GoPxL for Gocator Snapshot Sensors

USER MANUAL

Gocator 3210 & 3500 Series

Firmware version: 1.6.x.x

Document revision: A

Copyright

Copyright © 2026 by LMI Technologies, Inc. All rights reserved.

Proprietary

This document, submitted in confidence, contains proprietary information which shall not be reproduced or transferred to other documents or disclosed to others or used for manufacturing or any other purpose without prior written permission of LMI Technologies Inc.

No part of this publication may be copied, photocopied, reproduced, transmitted, transcribed, or reduced to any electronic medium or machine readable form without prior written consent of LMI Technologies, Inc.

Trademarks and Restrictions

Gocator™ is a registered trademark of LMI Technologies, Inc. Any other company or product names mentioned herein may be trademarks of their respective owners.

Information contained within this manual is subject to change.

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

Copyright **2**

Table of Contents **3**

Introduction **17**

Safety and Maintenance **18**

Electrical Safety 18

Handling, Cleaning, and Maintenance 18

Environment and Lighting 19

Using the Manual **21**

Installation, Setup & Hardware Overview **23**

Gocator Overview 23

Sensor Part Numbers 23

Upgrading from Gocator Classic Firmware 23

Hardware Overview 24

Gocator Sensor 24

Gocator 3x00 Cordsets 24

Master 100 25

Master 810 / 2410 26

System Overview 27

Standalone System 27

Multi-Sensor System 28

Installation 30

Torque Recommendations 30

Gocator 3210A or B, 3506B: 24 VDC Operation and Cordset Length 31

Mounting 31

Cordset Bend Radius Limits 32

Grounding 34

Gocator 34

Recommended Practices for Cordsets 34

Master Network Controllers 35

Grounding When Using a DIN Rail (Master 810/2410) 35

Additional Grounding Schemes 36

Installing DIN Rail Clips: Master 810 or 2410 36

Configuring Master 810 38

Setting the Divider 39

Encoder Quadrature Frequency 39

Setting the Debounce Period 40

Network and Sensor Setup 41

Client Setup 41

Gocator Setup 43

Standalone Sensor System 43

Required Ports 45

Key Concepts **47**

Sensors, Sensor Groups, and Systems 47

3D Acquisition 47

Clearance Distance, Field of View and Measurement Range 49

Stereo Correlation vs. Independent Triangulation 50

3D Data Output 51

Coordinate Systems 51

Unaligned Coordinates 51

Aligned Coordinates 52

Sensor Coordinates 53

System Coordinates 53

Part and Section Coordinates 54

Resampling 54

Data Generation and Processing 55

Part Detection 55

Sectioning 55

Part Matching 56

Measurement and Anchoring 56

Complex Defect Detection 57

Tool Chaining 57

Tool Data 57

Anchoring Measurements 58

Geometric Features 59

Arrays, Batching, and Aggregation 62

Output and Control 63

User Interface Overview **65**

Status Bar 68

System Messages 68

Metrics Area 69

Keyboard and Mouse Shortcuts **70**
File Formats **76**
Formats 76
File Name Conventions 78
Data Viewer **79**
Image Mode 85
Exposure Information 86
Profile Mode 87
Surface Mode 88
Surface and Perspective Display Options 89
Displaying and Pinning Outputs 94
Frame Information 100
Using Multiple Data Viewers 100
Intensity Output 102
Recording and Working with Scan Data **104**
Starting, Stopping, and Recording 105
Playing Back Recorded Data 107
Using the Replay Editor 108
Creating, Saving and Loading Jobs **112**
Pro Tools **113**
Tools Included by Package 113
Installing CodeMeter Runtime for GoPxL for PC 114
Configuring GoPxL **118**
Connect - Creating a System & Networking 119
Configuring Power 119
Projector Shutdown 120
Network Settings 120
Acquire - Alignment and Acquisition 122
Aligning Sensors 122
Aligning Sensors with up to 5 Degrees of Freedom 123
Performing Stationary Alignment 125
Stationary Flat Surface 126
Stationary Plate 126
Target Specifications 127
Configuring GoPxL for Plate Alignment 127
Clearing Alignment 129
Transformations 130
Performing the Alignment 131
Performing Stationary Alignment 133
Stationary Flat Surface 134
Stationary Plate 134
Target Specifications 135
Configuring GoPxL for Plate Alignment 135
Transformations 137
Scan Modes and Intensity 139
Spacing on X and Y (Uniform Spacing) 140
Reduce Occlusion 141
Triggers 141
Trigger Settings 143
Trigger Examples 144
Maximum Input Trigger Rate 144
Maximum Encoder Rate 145
Filters 145
Exposure 148
Using the Focus Pattern (Gocator 3506 only) 149
Gocator 3210A or B, 3506B on 24 VDC 151
Single Exposure 151
Multiple Exposure 152
Active Area 154
Advanced Settings 159
Material Type 160
Camera Gain 160
Other Advanced Settings 161
Inspect - Measurement and Processing Tools 162
Adding a Tool 164
Understanding the Tool List 166
Tool Categories 166
Tool Connection Logic 167
Removing a Tool 169
Duplicating a Tool 170
Renaming a Tool 171
Working with Tool Chains 172
Changing Tool Display Options 174
Reordering Tools 175
Data Types 176

Understanding the Data Flow in Tool Chains 177
Connecting Tools 182
Disconnecting Tools 184
Tool Grouping 185
Tool and Group Locking 190
Measurement Values and Labels 194
Tool Configuration 196
Inputs and Outputs 198
Arrays, Batching, and Aggregation 199
Arrays from Multi-instance Tools 200
Arrays from Multi-layer Output 202
G2 Multi-sensor Output 203
Batching 204
Aggregating 206
Batch and Array Indicators in GoPxL 208
Measurement Anchoring 209
Regions 214
Standard Regions 216
Flexible Regions 217
Working with Circular and Elliptical Regions 221
Region Rotation 223
Feature Points 224
Geometric Features 227
Fit Lines 227
Output Filters 228
Decisions 229
External AI Models 231
Roboflow 232
Array Tools 233
Array Create 234
Inputs 234
Parameters 234
Outputs 235
Array Decision 236
Inputs 236
Parameters 236
Outputs 238
Array Index 240
Inputs 240
Parameters 240
Outputs 241
Conditional Logic Tool 243
Inputs 243
Parameters 243
Outputs 245
Data Export Tool 247
Inputs 248
Parameters 249
Outputs 250
Feature Tools 252
Feature Circle Create 253
Inputs 253
Parameters 253
Output Types 254
Constant Circle 255
Circle from Points 255
Circle from point and plane 256
Circle tangent to two lines 257
Outputs 258
Feature Dimension 260
Inputs 260
Parameters 261
Outputs 262
Feature Intersect 266
Inputs 266
Parameters 267
Outputs 268
Feature Line Create 271
Inputs 271
Parameters 272
Output Types 273
Line 273
Constant Line 273
Line from Offset 273
Perpendicular or Parallel Line from 274

Point and Line
Perpendicular Line from Point to Plane 275
Intersect Line of Two Planes 276
Projected Line on Plane 276
Projected Center Line from Two Lines and a Plane 277
Rotate Line around a Point 277
Outputs 277
Feature Plane Create 280
Inputs 280
Parameters 280
Output Types 281
Constant Plane 281
Plane from a Point and Normal 282
Plane from a Point and Line 282
Plane from Three Points 283
Plane from Circle 283
Parallel Plane from Point and Plane 283
Perpendicular Plane from Point and Plane 284
Perpendicular Plane from Line and Plane 284
Bisect Plane from Two Planes 285
Outputs 285
Feature Point Create 288
Inputs 288
Parameters 289
Output Types 289
Point 289
Constant Point 290
Point from Offset 290
Point from Three Planes 291
Point from Line and Circle 291
Point from Line and Line 292
Point from Line and Plane 292
Projected Point on Line 293
Projected Point on Plane 293
Outputs 294
Feature Robot Pose 297
Inputs 297
Parameters 298
Outputs 298
Image Tools 300
Image Anomaly Detector 301
Inputs 303
Parameters 304
Development Workflow 306
Collect dataset 306
Implement a Preprocessing Toolchain 306
Train Model 307
Add Image Anomaly Detector Tool 307
Create New or Open Project 308
Label Frames and Choose Set 310
Configure Training 311
Test Model Performance 315
Deploy in production 317
Anomaly Tool Setup 317
Outputs 318
Anomaly Segmentation Outputs 319
Store production data 320
Train using additional data 320
Training Archive 324
Inputs 324
Parameters 325
Outputs 326
Image Arithmetic 328
Inputs 328
Parameters 329
Outputs 331
Image Barcode 332
Inputs 332
Parameters 333
Outputs 336
Image Blob 340
Inputs 340
Parameters 341
Outputs 346

Image Caliper 349
Image Circle 350
Inputs 350
Parameters 350
Outputs 355
Image Classifier 357
Inputs 357
Development Workflow 357
Collect dataset 358
Implement a Preprocessing Toolchain 358
Train Model 358
Add the Tool 358
Create New or Open Project 358
Add Classes 360
Label Frames 360
Configure Training 362
Model Performance Testing 365
Parameters 365
Outputs 367
Image Converter 371
Inputs 371
Parameters 371
Outputs 372
Image Feature Mask 374
Inputs 374
Parameters 375
Outputs 376
Image Filter 378
Inputs 378
Parameters 379
Outputs 381
Image Inference 383
Inputs 384
Parameters 385
Optimizing Models 389
Outputs 389
Image Intensity 392
Inputs 392
Parameters 392
Outputs 394
Image Edge 396
Inputs 396
Parameters 396
Outputs 400
Image Edge Count 402
Inputs 402
Parameters 403
1-Dimensional Mode 404
2-Dimensional Mode 405
Outputs 408
Image Matrix Code 410
Inputs 411
Parameters 411
Outputs 414
Image Object Detector 419
Inputs 420
Development Workflow 420
Collect dataset 421
Implement a Preprocessing Toolchain 421
Train Model 421
Add the Tool 422
Create New or Open Project 424
Add Classes 427
Label Objects in Frames, Set Bounding Boxes 427
Auto-label and Auto-accept Settings 432
Configure Training 432
Validate the Model 435
Models Tab 438
Parameters 439
Outputs 441
Image OCR 443
Inputs 444
Parameters 444
Outputs 449
Image Pattern 451

Inputs 452
Creating a Pattern 452
Editing a Pattern 455
Parameters 456
Pattern Parameters 456
Tool (Matching) Parameters 459
Outputs 462
Image Pixel Count 465
Inputs 465
Parameters 466
Outputs 466
Image QR Code 469
Inputs 470
Parameters 470
Outputs 474
Image Track 478
Inputs 480
Parameters 480
Outputs 484
Image Transform 486
Inputs 486
Parameters 486
Outputs 488
Measurement Formula Tool 489
Inputs 489
Parameters 490
Outputs 491
Mesh Tools 493
Mesh Bounding Box 494
Inputs 494
Parameters 495
Outputs 497
Mesh Plane 499
Inputs 500
Parameters 501
Outputs 503
Mesh Projection 505
Inputs 505
Parameters 507
Outputs 507
Mesh Template Matching 509
Inputs 509
Parameters 510
Outputs 513
Profile Tools 515
Profile Advanced Height 516
Inputs 516
Parameters 517
Master Comparison 518
X Correction 519
Reference Line 519
Outputs 519
Profile Area 522
Inputs 522
Parameters 523
Outputs 524
Profile Bounding Box 527
Inputs 527
Parameters 528
Outputs 529
Profile Circle 532
Inputs 532
Parameters 533
Outputs 534
Profile Circle Radii 537
Inputs 537
Parameters 538
Outputs 541
Profile Closed Area 544
Inputs 544
Parameters 545
Outputs 548
Profile Dimension 550
Inputs 550
Parameters 551
Outputs 552

Profile Edge 555
Inputs 555
Parameters 556
Outputs 561
Profile Filter 564
Inputs 564
Parameters 565
Outputs 568
Profile Groove 570
Groove Algorithm 570
Inputs 572
Parameters 573
Outputs 575
Profile Intersect 578
Inputs 578
Parameters 579
Outputs 581
Profile Line 583
Inputs 583
Parameters 584
Outputs 586
Profile Mask 590
Inputs 590
Parameters 591
Outputs 592
Profile Panel 594
Inputs 594
Parameters 595
Outputs 599
Gap and Flush Algorithms 601
Profile Part Detection 603
Profile Position 604
Inputs 604
Parameters 605
Outputs 606
Profile Roughness 609
Inputs 609
Parameters 610
Outputs 626
Profile Round Corner 628
Inputs 628
Parameters 629
Outputs 633
Profile Strip 636
Inputs 636
Parameters 637
Outputs 641
Strip Algorithm 643
Strip Start and Terminate Conditions 643
Strip Step Edge Definitions 645
Profile Template Matching 647
Inputs 647
Parameters 649
Outputs 651
Profile Thickness 654
Inputs 654
Parameters 656
Outputs 657
Profile Transform 659
Inputs 660
Parameters 661
Outputs 662
Raw Profile Matching 664
Inputs 664
Parameters 665
Outputs 667
Script Tool 670
Included Python Modules 670
Inputs 671
Parameters 671
Outputs 673
Functions and Classes 675
Getting Input 675
Drawing Graphics 676
Sending Output 678
Logging 681

Memory 681
Data Structures 681
Examples 688
Surface Tools 712
Isolating Parts from Surface Data 712
Surface Align Ring 714
Surface Align Wide 715
Surface Anomaly Detector 716
Hardware and Software Requirements 717
Inputs 719
Parameters 720
Development Workflow 721
Collect dataset 722
Implement a Preprocessing Toolchain 722
Train Model 722
Add Surface Anomaly Detector Tool 722
Create New or Open Project 723
Label Frames and Choose Set 725
Configure Training 726
Test Model Performance 732
Deploy in production 735
Anomaly Tool Setup 735
Outputs 737
Store production data 739
Train using additional data 739
Training Archive 742
Inputs 742
Parameters 743
Outputs 744
Sample Files 746
Surface Arithmetic 753
Inputs 753
Parameters 754
Outputs 755
Surface Ball Bar 756
Inputs 756
Parameters 756
Outputs 758
Surface Blob 760
Inputs 762
Parameters 763
Outputs 771
Surface Bounding Box 774
Inputs 774
Parameters 775
Outputs 780
Surface Circular Edge 784
Calipers, Extracted Paths, and Edge Points 785
Inputs 785
Parameters 786
Outputs 795
Surface Countersunk Hole 799
Inputs 799
Parameters 800
Outputs 802
Surface Curvature Correction 807
Inputs 808
Parameters 809
Outputs 811
Surface Cylinder 813
Inputs 813
Parameters 814
Outputs 816
Surface Dimension 818
Inputs 818
Parameters 819
Outputs 820
Surface Direction Filter 824
Inputs 825
Parameters 826
Outputs 829
Surface Edge 830
Paths and Path Profiles 831
Inputs 832
Parameters 833
Outputs 841

Surface Ellipse 846
Inputs 846
Parameters 847
Outputs 848
Surface Extend 851
Inputs 851
Parameters 852
Outputs 853
Surface Feature Mask 855
Inputs 855
Parameters 856
Outputs 860
Surface Filter 862
Inputs 862
Parameters 863
Outputs 867
Surface Flatness 869
Inputs 869
Parameters 870
Outputs 872
Surface Hole 875
Measurement Region 875
Hole Algorithm 876
Inputs 877
Parameters 878
Outputs 880
Surface Image Converter 883
Inputs 883
Parameters 883
Outputs 885
Surface Inference 886
Inputs 886
Parameters 886
Optimizing Models 890
Outputs 891
Surface Mask 893
Inputs 894
Parameters 895
Outputs 896
Surface Merge Wide 898
Surface Mesh 899
Surface Opening 900
Measurement Region 900
Opening Algorithm 901
Inputs 901
Parameters 902
Outputs 906
Surface Part Detection 908
Inputs 908
Parameters 908
Outputs 911
Surface Pattern Matching 913
Creating a Template 915
Inputs 916
Parameters 917
Outputs 921
Surface Plane 923
Inputs 923
Parameters 925
Outputs 926
Surface Position 929
Inputs 930
Parameters 931
Outputs 932
Surface Roughness 934
Inputs 935
Parameters 935
Outputs 938
Calibration Procedure 941
Roughness Calculation Procedure 941
Surface Section 943
Inputs 944
Parameters 946
Outputs 949
Surface Segmentation 951
Inputs 951

