, see *Installing DIN Rail Clips: Master 810 or 2410* on page 30.

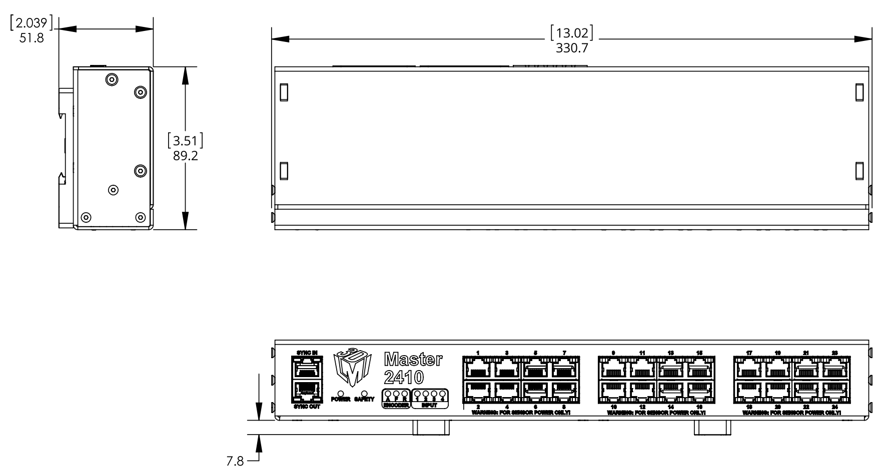
The CAD model of the DIN rail clip is available at https://www.winford.com/products/cad/dinm12-rc.igs.

Master 2410 Dimensions

With 1U rack mount brackets:



With DIN rail mount clips:



Older revisions of Master 810 and 2410 network controllers use a different configuration for the DIN rail clip holes.

For information on installing DIN rail clips, see *Installing DIN Rail Clips: Master 810 or 2410* on page 30.

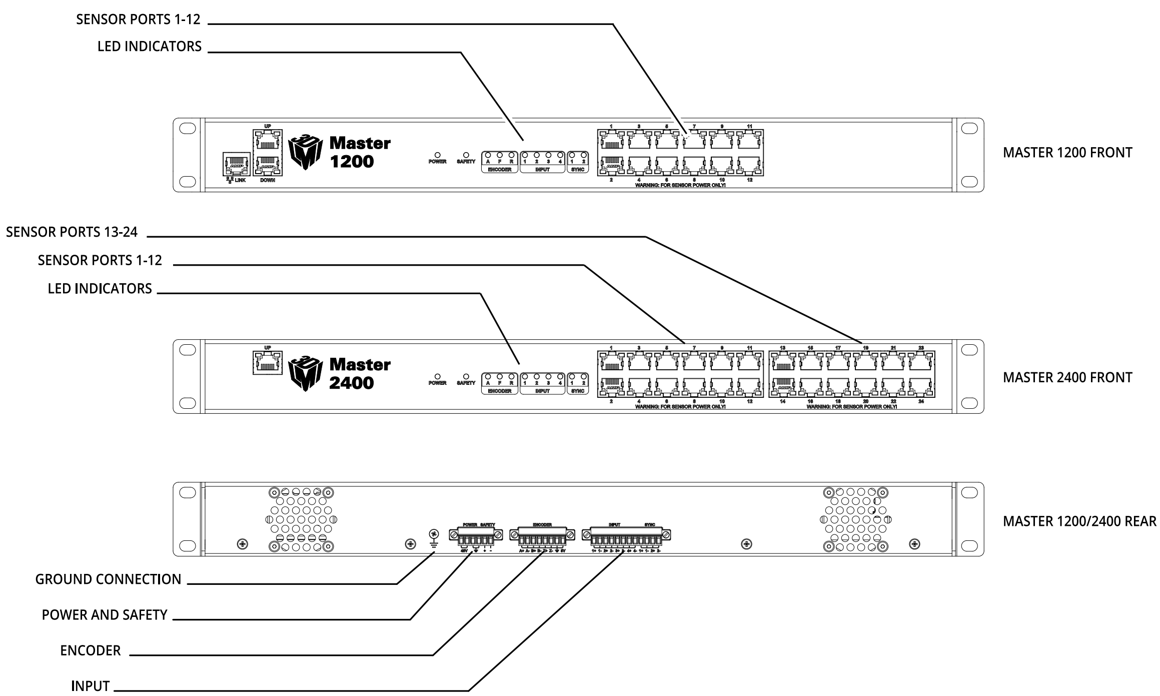
The CAD model of the DIN rail clip is available at https://www.winford.com/products/cad/dinm12-rc.igs.