Parameters ... 952
Outputs ... 958
Surface Sphere ... 961
Inputs ... 961
Parameters ... 963
Outputs ... 964
Surface Stitch ... 968
Inputs ... 969
Parameters ... 969
Outputs ... 972
Surface Stud ... 973
Stud Algorithm ... 974
Inputs ... 974
Parameters ... 975
Outputs ... 977
Surface Track ... 980
Key Concepts ... 981
Track Location ... 983
Peak Detection ... 983
Side Detection ... 983
Center Point Detection ... 984
Configuring the Surface Track Tool ... 984
Inputs ... 984
Parameters ... 985
Outputs ... 989
Training Archive ... 992
Surface Transform ... 993
Inputs ... 995
Combinations of geometric feature inputs and results ... 995
Plane ... 996
Line ... 996
Point ... 997
Plane + Line ... 998
Plane + Point ... 999
Line + Point ... 1000
Plane + Line + Point ... 1001
Parameters ... 1002
Outputs ... 1004
Surface Vibration Correction ... 1006
Inputs ... 1006
Parameters ... 1007
Outputs ... 1008
Surface Volume ... 1010
Inputs ... 1010
Parameters ... 1011
Outputs ... 1012
User Data Input Tool ... 1015
Inputs ... 1015
Parameters ... 1016
Outputs ... 1017
Communicate - PLC, Digital, GDP, HMI ... 1018
GoPxL Data Protocol ... 1023
Modbus TCP Protocol ... 1027
Modbus TCP Protocol Format - PLC Information ... 1031
Messages ... 1031
Connections Map ... 1032
Supported Block Types ... 1033
Control Input ... 1033
Control Output ... 1034
System State ... 1035
Sensor Group State ... 1035
Stamp ... 1036
Measurement ... 1036
String ... 1037
EtherNet/IP Protocol ... 1038
EtherNet/IP Protocol Format - PLC Information ... 1045
Supported EtherNet/IP Objects ... 1045
Object and Register Assembly Mapping ... 1046
Supported Block Types ... 1047
Identity Object (Class 0x01) ... 1048
TCP/IP Object (Class 0xF5) ... 1048
Ethernet Link Object (Class 0xF6) ... 1049
Assembly Object (Class 0x04) ... 1049

Explicit Messaging Command Input 1049
Input Structure 1050
Explicit Messaging Command Output ... 1051
Output Structure 1051
Explicit Messaging Sensor Group State .. 1051
Output Structure 1052
Explicit Messaging System State 1052
Output Structure 1052
Explicit Messaging Sample State 1053
Stamp Block 1054
Measurement Value Block 1054
String Block 1054
System Status Block 1055
Explicit Messaging User Data Input 1055
User Data Input Block 1056
Implicit Messaging Input 1056
Input Structure 1056
Implicit Messaging Input Large 1057
Input Structure 1058
Implicit Messaging Output 1058
Stamp Block 1059
Measurement Value Block 1059
String Block 1059
System Status Block 1060
Rockwell Allen-Bradley Instructions 1061
Software and Hardware Setup 1061
Byte Order Options 1062
Implicit Messaging - Setting Up GoPxL ... 1062
Implicit Messaging - Setting Up the PLC . 1064
Install EDS File 1064
Add Gocator IO Device to PLC Program 1068
Implicit Messaging - Using Gocator Command Assembly 1080
Starting a Sensor 1081
Loading a Sensor Job File 1083
Explicit Messaging - Setting Up GoPxL ... 1084
Explicit Messaging - Read Single Attribute on PLC 1085
Explicit Messaging - Set Single Attribute to GoPxL on PLC 1093
Loading a Sensor Job File 1105
Ethernet ASCII Protocol 1107
Ethernet ASCII Protocol Format - PLC Information 1112
Commands and Formats 1113
Client Commands 1113
Set Data Commands 1116
Timetick Interpretation 1119
Errors for Each Command 1119
Data Output Format Mode 1124
PROFINET Protocol 1126
Using PROFINET on a PC 1130
Connections Map 1130
Assembly Objects 1130
Supported Block Types 1131
Control Input 1132
Control Output 1133
System State 1133
Sensor Group State 1134
Stamp 1134
Measurement 1135
String 1135
Siemens PLC Instructions 1135
Creating and Configuring New Projects for Siemens PLCs 1136
Creating a New Project 1136
Adding and Configuring the Siemens S7-1200 PLC 1136
Configuring PLC Network Settings 1137
Verifying PLC Connectivity 1138
Configuring PROFINET Connection between a Siemens S7-1200 PLC and GoPxL 1139
Installing GSD Files 1139
Adding and Configuring GoPxL Sensor 1141
Verifying Configuration and Connection 1141
Compiling and Downloading 1142

Configuration to PLC
Monitoring Sensor State 1143
Device Overview: Module Mapping and Usage 1144
Digital Output 1147
HMI 1153
Data Storage Service 1158
Settings 1159
Log Format 1165
Health 1166
Monitor - Measurement Stats, Health & Performance 1167
Measurements 1167
Health 1168
Performance 1171
Audit Log 1172
Manage - Sensor Management and Maintenance 1175
Settings 1176
Autostart 1176
Administrator Login 1177
Date and Time 1178
Jobs 1179
Characters Allowed in Job Names 1181
Maintenance 1183
Software Upgrade 1184
Backup and Restore 1184
Factory Restore 1185
Support 1186
Running GoPxL on a Windows PC 1188
System Requirements 1189
Launching GoPxL on a Windows PC 1190
Running GoPxL Manager in Administrator Mode 1192
Closing GoPxL Manager and Stopping All PC Instances 1193
Application Settings 1194
Installing Licensing Software 1195
Instance Settings 1196
Creating a Diagnostic File 1199
Accelerating a Sensor 1200
Stopping Acceleration 1203
Loading Scan Data 1204
GoPxL Emulator 1207
GoHMI and GoHMI Designer 1210
Overview of Managers and Settings Pages 1215
Enabling and Configuring GoHMI in GoPxL 1216
Launching GoHMI Designer 1220
Opening, Editing, and Saving a Project 1224
Publishing Changes or Exiting 1225
Changing Mapped Outputs in GoPxL 1226
Importing and Exporting an HMI 1226
Using a Data Viewer in an HMI 1228
Limiting Flash Memory Write Operations 1233
GoPxL SDK and REST API 1234
Installation Guide 1234
Location of GoPxL SDK and Documentation 1234
Visual Studio Solution Files and Linux Makefile 1234
Building the SDK 1234
Running SDK Applications 1235
System Requirements 1235
Accessing Documentation 1235
Sample Projects 1235
Header Files 1236
Getting Started with the SDK 1236
Connecting a Device 1236
GoSystem Class Member Functions 1237
Discover Devices 1238
Configure Device Settings 1238
Accessing Sensor Resources 1239
Accessing Scanner Resources 1240
Accessing an Array Element 1240
Receive Device Data 1240
Receive Metrics Information 1243
Accelerated Devices 1243
GDP Data Types 1244
Value Types 1244
Output Types 1244
GoDataSet 1245

Measurement Values and Decisions 1246
Limiting Flash Memory Write Operations 1246
Python GDK 1247
Requirements and Installation 1247
Developing Custom Tools 1248
Enabling the Local Python Tool Server 1251
Debugging Custom Tools 1252
Deploying Your Custom Tools 1254
Integrations 1257
Protocols (PLCs and other hardware) 1257
GenICam GenTL Driver 1258
16-bit RGB Image 1261
16-bit Grey Scale Image 1262
Registers 1264
XML Settings File 1265
Interfacing with Halcon 1266
Setting Up Halcon 1266
Halcon Procedures 1272
XmlSetting and XmlCommand Parameters 1277
Sensor Settings 1277
GenTL Driver Settings and Commands 1278
Sample Program for Various GenTL Settings 1280
Generating Halcon Acquisition Code 1282
GoAIK Cognex VisionPro Driver 1284
16-bit Grey Scale Image 1284
XML Settings File 1286
Interfacing with VisionPro and Ether-Inspect 11 1286
Setting Up VisionPro 1286
Commands 1289
Utilities 1290
GoPxL Discovery Tool 1290
Replay Utility Tool 1294
CSV File Format 1295
Info 1295
DeviceInfo 1296
RecordingFilter 1297
Profile 1297
RawProfile 1298
Part 1299
SurfacePointCloud 1299
GoPxL Add-on Tool Manager 1300
Adding Beta Tools 1302
Removing Beta Tools 1305
Pattern Editor 1307
Launching the Pattern Editor 1308
Overview of the Editor 1310
Models 1311
Adding and Removing Features Manually 1313
Setting Required and Locating Features 1318
Model Creation Settings and Rebuilding 1319
Coarseness Levels 1320
Thresholds 1321
Tracking Inertia 1321
Feature Selection 1322
Saving and Discarding Changes 1323
Miscellaneous 1324
Track Editor 1325
Ruler Settings 1329
Specifications 1331
Sensors 1332
Gocator 3210 Sensor 1333
Gocator 3210 1335
Gocator 3500 Series 1338
Gocator 3506 1340
Gocator 3520 1343
Sensor Connectors 1346
Gocator Power/LAN Connector 1346
Grounding Shield 1347
Power 1347
Safety Input 1347
Gocator I/O Connector 1348
Grounding Shield 1349
Digital Outputs 1349
Inverting Outputs 1349

Digital Input 1350
Encoder Input 1350
Serial Output 1351
Analog Output 1351
Master Network Controllers 1353
Master 100 1353
Master 100 Dimensions 1354
Master 400/800 1355
Power and Safety 1355
Encoder 1356
Input 1356
Master 400/800 Electrical Specifications ... 1357
Master 400/800 Dimensions 1358
Master 810/2410 1359
Power and Safety 1361
Encoder 1362
Input 1362
Electrical Specifications 1363
Encoder 1364
Input 1366
Master 810 Dimensions 1368
Master 2410 Dimensions 1369
Master 1200/2400 1370
Power and Safety 1370
Encoder 1371
Input 1371
Master 1200/2400 Electrical Specifications .1372
Master 1200/2400 Dimensions 1372
Accessories **1374**
Troubleshooting **1376**
Return Policy **1377**
Software Licenses **1378**
Support **1399**
Contact **1400**

Introduction

This documentation describes how to connect, configure, and use a Gocator snapshot sensor ("G3 sensor" for short) using the GoPxL software.

> Before using a sensor, you should familiarize yourself with sensor safety and maintenance. For more information, see *Safety and Maintenance* on page 18.

For a breakdown of this documentation, see *Using the Manual* on page 21.

The documentation applies to the following:

- Gocator 3210
- Gocator 3500 series
- GoMax ORIN / ORIN+
- X64-based PC (Intel/AMD) running Windows 10 (for running GoPxL on a PC)

> GoPxL does not support the original, pre-NX version of GoMax.

> If you need to use the Anomaly Detector tool, the Image tools, or any other Pro Tool in GoPxL on PC, you must purchase a Pro Tools license and dongle, and download the "GoPxL Pro Utilities" package (14658-x.x.x.x_GoPxL_Pro_Utilities.exe).

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.

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 1348.

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 1332.

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
The sensor operating temperature range is 0 to 50 °C.

Keep ambient temperatures below 40 °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. In some applications, to maintain this maximum difference, you may need additional heatsinking. You can monitor this temperature on the **Monitor** > **Health** page.

For all sensors, the storage temperature is -30 to 70 °C.

For all sensors, relative humidity (non-condensing) is 25% to 85%.

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

Protect Master network controllers from dusty environments

⚠	If you are installing a Master 810 or 2410 in a dusty environment, LMI strongly recommends installing the device in a vented cabinet or enclosure to avoid contamination.

Using the Manual

Use the following to help decide which part of this manual you need.

> The tool topics from this manual are available in GoPxL, in the **Tool Help** panel at the bottom of the interface.

Manual breakdown

Category	Description
Safety and Maintenance	Important safety and maintenance information (see *Manage - Sensor Management and Maintenance* on page 1175).
Getting Started / Installation	Hardware overview and installation information (see *Installation, Setup & Hardware Overview* on page 23).
Key Concepts	Fundamental Gocator sensor concepts (see *Key Concepts* on page 47).
User Interface Overview	General overview of the GoPxL user interface (see *User Interface Overview* on page 65).
Data Viewer	Description of the data viewer and how to use it (see *Data Viewer* on page 79).
File Formats	Lists the formats used in GoPxL (see *File Formats* on page 76).
Working with Scan Data	Describes how to record scan data and how to play back recordings (see *Recording and Working with Scan Data* on page 104).
Creating, Saving and Loading Jobs	Describes how to work with jobs (see *Creating, Saving and Loading Jobs* on page 112).
Configuring Systems	Contains subsections describing how to do the following: • Perform general sensor maintenance • Create a sensor system • Align sensors • Configure acquisition • Configure inspection and processing tools • Configure control (including protocol and GoHMI) • Configure reporting For more information, see *Configuring GoPxL* on page 118.
Using a PC Instance of GoPxL	Describes how to start and use a PC instance of GoPxL, as well as GoPxL Manager. You use a PC instance of GoPxL to accelerate a sensor and to examine scan data offline. For more information, see *Running GoPxL on a Windows PC* on page 1188.
Development Kits and API	Provides an overview of the GoPxL SDK and the REST API and describes installation of the SDK (see *GoPxL SDK and REST API* on page 1234).
GoHMI	Describes how to configure GoHMI in a PC instance of GoPxL. Also describes the

Category	Description
	GoHMI Designer (see *GoHMI and GoHMI Designer* on page 1210).
Integrations	Describes the industrial protocols you can use in GoPxL (see See *Integrations* on page 1257).
Specifications	Provides sensor specifications and drawings (see *Specifications* on page 1331).

Installation, Setup & Hardware Overview

The following sections provide hardware and system overviews and describe installation and initial networking setup procedures.

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.

Before using sensors, see *Safety and Maintenance* on page 18.

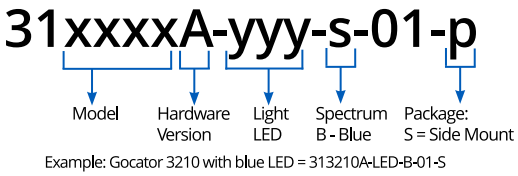
For information on the physical installation and mounting of sensors, as well as a general hardware overview, see *Installation, Setup & Hardware Overview* above.

For key concepts on how sensors acquire data and perform measurements, see *Key Concepts* on page 47.

For information on using the PC-based interface (and the differences between running GoPxL on-sensor and on a PC), see *Running GoPxL on a Windows PC* on page 1188.

Sensor Part Numbers

Use the following to understand sensor part numbers:



Example: Gocator 3210 with blue LED = 313210A-LED-B-01-S

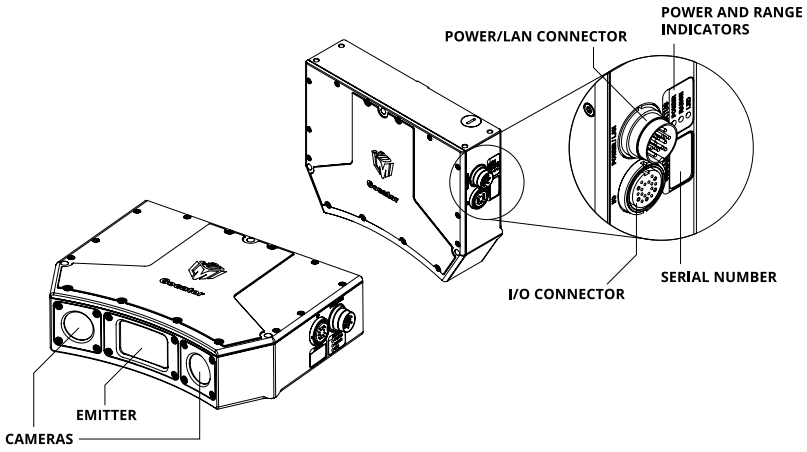
Upgrading from Gocator Classic Firmware

If you are upgrading your sensor from Gocator firmware 6.2 or earlier to GoPxL, and you have created pattern files using Surface Pattern Matching or track files using Surface Track in the Gocator firmware, be sure to add instances of those tools GoPxL and then remove those files manually. These files can take up considerable space.

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.

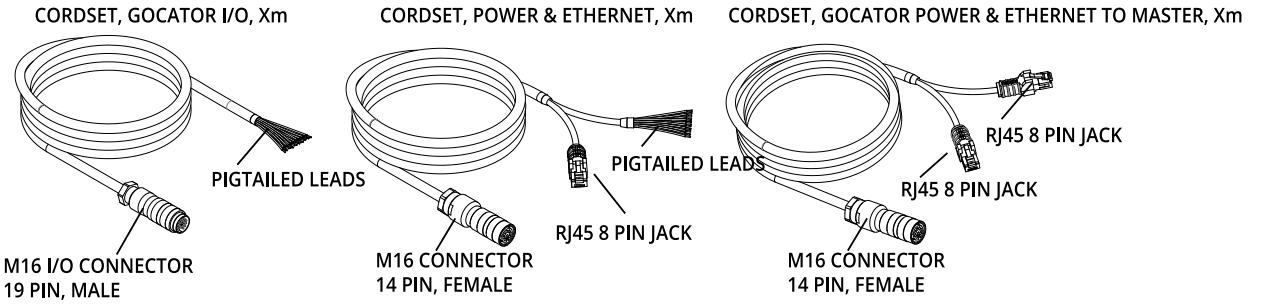
> When connecting cordsets to the sensor's connectors, hand-tighten only and do not exceed a torque of 2 N· m (18 in-lbs). Ensure that you properly secure the cordset cabling to avoid stress loading on the sensor connectors.

The Power & Ethernet cordset provides power. 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 1353.

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.

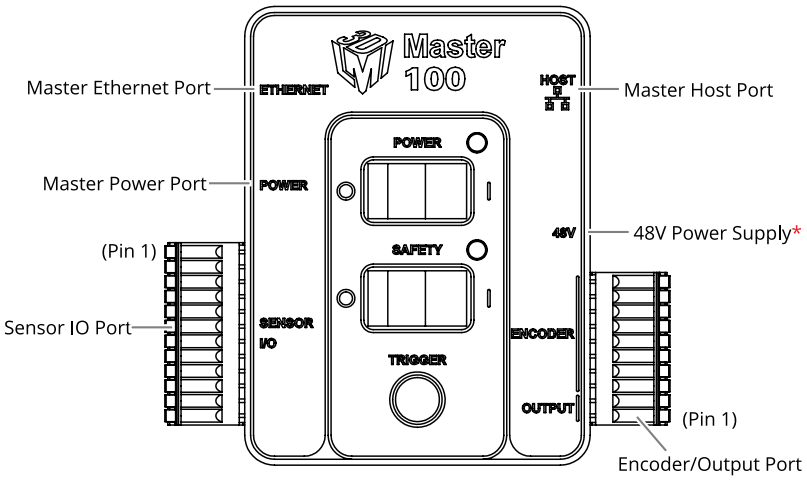


Cordsets are available with two types of cabling: standard and high flex. Use the high flex versions in applications where a tighter bend in the cordset is required, especially when the cordset must bend repeatedly. For more information, see *Cordset Bend Radius Limits* on page 32.

The maximum cordset length is 60 m. For pinout details, see *Gocator I/O Connector* on page 1348 and *Gocator Power/LAN Connector* on page 1346.

See *Accessories* on page 1374 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

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

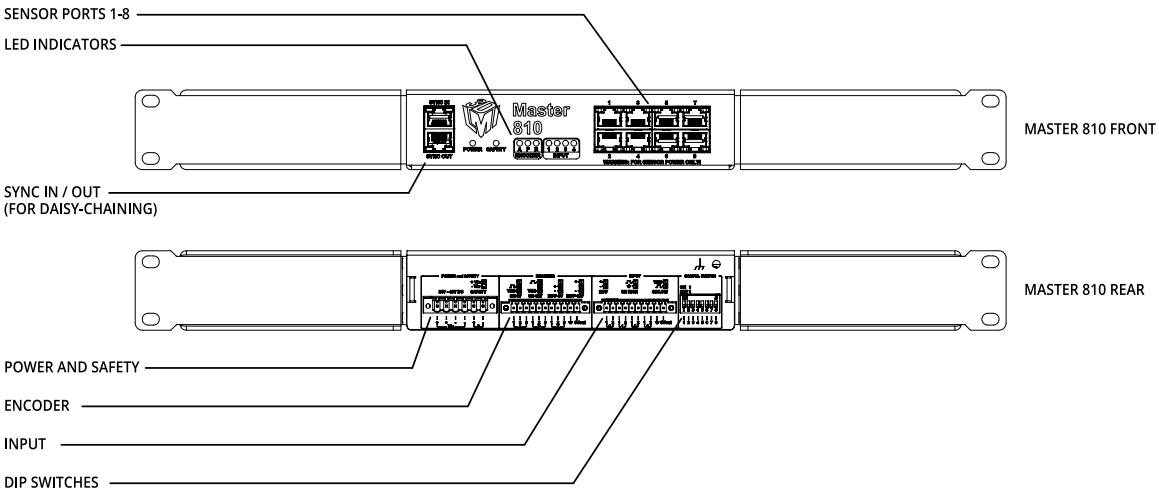
See *Master 100* on page 1353 for pinout details.

Master 810 / 2410

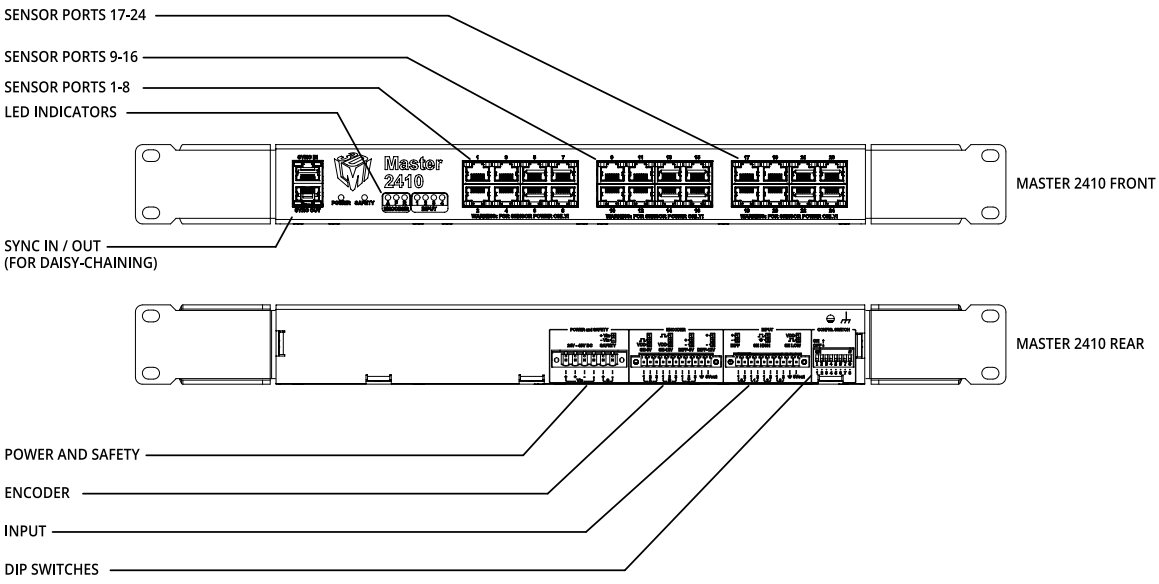
The Master 810 and 2410 network controllers let you connect multiple sensors to create a multi-sensor system:

- 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 38. (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 38.
LED Indicators	For more information, see *Master 810/2410* on page 1359.

For pinout details, see *Master 810/2410* on page 1359.

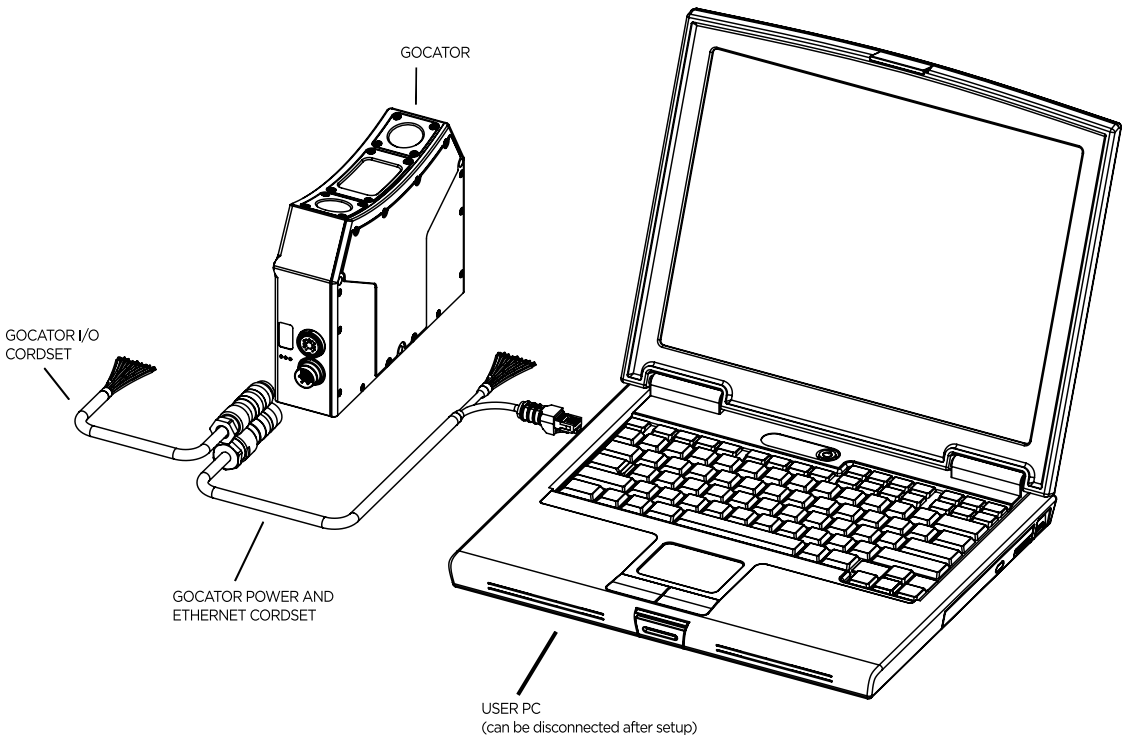
System Overview

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

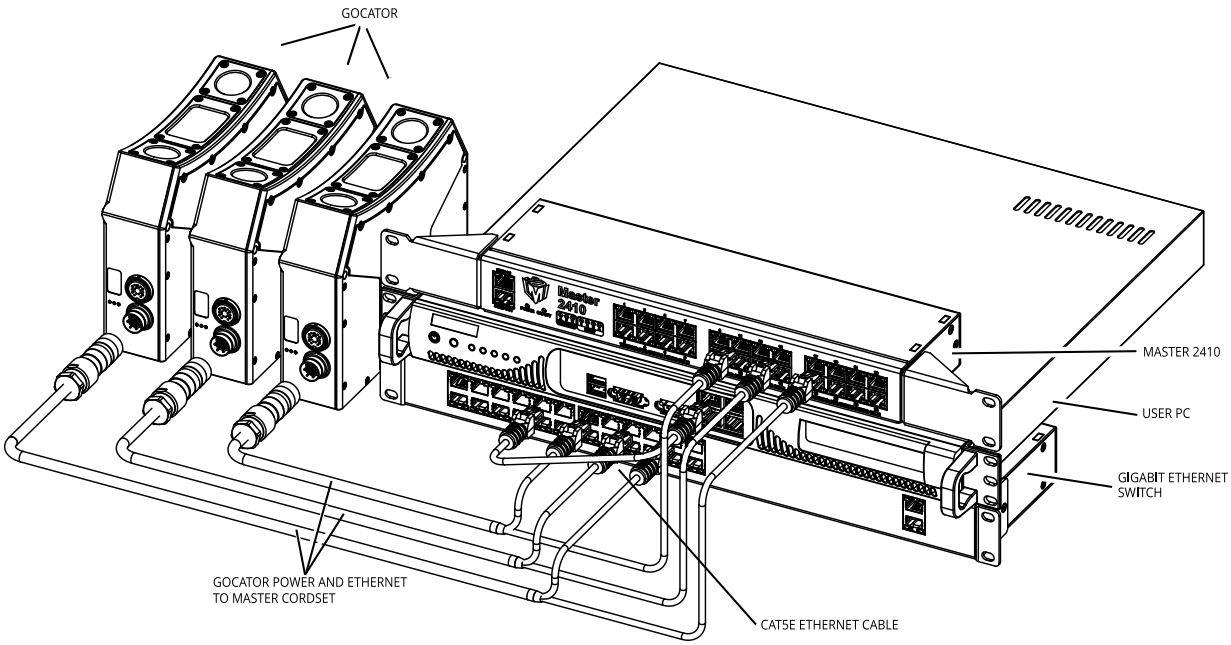
Standalone systems include only a single sensor. The device can be connected to a computer's Ethernet port for setup and can also be connected to devices such as encoders, photocells, or PLCs.

For information on physical installation, see *Installation* on page 30.



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.

> When connecting cordsets to the sensor's connectors, hand-tighten only and do not exceed a torque of 2 N· m (18 in-lbs). Ensure that you properly secure the cordset cabling to avoid stress loading on the sensor connectors.

Torque Recommendations **30**

Gocator 3210A or B, 3506B: 24 VDC Operation and Cordset Length **31**

Mounting **31**

Cordset Bend Radius Limits **32**

Grounding **34**

Gocator 34

Recommended Practices for Cordsets 34

Master Network Controllers 35

Grounding When Using a DIN Rail (Master 810/2410) 35

Additional Grounding Schemes 36

Installing DIN Rail Clips: Master 810 or 2410 ..... **36**

Configuring Master 810 ..... **38**

Setting the Divider 39

Encoder Quadrature Frequency 39

Setting the Debounce Period 40

Torque Recommendations

The following table lists the recommended torque ranges for the Gocator sensor mounting holes. For the appropriate screw diameter and bolt hole diameter, pitch, and length, as well as recommended torque, see the sensor dimension drawings in *Specifications* on page 1331.

Mounting hole spec	Depth (mm)	Torque range (N·m)
M3x0.5	Depth 3	0.5 - 1
M4x0.7	Depth 2.25	1 - 1.75
M4x0.7	Depth 3	1 - 1.75
M5x0.8	Depth 5	2.7 - 4.16
M6x1.0	Depth 5	3.8 - 5.97

Mounting hole spec	Depth (mm)	Torque range (N·m)
M8x1.25	Depth 5	6 - 10.68
1/4-20 UNC	Depth 5	3.8 - 5.97

Gocator 3210A or B, 3506B: 24 VDC Operation and Cordset Length

Although LMI recommends operating Gocator 3210A, 3210B, and 3506B at 48 VDC to take full advantage of their capabilities, these sensors can operate when connected to a 24 VDC power supply. This may sometimes be necessary, for example, because of local regulations. However, when running these sensors with a 24 VDC power supply, the projector intensity is reduced and as a result, you may need to increase exposure time with darker targets to compensate for the reduced projector intensity. Gocator 3506A can also run at 24 VDC but is not subject to these limitations. For information on setting exposure, as well as the impact of running the sensor at 24 VDC on projector intensity, see *Gocator 3210A or B, 3506B on 24 VDC* on page 151.

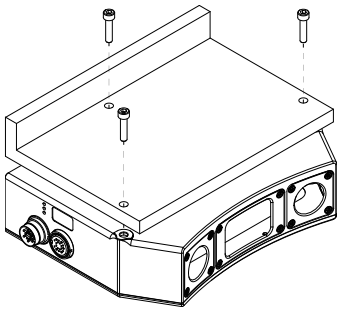
Note that because projector intensity decreases with longer cordsets (caused by the increased resistance and higher current), if limiting exposure time is important, try to limit the length of the cordset.

Finally, note that if you run these sensors at 24 VDC, you must specify this voltage and the cordset length in the web interface. For more information, see *Configuring Power* on page 119.

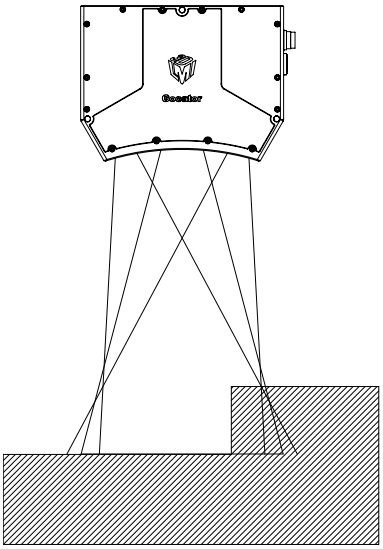
Mounting

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.

⚠ When mounting a sensor, ensure that the internal threads are not damaged from cross-threading, improper insertion of screws, or excessive torque. For the appropriate screw diameter and bolt hole diameter, pitch, and length, as well as recommended torque, see the sensor dimension drawings in *Specifications* on page 1331.



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.