Master 1200/2400

Master network controllers provide sensor power and safety interlock, and broadcast system-wide synchronization information (i.e., time, encoder count, encoder index, and digital I/O states) to all devices on a sensor network.

	Safety interlock is not supported by snapshot sensors.

	The Phoenix connectors on Master 400/800/1200/2400 are not compatible with the connectors on Master 810/2410. For this reason, if you are switching models in your network, you must rewire the connections to the Master.



Power and Safety (6 pin connector)

Function	Pin
+48VDC	1
+48VDC	2
GND (24-48VDC)	3
GND (24-48VDC)	4
Reserved	5
Reserved	6

	The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

Input (12 pin connector)

Function	Pin
Input 1	1
Input 1 GND	2
Reserved	3
Reserved	4
Reserved	5
Reserved	6
Reserved	7
Reserved	8
Reserved	9
Reserved	10
Reserved	11
Reserved	12

The Input connector does not need to be wired up for proper operation.

Encoder (8 pin connector)

Function	Pin
Encoder_A+	1
Encoder_A-	2
Encoder_B+	3
Encoder_B-	4
Encoder_Z+	5
Encoder_Z-	6
GND	7
+5VDC	8

Master 1200/2400 Electrical Specifications

Electrical Specifications

Specification	Value
Power Supply Voltage	+48 VDC
Power Supply Current (Max.)	10 A
Power Draw (Min.)	5.76 W
Encoder Signal Voltage	Differential (5 VDC)
Digital Input Voltage Range	Logical LOW: 0 to +0.1 VDC Logical HIGH: +3.5 to +6.5 VDC

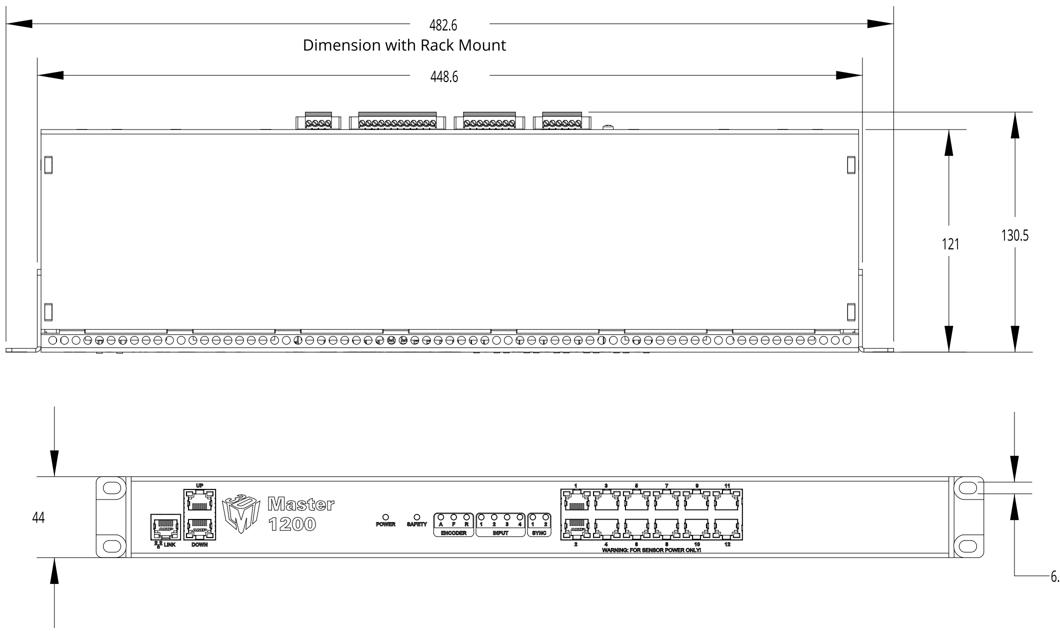
When using a Master hub, the chassis must be well grounded.

The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

The Power Draw specification is based on a Master with no sensors attached. Every sensor has its own power requirements that need to be considered when calculating total system power requirements..

Master 1200/2400 Dimensions

The dimensions of Master 1200 and Master 2400 are the same.