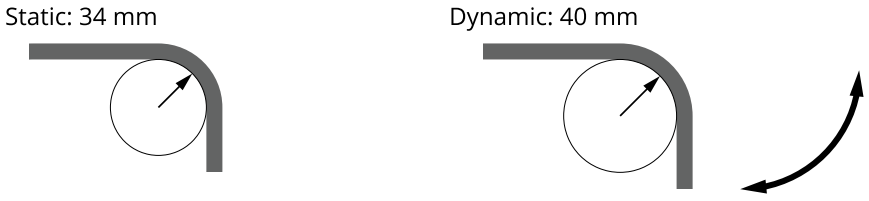
⚠	Gocator sensors are high-accuracy devices. The temperature of all of its components must be in equilibrium. When the sensor is powered up, a warm-up time of at least one hour is required to reach a consistent spread of temperature within the sensor.

For more information on heat considerations, see *Environment and Lighting* on page 19.

Cordset Bend Radius Limits

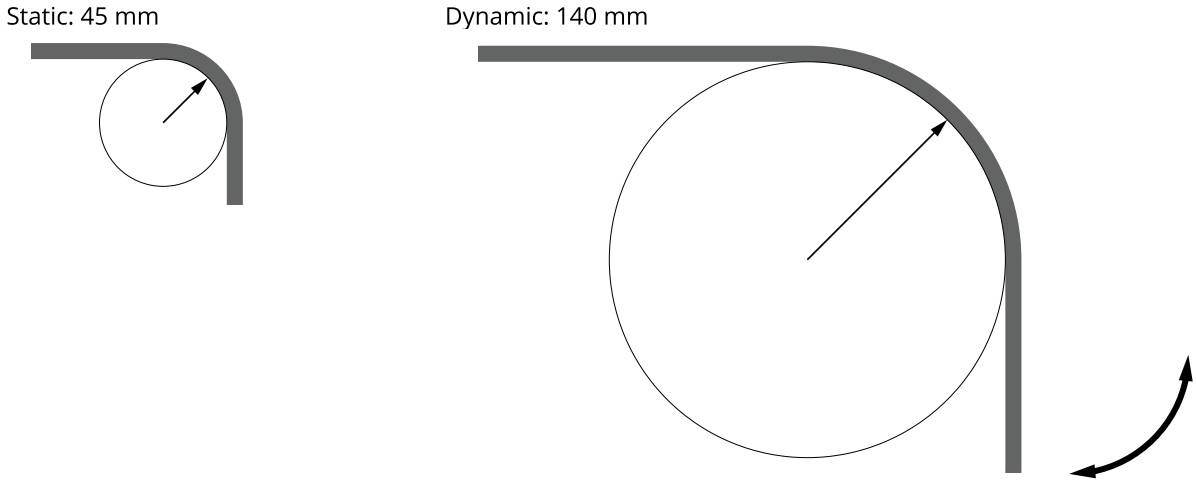
With high flex cordsets of lengths 25 meters and lower, limit bends as follows:

- In installations where a cordset does not bend continuously, limit bending to the static bend radius of 34 mm.
- In installations where a cordset bends continuously, limit bending to the dynamic bend radius of 40 mm.



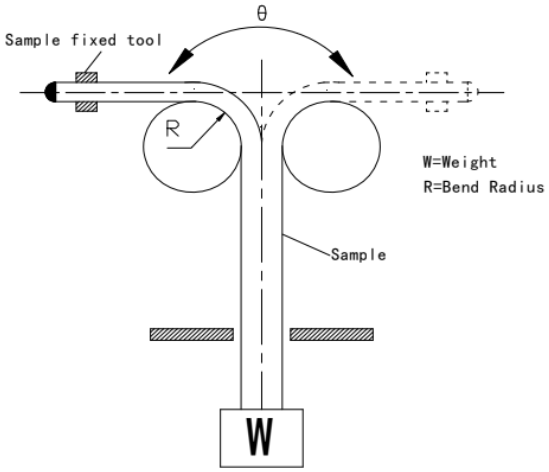
High flex cordset bend radius limits

Custom cordsets between 25 and 60 meters (the maximum length available) have a static bend radius limit of 45 mm and a dynamic limit of 140 mm.

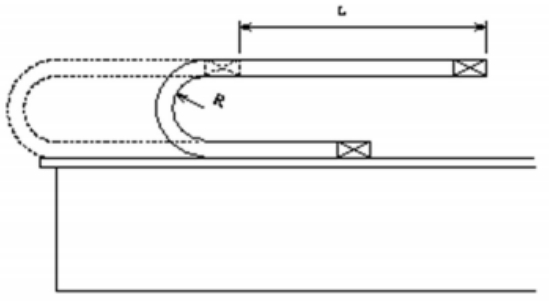


Standard cordset bend radius limits

High flex cordsets are rated for a minimum of 2 million 90° Tick Tock bends and 7 million U-shaped bends, both at the dynamic bend radius limit of 40 mm. The following illustrations show the test setups used to determine the number of bends in high flex cordsets.



Tick-tock test setup (θ = 180°)



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

For cordset part numbers, see *Accessories* on page 1374.

> Standard (non high flex) cordsets, which are no longer available, have a static bend radius limit of 45 mm and a dynamic limit of 140 mm. Standard cordsets are rated for a minimum of 2 million 90° Tick Tock bends.

For more information on cordsets, see *Gocator 3x00 Cordsets* on page 24.

Grounding

Components of a sensor system should be properly grounded.

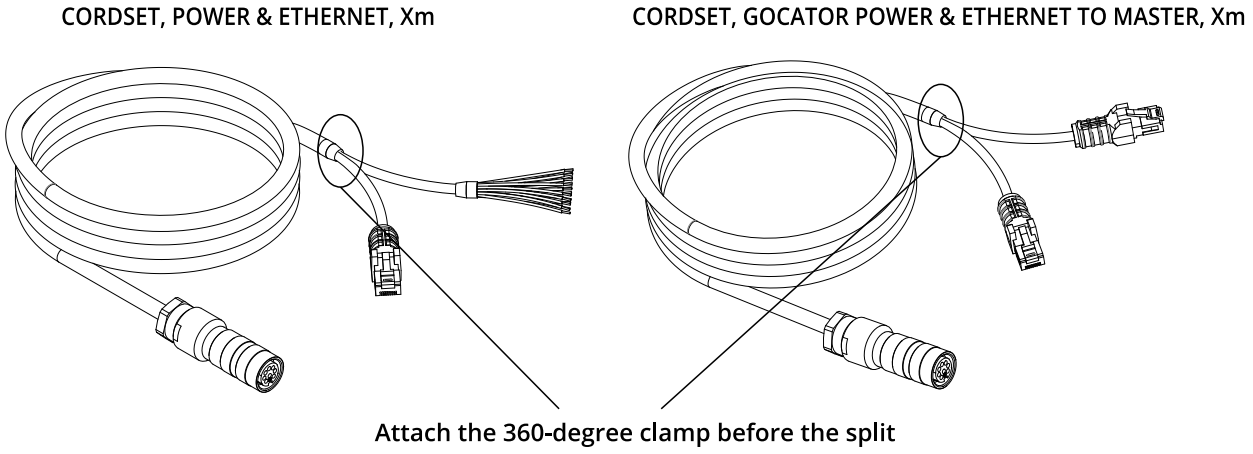
Gocator

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.

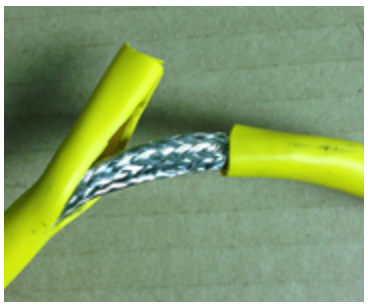
Recommended Practices for Cordsets

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.

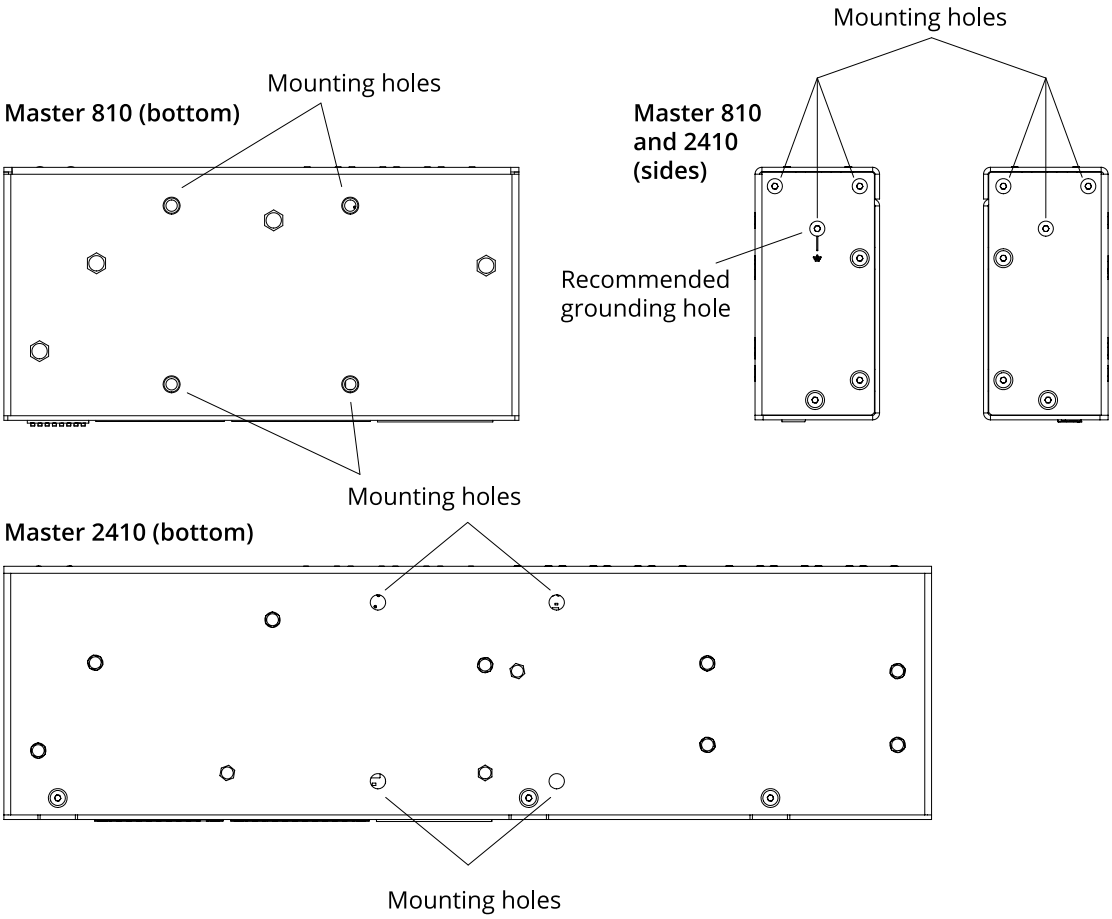
> You *must* check electrical continuity between the mounting frame and RJ45 connectors on the front using a multi-meter.

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:

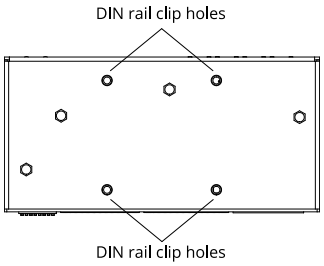


Older revisions of Master 810 and 2410 network controllers use a different configuration for the DIN rail clip holes.

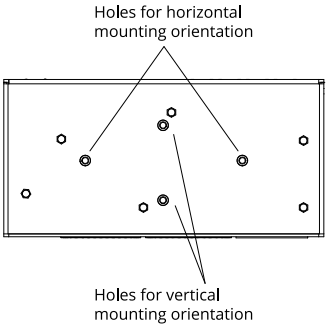
To install the DIN rail clips:

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

   Master 810:

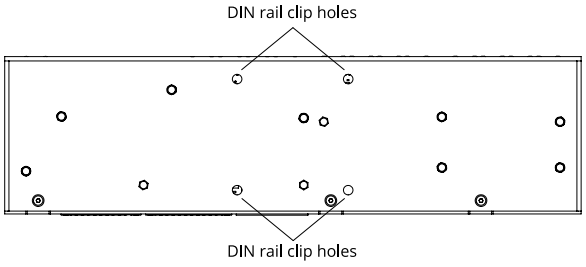


Current revision

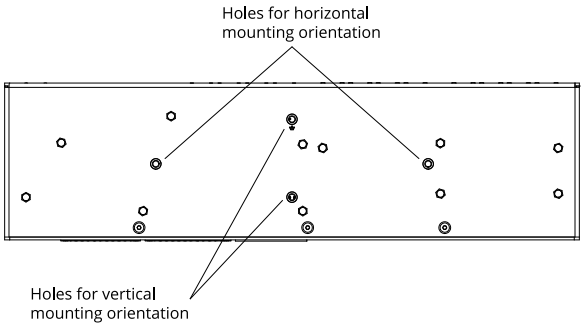


Older revision

   Master 2410:



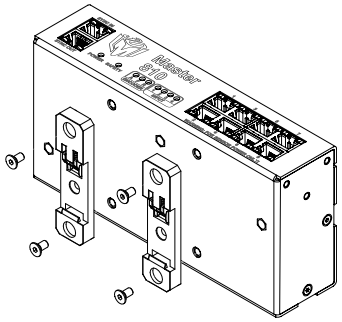
Current revision



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:



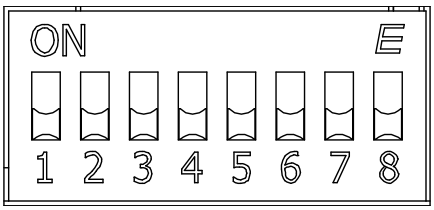
Ensure that there is enough clearance around the Master for cabling.

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* below) and a divider from the following table so that the Output Quadrature Frequency is no more than 300 kHz.

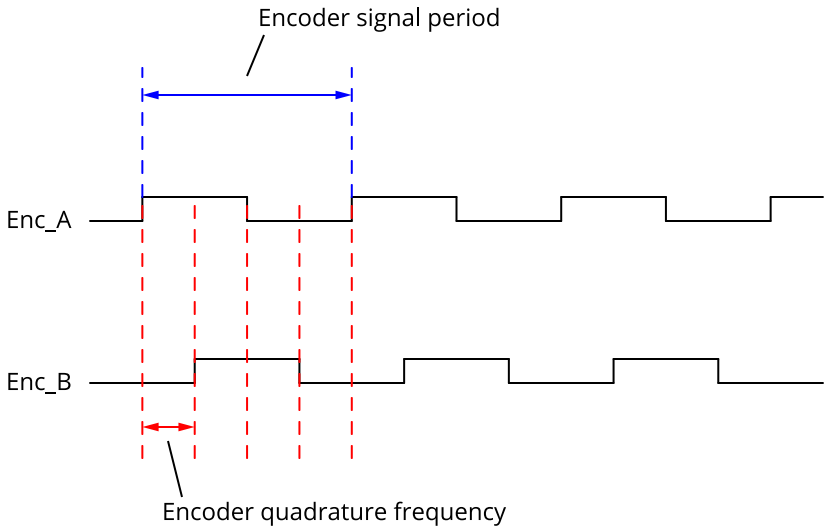
Divider	Switch 1	Switch 2	Switch 3
1	OFF	OFF	OFF
2	ON	OFF	OFF
4	OFF	ON	OFF
8	ON	ON	OFF
16	OFF	OFF	ON
32	ON	OFF	ON
64	OFF	ON	ON
128	ON	ON	ON



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

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 and Sensor 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.

Client Setup

To connect to a sensor from a client PC (using a web browser), you must ensure the client's network card is properly configured. The network ID of the client PC's network card must match that of any sensors you want to connect to.

Sensors are shipped with the following default network configuration:

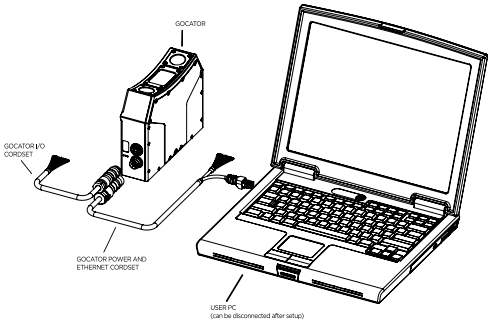
Setting	Default
DHCP	Disabled
IP Address	192.168.1.10
Subnet Mask	255.255.255.0
Gateway	0.0.0.0

So in order to connect to the sensor, you must set the network adapter of the client computer to 192.168.1.x, where x is unique on that network.

> When connecting cordsets to the sensor's connectors, hand-tighten only and do not exceed a torque of 2 N· m (18 in-lbs). Ensure that you properly secure the cordset cabling to avoid stress loading on the sensor connectors.

To connect to a sensor for the first time

1. Connect cables and apply power.



   Sensor cabling is illustrated in *System Overview* on page 27.
2. Change the client PC's network settings.

   Windows 11

   a. From the **Start** menu, launch the Settings app and click **Network & internet**.

   b. Click **Properties** of the desired network connection.

c. Next to **IP assignment**, click **Edit**.

d. Change the drop-down to Manual.

e. Enable IPv4.

f. Enter IP Address “192.168.1.5” and Subnet Mask “255.255.255.0”, then click **Save**.

Mac OS 11

a. Click Apple menu > **System Preferences**, and then click **Network**.

b. In the list to the left, select **Ethernet**.

c. Click **Advanced**, click **Hardware**, click the **Configure** pop-up menu, and set it to “Manually”.

d. Enter IP Address “192.168.1.5” and Subnet Mask “255.255.255.0”, and then click **Apply**.

⚠ See *Troubleshooting* on page 1376 if you experience any problems while attempting to establish a connection to the sensor.

Gocator Setup

Gocator sensors are 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.

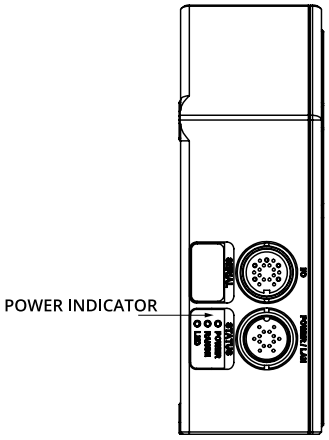
Standalone Sensor System

> When connecting cordsets to the sensor's connectors, hand-tighten only and do not exceed a torque of 2 N· m (18 in-lbs). Ensure that you properly secure the cordset cabling to avoid stress loading on the sensor connectors.

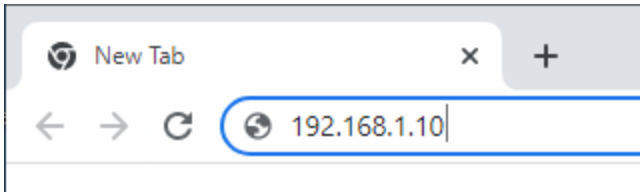
To configure a standalone sensor system

1. Power up the sensor.

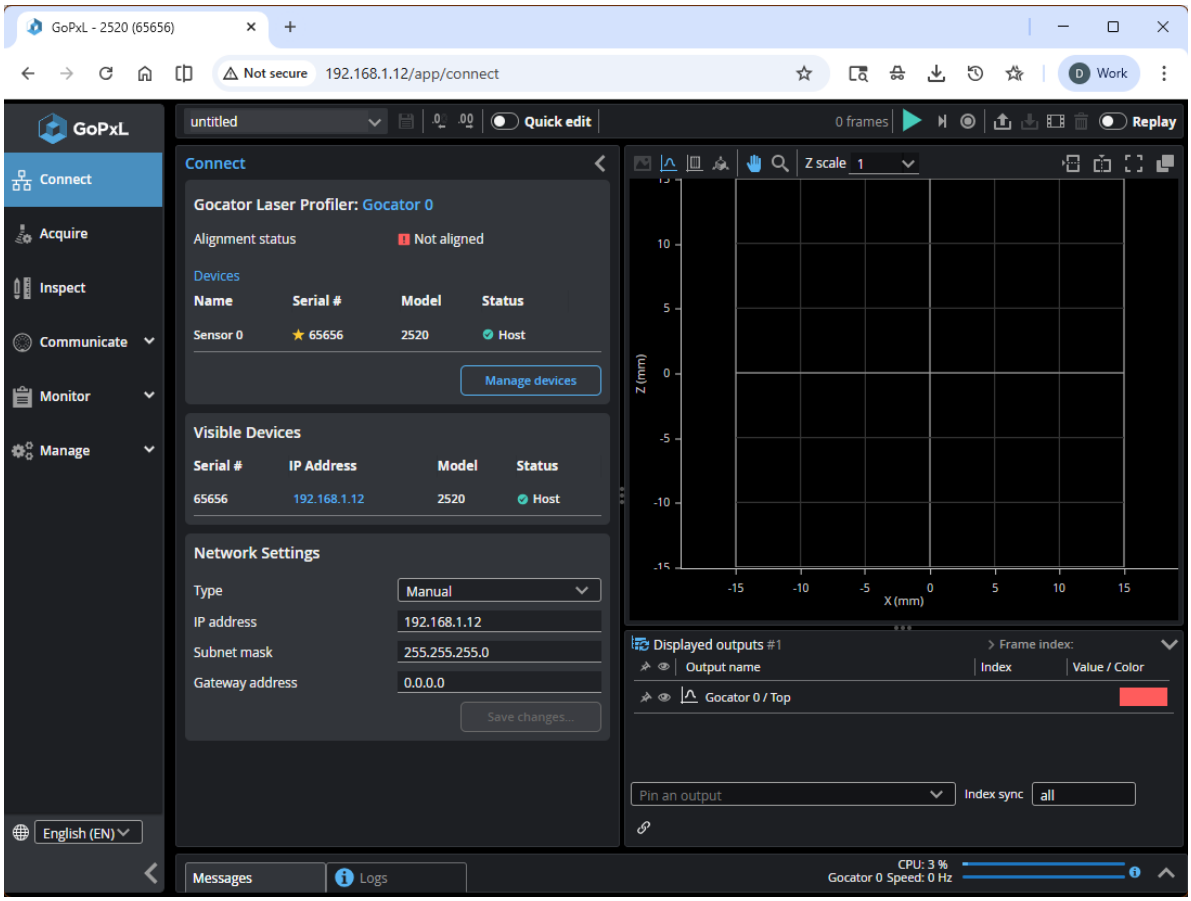
   The power indicator (blue) should turn on immediately.



2. Enter the sensor's default IP address (192.168.1.10) in a web browser.



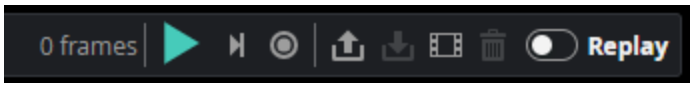
   The sensor interface loads.



If you can't connect to the sensor's interface, use the Discovery tool to find it on your network (for more information, see *GoPxL Discovery Tool* on page 1290) and use the IP address of the sensor.

3. Ensure that Replay mode is off

   Replay mode is off when the Replay toggle above the data viewer is to the left.



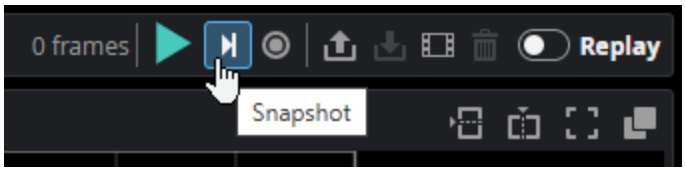
   For an overview of the user interface, see *User Interface Overview* on page 65.

4. Move a target under the sensor within its measurement range and field of view (FOV).
5. Press the "Start" or the "Snapshot" button on the Toolbar to start the sensor.

   The Start button is used to run a sensor continuously.



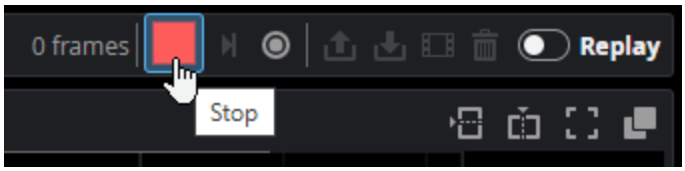
   The Snapshot button is used to trigger the capture of a single frame.



If a target object is within the sensor's measurement range and FOV, 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 1376.

6. If you started the sensor using the "Start" button, press the Stop button.



Required Ports

The table below lists the ports used by the following:

- sensors
- the Ethernet-based protocols
- the SDK and the REST API
- PC instances of GoPxL

Use this information to decide whether you or your network administrator needs to open network ports and to understand the traffic that a system will produce over a network.

Ports used by GoPxL

Port	Data Packet Protocol	Description
80	TCP	GoPxL web port for GoPxL GUI. **Sensors and GoMax**: All devices use this port. **For GoPxL running on a PC**: Use port 8100; for more information, see below.
502	TCP	Used by the GoPxL Modbus protocol as the listener port.
2222	UDP	Used by the EtherNet/IP protocol for implicit messaging.
3192	TCP	FireSync physical sensor upgrade port.
3194	TCP	FireSync physical sensor health port.
3320	UDP	GoPxL discovery protocol server listener port
3500	UDP	GoPxL Remote Procedure Call port.
3600	TCP	GoPxL control port. **Physical sensors and GoMax**: All devices use this port. **GoPxL running on a PC**: By default, the first PC instance starts at

Port	Data Packet Protocol	Description
		this port number, and subsequent instances are offset (for example, 3600, 3620, and so on). You can set custom values in GoPxL Manager. For more information, see *Running GoPxL on a Windows PC* on page 1188.
3601	TCP	Gocator Data Protocol (GDP) port. **Physical sensors and GoMax**: All devices use this port. **GoPxL running on a PC**: This port is offset from the control port by +1 (for example, 3601, 3621, and so on).
8080	TCP	Default accelerated sensor web server port number.
8100		GoPxL web port for GoPxL GUI for PC instances of GoPxL. (Sensors and GoMax units use port 80.) By default, the first PC instance starts at this port number, and subsequent instances are offset (for example, 8100, 8120, and so on). You can set custom values in GoPxL Manager. For more information, see *Running GoPxL on a Windows PC* on page 1188.
8190	TCP	Used by GoPxL Ethernet for default ASCII control, data, and health port. Actual ports can be configured by the user.
44818	TCP and UDP	Used by the GoPxL EtherNet/IP protocol as a UDP and TCP listener port for explicit messaging.

Key Concepts

The following sections provide an overview of how devices acquire and produce data, detect and measure parts, and control 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 PC-based version of GoPxL to speed up the processing of data.

Sensors, Sensor Groups, and Systems

In GoPxL, the term *sensor* refers to a single device. A *sensor group* contains two or more sensors of the same model. The term "system" refers to the sensor or sensor groups as a whole. Currently GoPxL only supports a single sensor group in a system.

> In this document, we will use the term multi-sensor to describe systems with *two* or more sensors.

In a multi-sensor system, one sensor controls the other sensors, and combines the scan data of the individual sensors. So for example, in a multi-sensor G2 system, the profiles from the individual sensors are combined into a single, wider profile.

> In Gocator Classic, we use the terms "buddy" or "buddying" to refer to the controlled sensors in a sensor group.

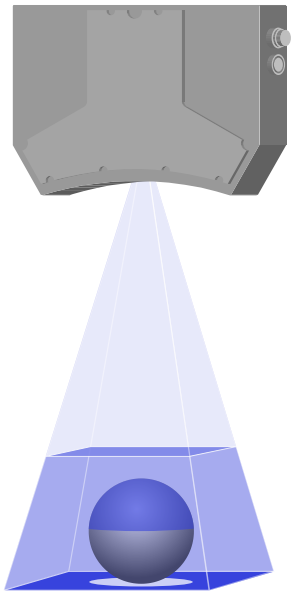
Sensors and sensor groups also have an associated "scan engine." For example, when designing a sensor system containing one or more G2 sensors, the associated scan engine is "Gocator Laser Profiler," with acquisition settings specific to that family of sensors.

> You may still see the term "scanner" in the GoPxL UI, SDK, and REST API. "Scanner" is equivalent to "sensor group."

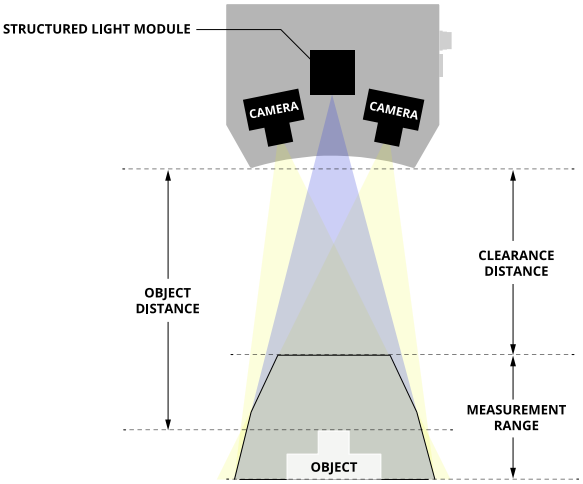
3D Acquisition

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

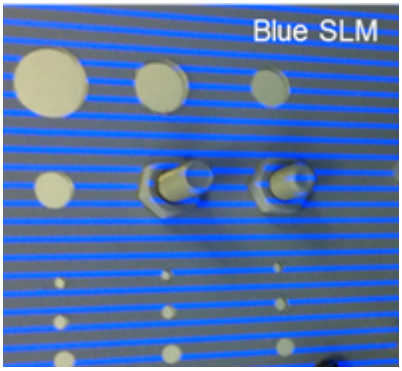
3x00 series sensors are 3D snapshot sensors, meaning they capture an entire surface in 3D in a single snapshot. These sensors project several structured light patterns in a rapid sequence onto the target. The reflection of the pattern off the target is captured by two cameras. The target must remain 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.



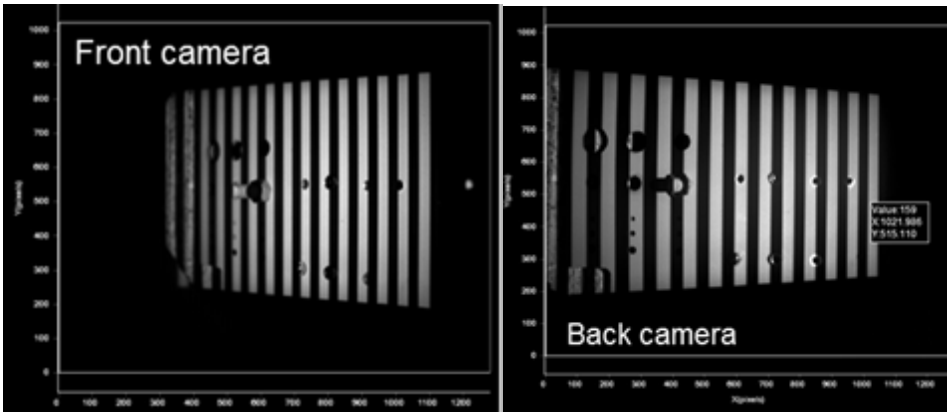
A Structured Light Modulator (SLM) produces a sequence of high resolution/high contrast light patterns using a blue LED. Two cameras capture the reflected light pattern from different viewing angles. The sensor can then use either stereo correlation or independent triangulation to generate 3D points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



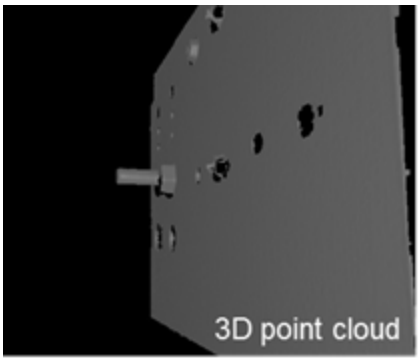
These are the steps of the acquisition of a 3D point cloud:



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 (mm) throughout their measurement range.

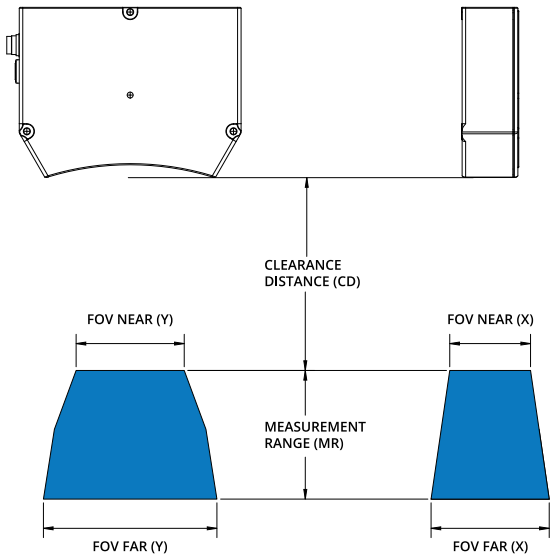
Clearance Distance, Field of View and Measurement Range

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 141 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.

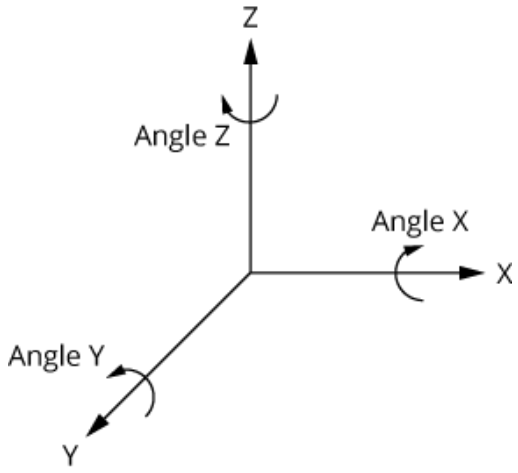
Coordinate Systems

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

- **Unaligned (sensor) coordinates**: Used on unaligned sensors.
- **Aligned (system) coordinates**: Used on aligned sensors.