Accessories

Masters

Description	Part Number
Master 100 - for single sensor (development only)	30705
Master 810 - for networking up to 8 sensors	301114
Master 2410 - for networking up to 24 sensors	301115

Cordsets

Description	Part Number
1.2m I/O cordset, open wire end	30864-1.2m
2m I/O cordset, open wire end	30864-2m
5m I/O cordset, open wire end	30864-5m
10m I/O cordset, open wire end	30864-10m
15m I/O cordset, open wire end	30864-15m
20m I/O cordset, open wire end	30864-20m
25m I/O cordset, open wire end	30864-25m
2m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-2m
5m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-5m
10m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-10m
15m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-15m
20m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-20m
25m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-25m
1.2m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-1.2m
2m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-2m
5m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-5m
10m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-10m
15m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-15m
20m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-20m
25m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-25m

Cordsets - 90-degree

Description	Part Number
2m I/O cordset, 90-deg, open wire end	30883-2m
5m I/O cordset, 90-deg, open wire end	30883-5m

Description	Part Number
10m I/O cordset, 90-deg, open wire end	30883-10m
15m I/O cordset, 90-deg, open wire end	30883-15m
20m I/O cordset, 90-deg, open wire end	30883-20m
25m I/O cordset, 90-deg, open wire end	30883-25m
2m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-2m
5m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-5m
10m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-10m
15m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-15m
20m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-20m
25m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-25m
2m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-2m
5m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-5m
10m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-10m
15m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-15m
20m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-20m
25m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-25m

90-degree cordsets are not stocked. Lead times for delivery may be longer.

Custom cordset lengths and connector orientations are available upon request. Prices depend on length and orientation requested. For standard cordsets, the maximum cable length is 60 m.

High Flex Cordsets

Description	Part Number
2m I/O cordset, open wire end	301175-2m
5m I/O cordset, open wire end	301175-5m
10m I/O cordset, open wire end	301175-10m
2m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	301176-2m
5m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	301176-5m
10m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	301176-10m
2m Power and Ethernet to Master cordset, 2x RJ45 ends	301165-2m
5m Power and Ethernet to Master cordset, 2x RJ45 ends	301165-5m
10m Power and Ethernet to Master cordset, 2x RJ45 ends	301165-10m
Description	Part Number
2m I/O cordset, 90-deg, open wire end	301172-2m
5m I/O cordset, 90-deg, open wire end	301172-5m
10m I/O cordset, 90-deg, open wire end	301172-10m
2m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	301171-2m

Description	Part Number
5m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	301171-5m
10m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	301171-10m
2m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	301173-2m
5m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	301173-5m
10m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	301173-10m

Return Policy

Return Policy

Before returning the product for repair (warranty or non-warranty) a Return Material Authorization (RMA) number must be obtained from LMI. Please call LMI to obtain this RMA number.

Carefully package the sensor in its original shipping materials (or equivalent) and ship the sensor prepaid to your designated LMI location. Please ensure that the RMA number is clearly written on the outside of the package. Inside the return shipment, include the address you wish the shipment returned to, the name, email and telephone number of a technical contact (should we need to discuss this repair), and details of the nature of the malfunction. For non-warranty repairs, a purchase order for the repair charges must accompany the returning sensor.

LMI Technologies Inc. is not responsible for damages to a sensor that are the result of improper packaging or damage during transit by the courier.

Software Licenses

CLI11

Website:

https://github.com/CLIUtils/CLI11

License:

xxhash

Website:

https://github.com/Cyan4973/xxHash

License:

xxHash Library

BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

----

xxhsum command line interface

Copyright (c) 2013-present, Yann Collet

All rights reserved.

GPL v2 License

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

JSON for C++

Website:

https://github.com/nlohmann/json

License:

OpENer

Website:

https://github.com/EIPStackGroup/OpENer

License:

The Communications Stack Software for EtherNet/IP, or any portion thereof, with or without modifications, may be incorporated into products for sale. However, the software does not, by itself, convey any right to make, have made, use, import, offer to sell, sell, lease, market, or otherwise distribute or dispose of any products that implement this software, which products might be covered by valid patents or copyrights of ODVA, Inc., its members or other licensors nor does this software result in any license to use the EtherNet/IP mark owned by ODVA. To make, have made, use, import, offer to sell, sell, lease, market, or otherwise distribute or dispose of any products that implement this software, and to use the EtherNet/IP mark, one must obtain the necessary license from ODVA through its Terms of Usage Agreement for the EtherNet/IP technology, available through the ODVA web site at www.odva.org. This license requirement applies equally (a) to devices that completely implement ODVA's Final Specification for EtherNet/IP (“Network Devices”), (b) to components of such Network Devices to the extent they implement portions of the Final Specification for EtherNet/IP, and (c) to enabling technology products, such as any other EtherNet/IP or other network protocol stack designed for use in Network Devices to the extent they implement portions of the Final Specification for EtherNet/IP. Persons or entities who are not already licensed for the EtherNet/IP technology must contact ODVA for a Terms of Usage Agreement.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

picoc

Website:

https://github.com/jpoirier/picoc

License:

Copyright (c) 2009-2011, Zik Saleeba

Copyright (c) 2015, Joseph Poirier

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Zik Saleeba nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

tar (binary only)

License:

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modifiedit, and giving a relevant date.

b) The work must carry prominent notices stating that it isreleased under this License and any conditions added under section7. This requirement modifies the requirement in section 4 to “keep intact all notices”.

c) You must license the entire work, as a whole, under thisLicense to anyone who comes into possession of a copy. ThisLicense will therefore apply, along with any applicable section 7additional terms, to the whole of the work, and all its parts,regardless of how they are packaged. This License gives nopermission to license the work in any other way, but it does notinvalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must displayAppropriate Legal Notices; however, if the Program has interactiveinterfaces that do not display Appropriate Legal Notices, yourwork need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product(including a physical distribution medium), accompanied by theCorresponding Source fixed on a durable physical mediumcustomarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product(including a physical distribution medium), accompanied by awritten offer, valid for at least three years and valid for aslong as you offer spare parts or customer support for that productmodel, to give anyone who possesses the object code either (1) acopy of the Corresponding Source for all the software in theproduct that is covered by this License, on a durable physicalmedium customarily used for software interchange, for a price nomore than your reasonable cost of physically performing thisconveying of source, or (2) access to copy theCorresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of thewritten offer to provide the Corresponding Source. Thisalternative is allowed only occasionally and noncommercially, andonly if you received the object code with such an offer, in accordwith subsection 6b.

d) Convey the object code by offering access from a designatedplace (gratis or for a charge), and offer equivalent access to theCorresponding Source in the same way through the same place at nofurther charge. You need not require recipients to copy theCorresponding Source along with the object code. If the place tocopy the object code is a network server, the Corresponding Sourcemay be on a different server (operated by you or a third party)that supports equivalent copying facilities, provided you maintainclear directions next to the object code saying where to find theCorresponding Source. Regardless of what server hosts theCorresponding Source, you remain obligated to ensure that it isavailable for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, providedyou inform other peers where the object code and CorrespondingSource of the work are being offered to the general public at nocharge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from theterms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices orauthor attributions in that material or in the Appropriate LegalNotices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, orrequiring that modified versions of such material be marked inreasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors orauthors of the material; or

e) Declining to grant rights under trademark law for use of sometrade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of thatmaterial by anyone who conveys the material (or modified versions ofit) with contractual assumptions of liability to the recipient, forany liability that these contractual assumptions directly impose onthose licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

ant-design

Website:

https://ant.design/

array-move

Website:

https://github.com/sindresorhus/array-move

License:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

lodash

Website:

https://lodash.com/

License:

Copyright OpenJS Foundation and other contributors <https://openjsf.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A

mobx

Website:

https://mobx.js.org/README.html

ramda

Website:

https://ramdajs.com/

License:

CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

rc-menu

Website:

https://github.com/react-component/menu

License:

The MIT License (MIT)

Copyright (c) 2014-present yiminghe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react

Website:

https://reactjs.org/

License:

MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-dnd

Website:

https://github.com/react-dnd/react-dnd/

License:

The MIT License (MIT)

Copyright (c) 2015 Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-router

Website:

https://github.com/ReactTraining/react-router

License:

MIT License

Copyright (c) React Training 2016-2018

rxjs

Website:

https://github.com/ReactiveX/rxjs

License:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License.

You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other

Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Support

For help with a component or product, please submit a technical support request at http://support.lmi3d.com/.

If you are unable to submit a support request or prefer to contact LMI by phone or email, use the contact information below.

	Response times for phone or email support requests are longer than requests submitted through LMI's support request system.

North America

Phone	Vancouver +1 833 Gocator (+1 833 462 2867) Detroit +1 248 298 2839
Fax	+1 604 516 8368
Email	support@lmi3d.com

Europe

Phone	+31 45 850 7000
Fax	+31 45 574 2500
Email	support@lmi3d.com

Contact

Americas	EMEAR	ASIA PACIFIC
LMI Technologies (Head Office)	LMI Technologies GmbH	LMI (Shanghai) Trading Co., Ltd.
Burnaby, Canada	Berlin, Germany	Shanghai, China
+1 604 636 1011	+49 (0)3328 9360 0	+86 21 5441 0711

LMI Technologies has sales offices and distributors worldwide. All contact information is listed at lmi3D.com/contact/locations.