Understanding coordinate systems is an important part of understanding measurement results.

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).

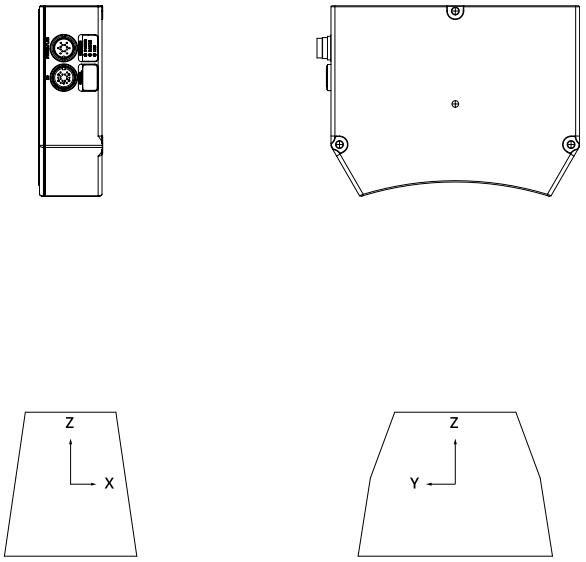
X offset, Y offset, and Z offset define the translations from the origin.

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.

Unaligned Coordinates

Unaligned sensors use *sensor* coordinates, that is, the coordinate system is relative to the sensor itself. 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).



Aligned Coordinates

Understanding aligned 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. For more information on aligning sensors, see *Aligning Sensors* on page 122.

The adjustments resulting from alignment are called *transformations* (offsets along the axes and rotations around the axes). Transformations are displayed on the **Acquire** page, in the **Sensor Properties** section, under **Transform**. Note that currently, you can ignore Sensor Group Transforms, which represent the transformations between *groups* of sensors; support for this will be available in future versions of the sensor software. For more information on transformations in the web interface, see *Transformations* on page 137.

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

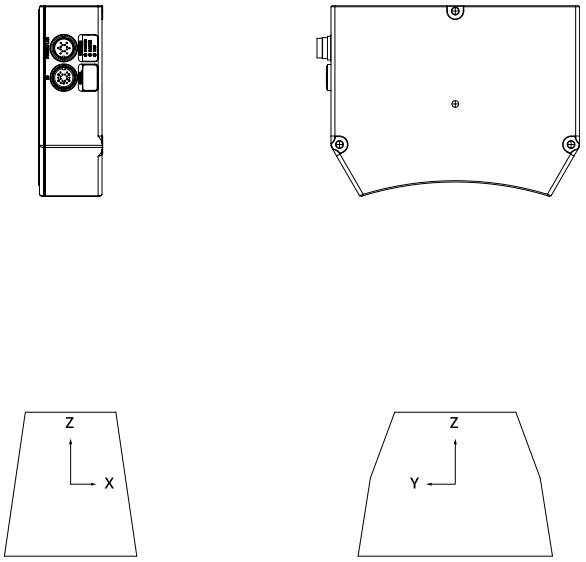
Alignment can be used to establish a transformed coordinate system according to the user's needs. Alignment determines the adjustments to X, Y, and Z, as well as rotation angle around each axis. Transformed coordinate systems can be associated with specific sensor jobs. For details, see *Aligning Sensors* on page 122.

When applying the transformations, the data is first rotated around X (clockwise, with the X axis toward the viewer), then Y (counterclockwise), and then Z (clockwise), and then the offsets are applied.

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 *Aligning Sensors* on page 122.

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 137.

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 *Aligning Sensors* on page 122.

When applying the transformations, the data is first rotated around X (clockwise, with the X axis toward the viewer), then Y (counterclockwise), and then Z (clockwise), 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).

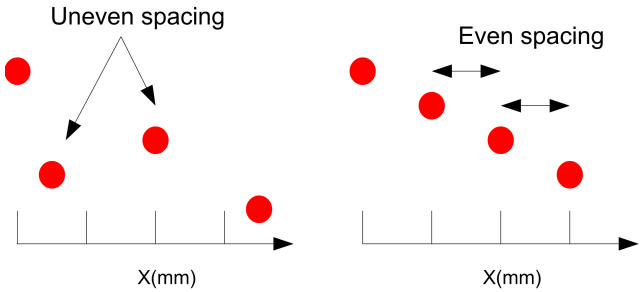
> The **Frame of Reference** setting, in the **Part Detection** panel on the **Scan** page, controls whether part data is recorded using sensor/system coordinates or part coordinates.

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).

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 **Enable uniform spacing** setting is disabled, the sensor returns this point cloud.

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 **Spacing Interval X/Y** setting; for details, see *Triggers* on page 141. 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

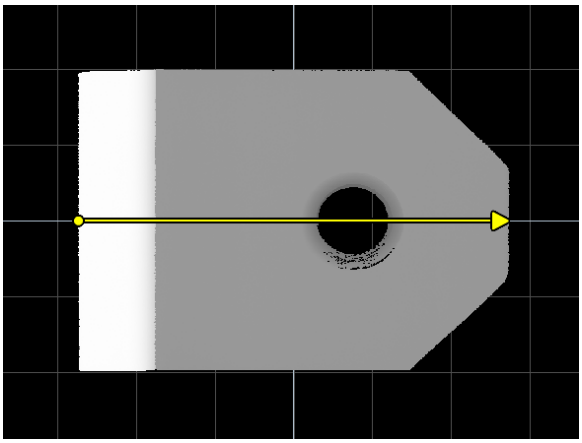
Part Detection

A sensor can isolate discrete parts on a surface into separate scans representing parts. For more information, see *Surface Part Detection* on page 908.

Gocator can then perform measurements on these isolated parts.

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 *Surface Section* on page 943.

Part Matching

The sensor can match scanned parts to the edges of a model based on a previously scanned part using the Surface Pattern Matching tool (for more information, see *Surface Pattern Matching* on page 913) or to the dimensions of a fitted bounding box that encapsulates the part data (for more information see *Surface Bounding Box* on page 774). When parts match, the sensor can rotate scans so that they are all oriented in the same way, using Surface Transform (for more information, see *Surface Transform* on page 993). This allows measurement tools to be applied consistently to parts, regardless of the orientation of the part you are trying to match.

Measurement and Anchoring

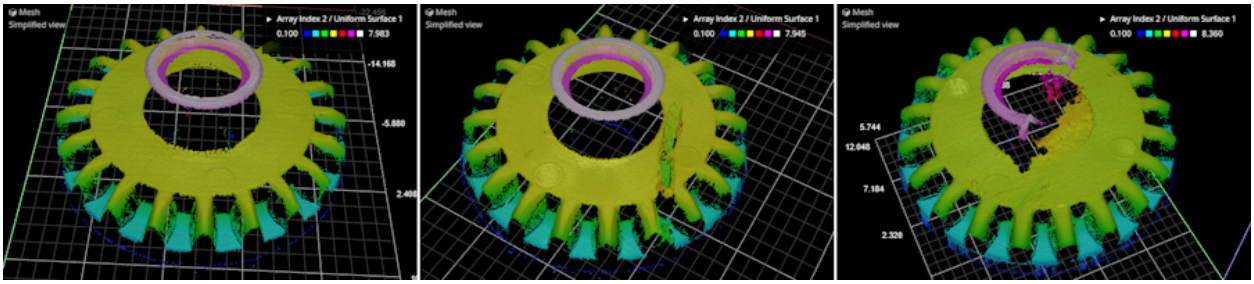
After GoPxL has acquired data and, optionally, further processes the data, GoPxL is ready to take measurements on the scan data.

GoPxL provides dozens of 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 outputs to control devices such as PLCs, which can in turn control ejection or sorting mechanisms. (For more information on measurements and configuring measurements, see *Inspect - Measurement and Processing Tools* on page 162.

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. For more information, see *Measurement Anchoring* on page 209.

Complex Defect Detection

Many applications rely on more easily quantified differences part to part, such as a simple pass / fail of the height of a feature on a part. In other cases, flaws or defects are much more random and difficult to characterize so simply: a part or material can be damaged or malformed anywhere on its surface. For example, the following images show scan data of a good part, and two defective parts:



GoPxL Anomaly Detector, an AI-based toolset, lets you train a model on representative good ("OK") and bad ("NG") frames of acquired data. After, when the device scans parts in production, the tool uses the trained model in its inference engine to classify the new scan data as either OK or NG, resulting in flexible defect detection. It is also possible to refine the model with other representative frames later on, and even produce simulated variations and flaws / defects to further improve the model's accuracy. For more information, see *Surface Anomaly Detector* on page 716.

Tool Chaining

GoPxL'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 sections describe the types of output and how you use them as input.

Tool Data

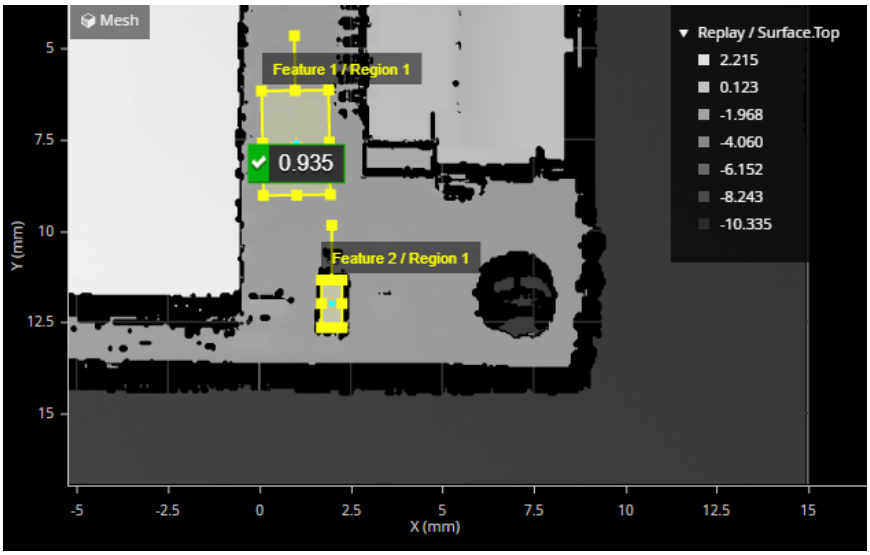
Most measurement and processing tools can output Profile or Surface data, which can be used as input by other tools or SDK applications.

Profile and Surface data produced by a tool is 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 *Surface Transform* on page 993.

Anchoring Measurements

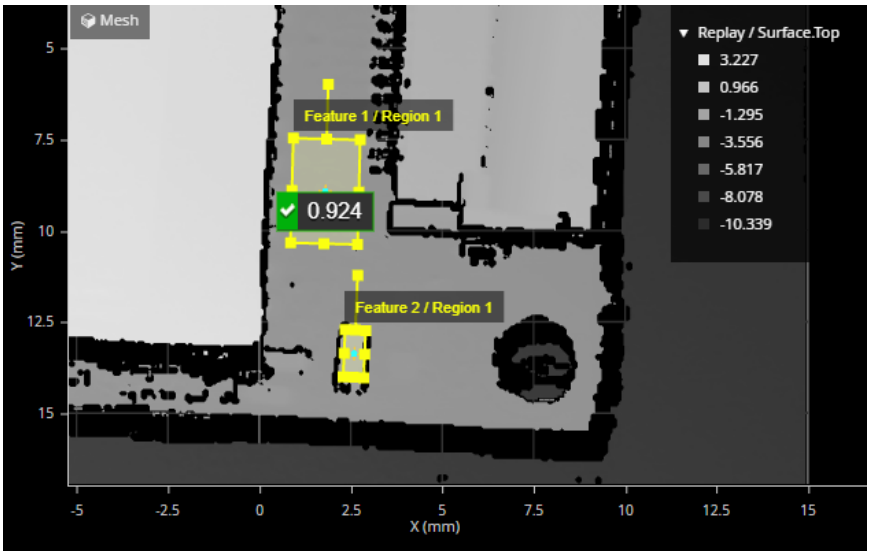
Tools can use the positional measurements (X, Y, or Z) of other tools as anchors to compensate for minor shifts of parts: anchored tools are “locked” to the positional measurements of the anchoring tool’s measurements. Some tools can also use a Z Angle measurement as an anchor. Typically, you will use 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.

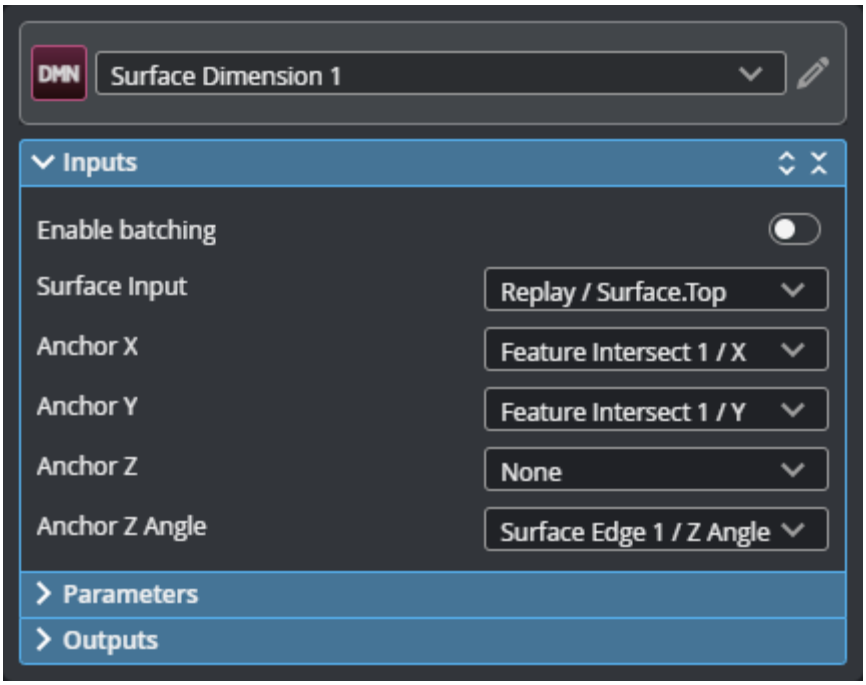


The height of a small PCB component (Feature 2) relative to nearby surface (Feature 1), anchored to positional (X and Y) measurements of the hole (lower right) and to the Z angle of the edge of a larger component to the left.

In the following, the part has rotated, but the tool's measurement regions follow the features, ensuring correct measurement.



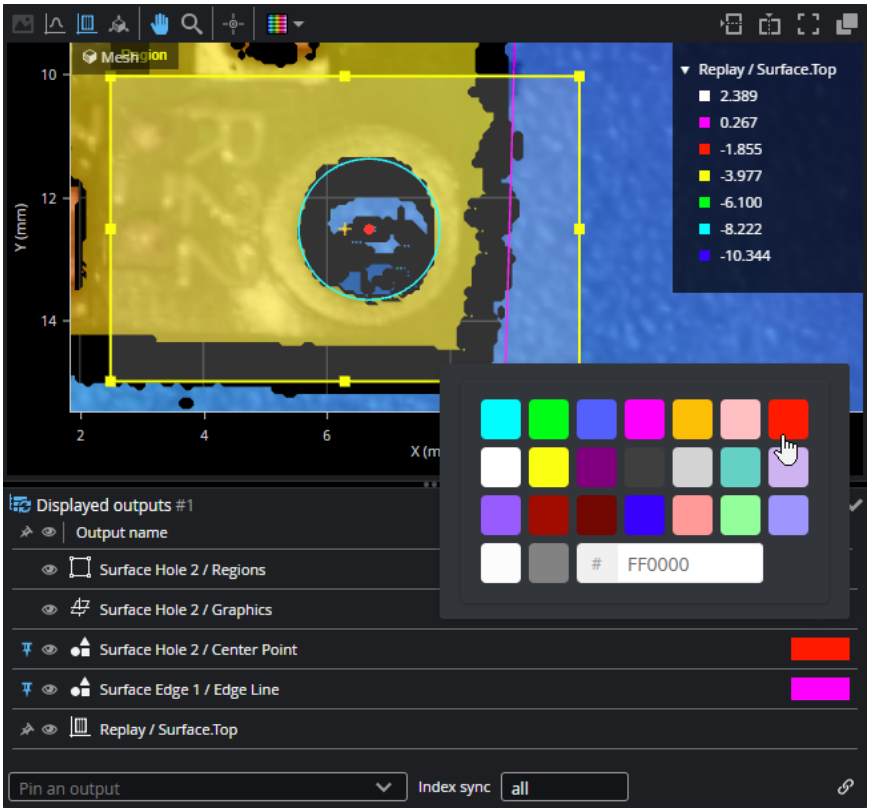
You enable anchoring in the expandable **Inputs** section of a tool:



Geometric Features

Many of GoPxL'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.

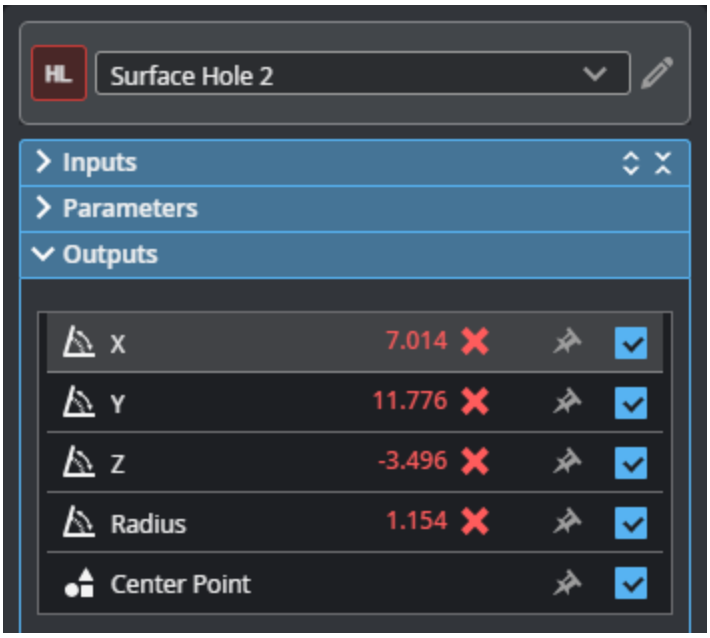
Geometric features are rendered as overlays on a tool's input data.



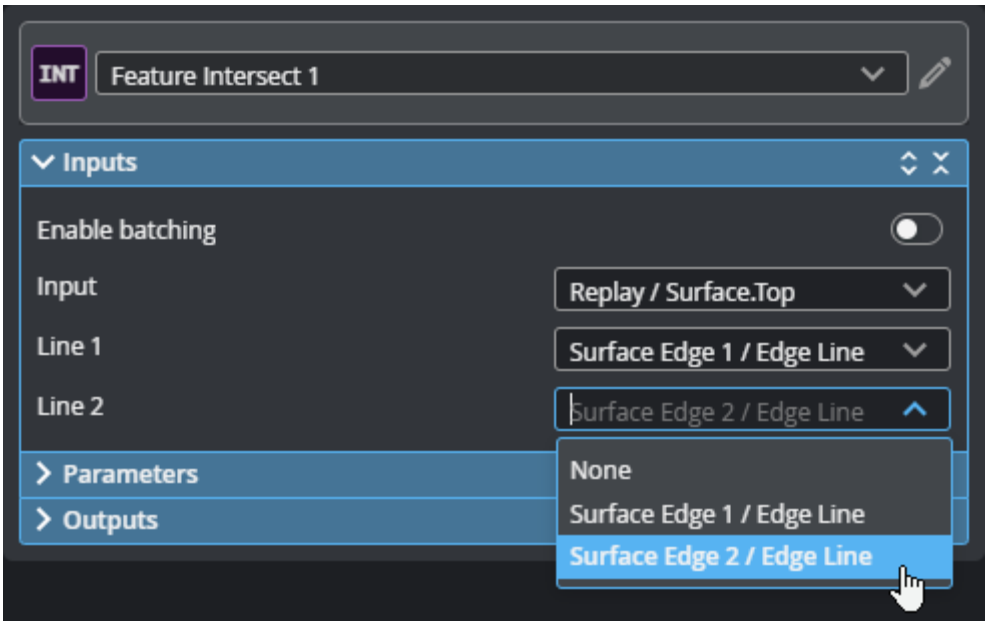
Point geometric feature (a hole's Center Point) rendered on a tool's input as a small circle; the color of geometric features is user-configurable.

GoPxL'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 Tools* on page 252.

You enable geometric feature *outputs* in a tool's expandable **Outputs** section:

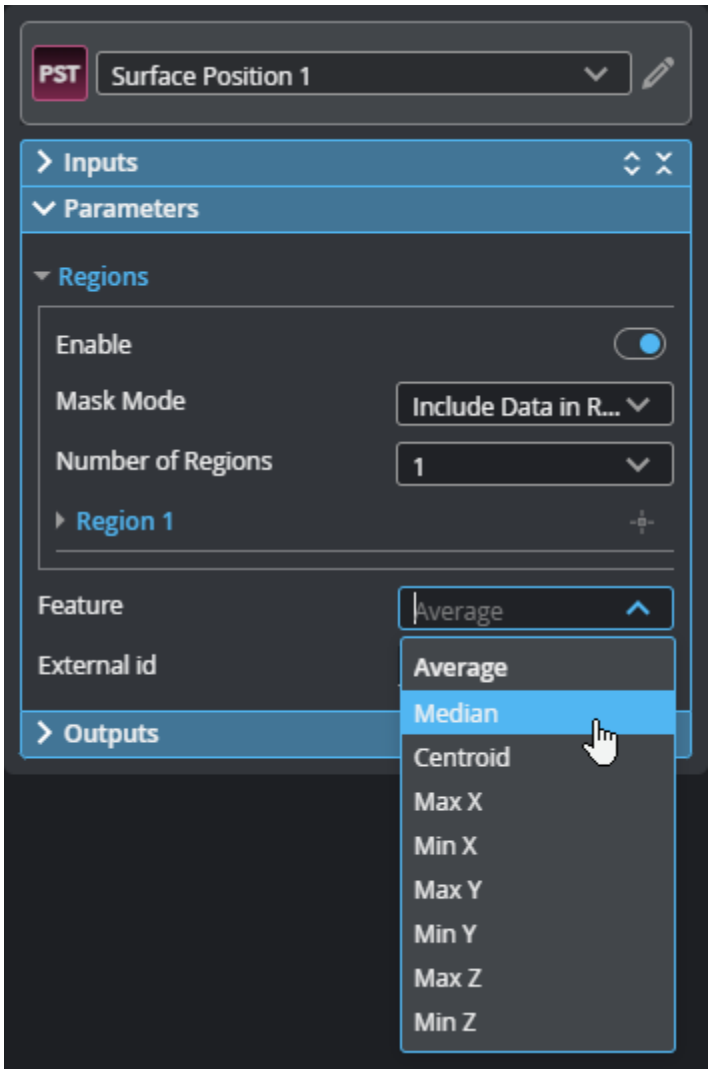


You enable geometric feature *inputs* in a tool's expandable **Inputs** section:



Setting the Line inputs to two different lines to compute their intersection

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:



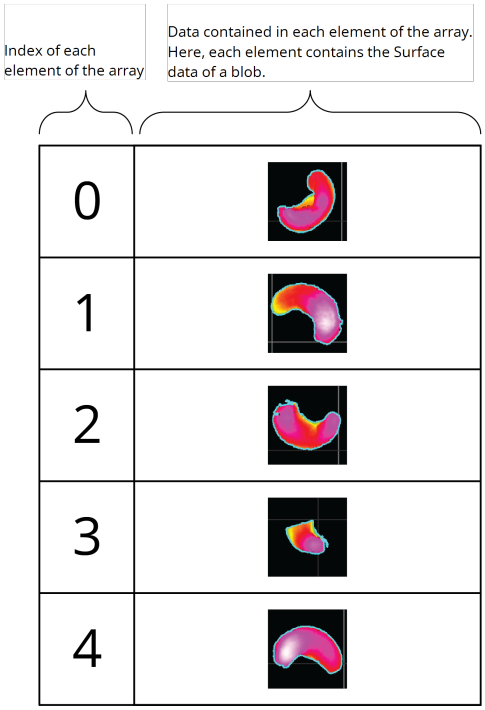
For more information on feature points, see *Feature Points* on page 224.

Arrays, Batching, and Aggregation

GoPxL supports data structures called arrays to help simplify otherwise complicated applications.

An array is a group of data items bundled into a single structure, such as multiple profiles or surfaces, multiple measurements, or multiple features.

For example, the individual parts in a frame of scan data can be isolated and stored in an array. Each part's scan data occupies on element of the array. Note that arrays start at 0. That is, they use zero-based indexing.



Each item, or element, in the array can be processed by other tools in one of two ways:

- Batching: Tools in batching mode process each array element in an array input separately.
- Aggregating: Aggregating tools combine each element in an array output and process it as a single piece of data.

For more information, see *Arrays, Batching, and Aggregation* on page 199.

Output and Control

After the sensor has scanned and measured parts, the last step in the operation flow is to output the results.

One of the main functions of GoPxL 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. Currently, GoPxL can communicate using the following over Ethernet:

- With PLCs (programmable logic controllers) using industry-standard protocols (EtherNet/IP, Modbus, PROFINET, and ASCII)

- With digital devices (such as ejector chutes or stack lights)
- With the GoPxL SDK and REST API, using the Gocator Communication Protocol

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, GoPxL 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." The sensor will know when a defective part has traveled far enough and trigger a PLC, for example, to activate an ejection/sorting mechanism at the correct moment.

For more information on configuring control, see *Communicate - PLC, Digital, GDP, HMI* on page 1018.

For more information on the GoPxL SDK and REST API, see *GoPxL SDK and REST API* on page 1234.

User Interface Overview

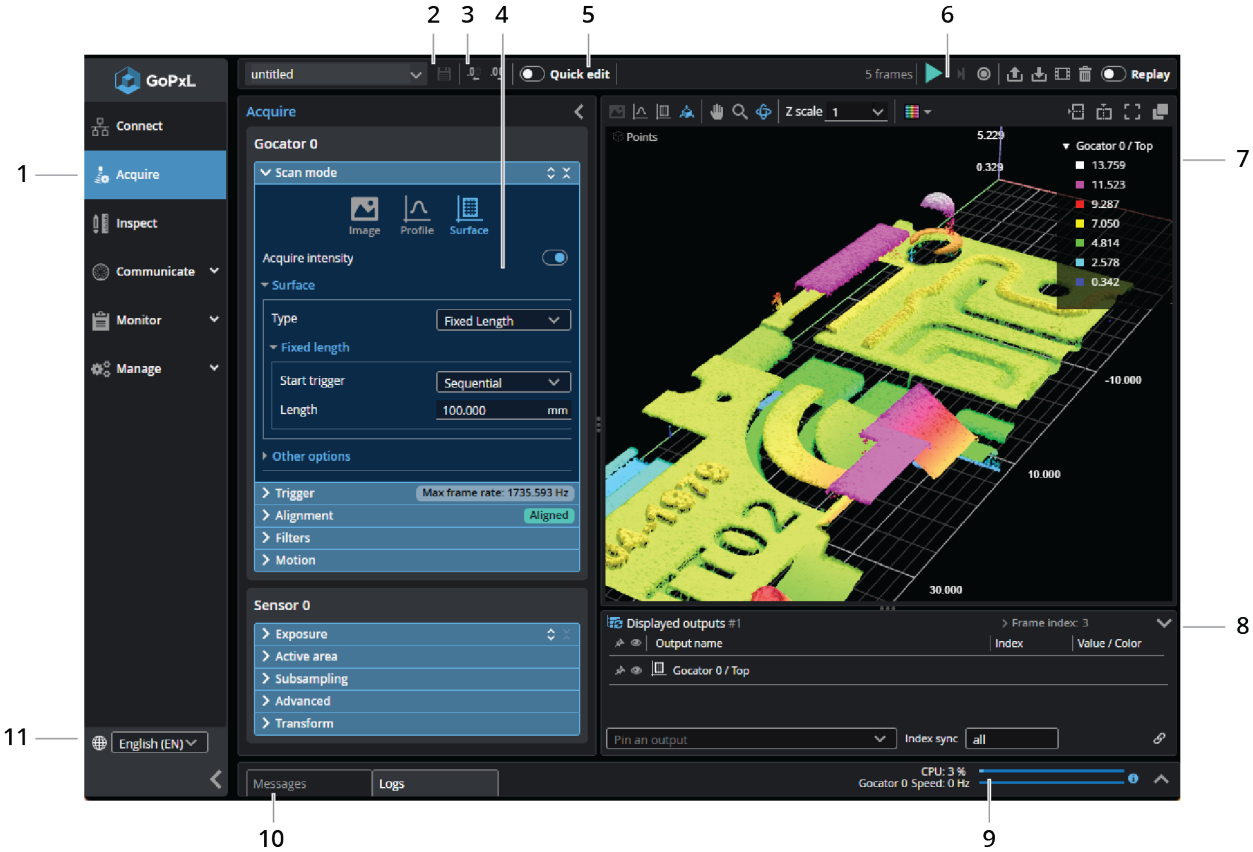
You configure sensors using GoPxL by connecting to an IP address with a web browser.

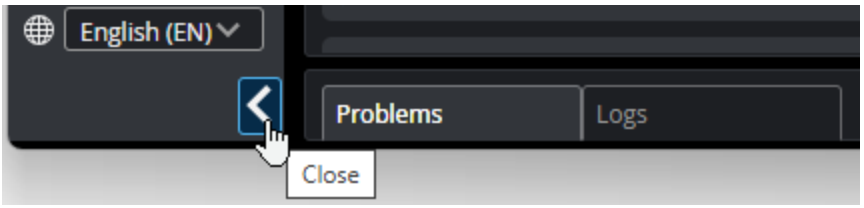
If you are running GoPxL on-sensor, in the browser, you connect to the IP address of the sensor. By default, the IP address of Gocator sensors is 192.168.1.10.

If you are running the Windows-based GoPxL application (to accelerate the sensor on a PC or view previously recorded scans), in the browser, by default you connect to the localhost address (127.0.0.1). For more information on running GoPxL on a PC, see *Running GoPxL on a Windows PC* on page 1188.

GoPxL provides categories in a vertical navigation bar on the left of the interface that group similar functions together in pages. Each page contains one or more panels that further categorize functionality. Generally speaking, the order of the categories and the pages reflects the workflow involved in configuring systems for an application for the first time, from the top down.

> Some UI screenshots in this manual may not depict the latest version of GoPxL. These screenshots will be updated soon.



	Element	Description
1	Navigation bar	The navigation bar (sometimes called the "nav" bar) along the far left of the user interface displays tabs for pages (for example, from the top, **Connect**, **Acquire**, **Inspect**, and so on) that contain related settings. Some pages contain sub-pages (for example, the **Manage** category contains the **Settings** and **Jobs** pages, among others). Each page contains one or more panels. The category and page currently selected in the navigation bar are highlighted. By default, the navigation bar is open and shows the category, as well as the page names and icons. To close the navigation bar, click the Close button at the lower left of the interface.  For information on the categories and what you can do with the pages they contain, see *What you can do in the main pages* on the next page.
2	Job-related functions	These controls let you work with jobs, for example, creating and saving jobs. For more information on jobs, see *Jobs* on page 1179.
3	Decimal places	Use the decimal places buttons to control how values are displayed throughout the interface. The values displayed in the interface are rounded to the chosen number of data places, but internally all decimal places are used. Click .0 to decrease the number of decimal places, down to zero decimal places. Click .00 to increase precision, up to six decimal places.
4	Configuration area	This area contains the configurations of the selected page and panel. When configuring tools, an additional **Tools Configuration** panel lets you configure a tool's parameters.
5	Quick Edit	When working with a very large number of measurement tools (for example, a few dozen) or a very complex tool (built-in or user-created GDK tool), you can enable Quick Edit mode to make configuration faster, as the data viewer is not refreshed after each setting change. You should enable this setting if changing parameters feels sluggish. It is important to disable this mode after you have finished making parameter changes to save the changes.
6	Global controls	These controls let you start and stop acquisition, and also work with replay data. For information on starting and stopping acquisition, see *Starting, Stopping, and Recording* on page 105.

	Element	Description
		For information on working with replay data, including adding and removing frames of replay data, see *Playing Back Recorded Data* on page 107.
7	Data viewer	The data viewer (sometimes called the data visualizer) displays scan data, providing several tools in its toolbar to control how data is visualized (such as top view, front view, or perspective view). The data viewer is available on all pages. You can split it horizontally and vertically, and create pop-out windows. When GoPxL is in Replay mode, a blue border is displayed around the data viewer. For more information, see *Data Viewer* on page 79.
8	Displayed outputs pane	This area lets you show, hide, and pin outputs such as measurements. You can also control how outputs are displayed. For more information, see *Displaying and Pinning Outputs* on page 94.
9	Metrics area	This area provides CPU usage and frame rate, including the calculated maximum frame rate. For more information, see *Metrics Area* on page 69.
10	System messages	This area provides log messages, error alerts and warnings. For more information, see *System Messages* on the next page.
11	Interface language	Lets you change the language of the GoPxL interface. To apply the change, GoPxL must restart. For GoPxL on PC, after changing the interface language, you must open GoPxL Manager, select the instance, and then stop and start it manually; for more information, see *Running GoPxL on a Windows PC* on page 1188.

The following lists the categories available in GoPxL and where you can learn how to work with them.

What you can do in the main pages

Page	Description
Connect	Lets you configure power options. Also lets you configure network settings. For more information, see *Connect - Creating a System & Networking* on page 119.
Acquire	Lets you configure acquisition. Also lets you perform alignment. For more information, see *Acquire - Alignment and Acquisition* on page 122.
Inspect	Lets you add and configure measurement and processing tools. For more information on tools, see *Inspect - Measurement and Processing Tools* on page 162.
Communicate	Lets you enable and configure the GoPxL Data Protocol and industrial protocols. Also lets you configure GoHMI. For more information, see *Communicate - PLC, Digital, GDP, HMI* on page 1018.

Page	Description
Monitor	Lets you monitor measurements, as well as sensor health and performance. For more information, see *Monitor - Measurement Stats, Health & Performance* on page 1167.
Manage	Lets you manage jobs, upgrade sensors, perform a factory restore, and so on. For more information, see *Manage - Sensor Management and Maintenance* on page 1175.

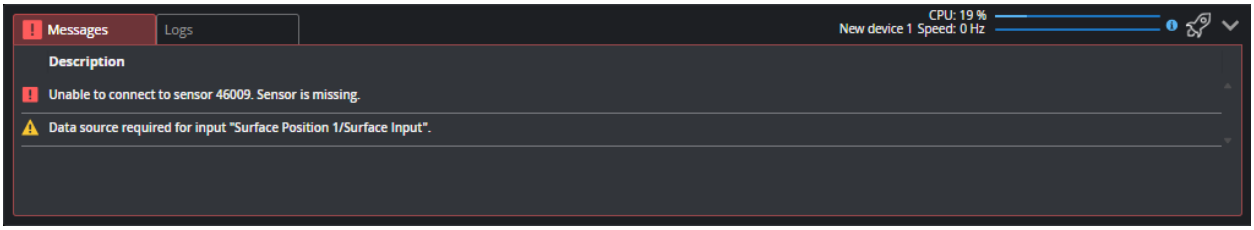
Status Bar

The status bar lets you do the following:

- Access system messages in the in the **System Messages** panel. For details, see *System Messages* below.
- See important system metrics. For details, see *Metrics Area* on the next page.

System Messages

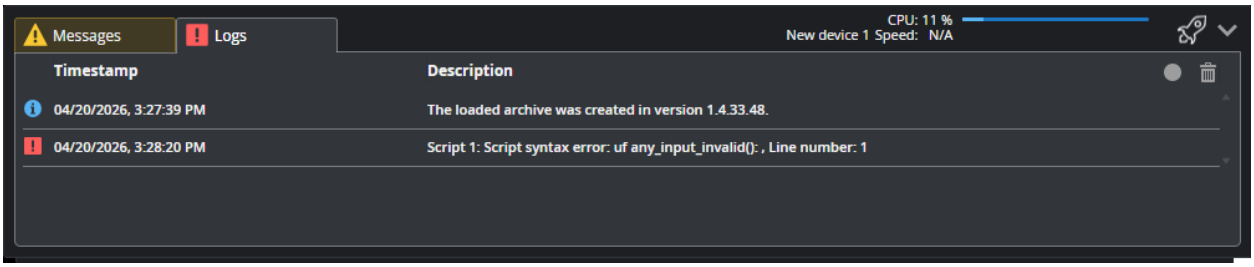
GoPxL displays current messages and a system log containing errors, warnings, and general information, in two tabs at the bottom of the user interface. The **Messages** tab lists the *current* issues. Once you fix an issue, it is removed from the **Messages** tab.

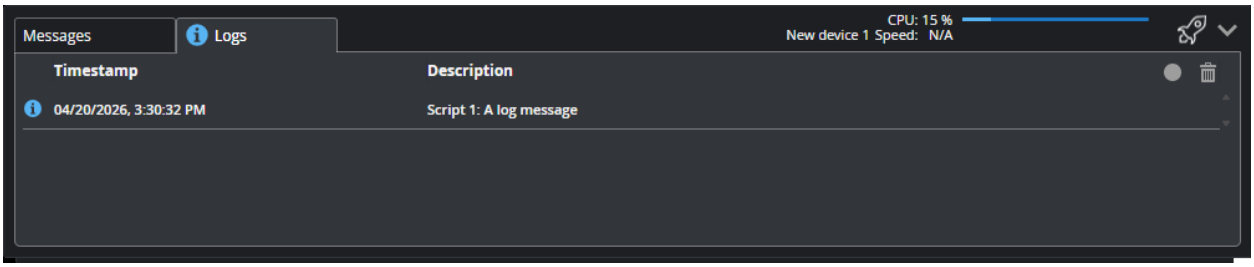


Messages tab showing a current problem with the system.

The **Logs** tab displays a complete list of system-level messages. You can filter the log entry types using the filter drop-down to the right on the **Logs** tab, and clear the entries using the eraser button.

Problems that prevent the system from running (they are blocking problems) are indicated with a square red icon ( ). Problems that are less serious (they are potential problems) are indicated with a yellow triangle icon ( ). Informational messages are displayed with a blue circle icon ( ).





Metrics Area

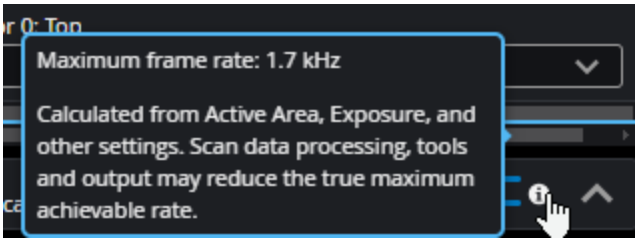
The Metrics area on the right side of the status bar displays two important performance metrics: CPU load and sensor speed (current frame rate).



The **CPU** bar in the Metrics area displays how much of the CPU is being used.

The **Speed** bar displays the frame rate of the sensor.

You can see the maximum calculated frame rate by clicking the info icon to the right of the metrics area.



Keyboard and Mouse Shortcuts

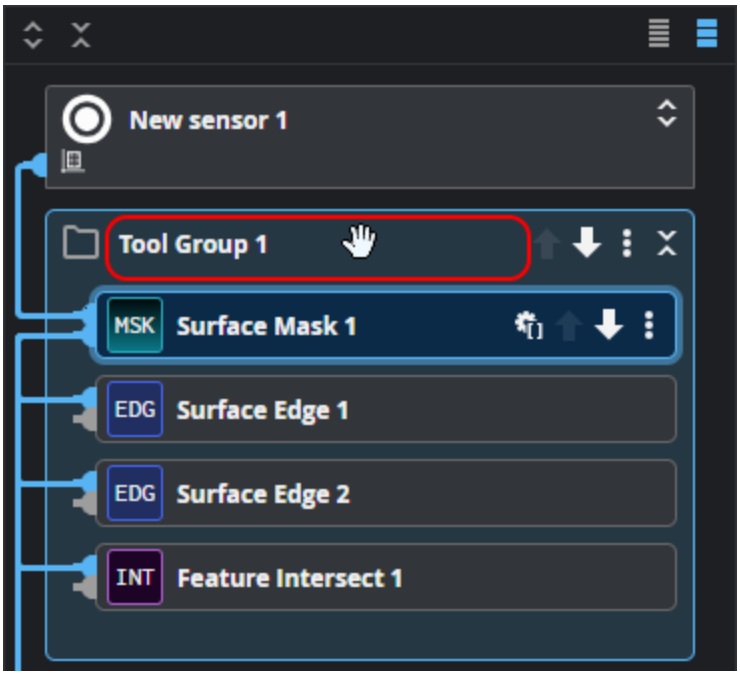
GoPxL supports various keyboard and mouse shortcuts.



Multiple tools are selected in the Tool Diagram panel (2); to the left is the Tool List (1).

Left-clicking a block in the Tool Diagram panel selects that tool or tool group. You can also select multiple tool blocks to perform operations on all of the selected tools.

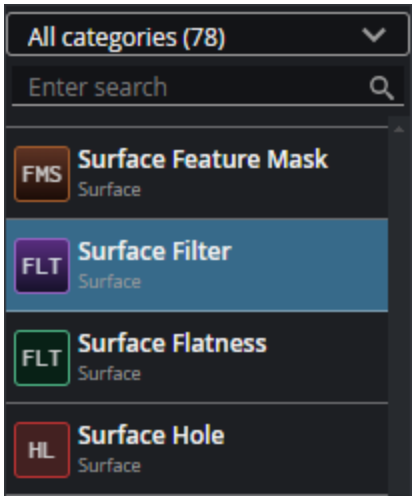
Note that whenever you select a tool group, the first tool in the group is selected automatically, and will receive any keyboard shortcuts. To use a shortcut on a tool group, click the area at the top of the tool group first.



GoPxL supports the following shortcuts for selecting tools and tool groups and performing operations on them; the term "block" is refers to tools or tool groups.

Tool block and tool group block shortcuts

Keyboard / mouse shortcut	Description
Drag and drop	Moves the selected blocks to a new position.
Shift + mouse click	Clicking a block and then pressing Shift while clicking a second block selects a range of blocks.
Ctrl + mouse click	Clicking a block while pressing Ctrl adds that block to the selection; you can select multiple blocks this way, including non-contiguously.
Ctrl + A	Selects all blocks (except for the sensor block).
Delete (keyboard)	Removes the selected blocks.
Up or Down arrow	In the Tool Diagram, the arrow keys move the selected blocks up or down in the tool list.
D key Duplicate (UI context menu)	Duplicates the currently selected blocks.
Batching (tool block icon)	Enables batching for the selected tools.

Keyboard / mouse shortcut	Description
	
Tab key	When a tool is selected in the Tool List, pressing Tab moves the focus to the next tool down. At the bottom of the list, the focus switches to the first tool in the Tool Diagram. Additional tabbing switches the focus to the next added tool. When focus reaches the bottom of the Tool Diagram, the tool selector drop-down in the **Tool Configuration** panel gets focus. 
Enter	**Tool List** Adds the tool to the Tool Diagram. Focus switches to the next tool in the Tool List. **Tool Diagram** Expands or collapses the block that has focus; tools in tool groups don't expand. **Tool Configuration** Expands or collapses the selected sub-section, such as a measurement under **Outputs**. You must click on the sub-section first. After that, toggling works.
Double-click	**Tool List**

Keyboard / mouse shortcut	Description
	Adds the selected tool to the Tool Diagram. **Tool Diagram** Expands or collapses the tool that has focus.
G key	**Tool Diagram** Group two or more selected tools into a new tool group. For more information, see *Tool Grouping* on page 185.
U key	**Tool Diagram** (when a tool group is selected) Group two or more selected tools into a new tool group. For more information, see *Tool Grouping* on page 185.
E key	**Tool Diagram** (when a tool group is selected) Edit the description of the selected tool group.
L key	**Tool Diagram** Lock the tool or the tool group. For more information, see *Tool and Group Locking* on page 190.
M key	**Tool Diagram** Merge the selected tool groups. For more information, see *Tool Grouping* on page 185.

Note that the last selected tool gets focus (its configuration is displayed in the **Tool Configuration** panel and its outputs are displayed in the data viewer). Selected tool blocks are outlined in blue; the last selected tool is highlighted with a darker background.

In the data viewer, GoPxL supports the following shortcuts:

Data Viewer shortcuts

Name	Keyboard / mouse shortcut	Description
Force pan mode	Shift + Mouse drag	When the data viewer is in orbit or zoom mode, holding Shift and dragging scan data in the data viewer with the mouse switches the data viewer to pan mode. This shortcut is only active when the mouse pointer is over the data viewer window.
Ignore regions	Alt + Mouse drag	Causes the mouse pointer to ignore regions so that you can pan or orbit over a region. This shortcut is only active when the mouse pointer is over the data viewer window.
Temporary pan	Middle mouse button	When the data viewer is in Zoom mode, holding the middle mouse button while moving the mouse temporarily switches to Pan mode.

Name	Keyboard / mouse shortcut	Description
Center camera on clicked point	Double click surface in Perspective view	Double-clicking on a point in Perspective view centers that point in the data viewer. (Click  above the data viewer to choose Perspective view.)
Z scale	Shift + mouse wheel	Modifies the Z scale value. For more information on Z scaling, see *Data Viewer* on page 79.
Toggle demo mode	Alt + Shift + D	Toggles demo mode (automatic camera orbiting) on or off when the data viewer is in Perspective mode.
Increase demo speed	Alt + Shift + "+"	Increases the demo speed.
Decrease demo speed	Alt + Shift + "-"	Decreases the demo speed.
Undo region modification	Ctrl + Z	Undoes changes to the last modified region (size or location).
Redo region modification	Ctrl + Shift + Z Ctrl + Y	Redoes a change to the last modified region (size or location).

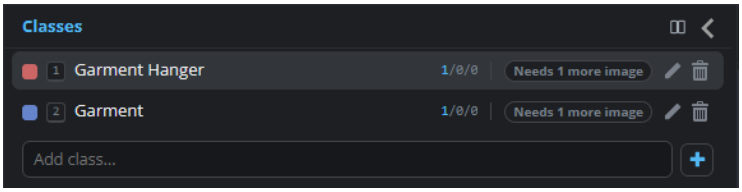
GoPxL provides the following keyboard shortcuts in the Replay Editor; for more information, see *Using the Replay Editor* on page 108.

Replay Editor shortcuts

Name	Keyboard / mouse shortcut	Description
Delete frames	Del key	Deletes the current frame, or frames that you've selected by clicking their checkbox.
Move to next / previous frame	Left arrow Right arrow	Move to the previous or next frame in the list.
Scroll	Up arrow Down arrow	Scroll up or down in the frame list.
Move to the first / last frame	Ctrl + Home Ctrl + End	Move to first or last frame.
Page up / down	Page Up / Down key	Page up or down in the list.
Space		

In the Image Object Detector tool, you can use the following keyboard shortcuts during labeling. For more information, see*Image Object Detector* on page 419 .

Image Object Detector shortcuts

Name	Keyboard / mouse shortcut	Description
Add label	Number keys	Adds a label with the corresponding class. The shortcut for a class is displayed in the **Classes** section on the **Label** tab. 
Delete frames	Del key	Deletes the currently active frame, or frames that you've selected by clicking their checkbox.
Move to previous / next frame	Q / W keys Left arrow / Right arrow	Move to the previous or next frame.

File Formats

You can download various formats in GoPxL (see below). For information on GoPxL's file name conventions, see *File Name Conventions* on page 78.

Formats

The following formats are used with GoPxL, or can be converted using the replay utility tool.

File formats

Format	GoPxL extension	Gocator firmware 6.x and earlier	Content	Description
Job	.gpjob	.job	• Scan settings • Tool settings • Control and output settings	Quickly change a set of settings by switching from one job to another. For more information, see *Jobs* on page 1179.
Recording	.gprec	.rec	• Current job settings[1] • Recorded scan data Note that sensor settings are not necessarily visible after loading a recording.	Quickly save and load recorded data for development and diagnostic purposes. For more information on recording and on loading recorded data, see *Recording and Working with Scan Data* on page 104. You can use the Replay Utils tool, provided in the utilities package, to convert .gprec files to and from various formats. For more information, see *Replay Utility Tool* on page 1294.
Backup	.gpbak	.bak	• All jobs • Current job settings[1] • Sensor system layout and	Allows restoring a sensor to a previous state in case undesired changes were made. You back up and restore sensors on the

Format	GoPxL extension	Gocator firmware 6.x and earlier	Content	Description
			transformations • Files generated by tools such as Surface Pattern Matching • Global settings	**Maintenance** page. For more information, see *Backup and Restore* on page 1184.
Support	.gpsup	.gs	• Everything the backup file contains • Additional diagnostic information	For supporting development and diagnosing issues. You download and upload support files on the **Support** page. For more information, see *Support* on page 1186.
Audit logs	.json	n/a	JSON-formatted content representing configuration changes and operations.	For performing audit logging. For more information, see *Audit Log* on page 1172.

1. See Job format details above. This also includes unsaved changes.

The following formats can be converted using the Replay Utils tool. For more information, see *Replay Utility Tool* on page 1294.

File formats

Format	GoPxL extension	Gocator firmware 6.x and earlier	Content	Description
CSV	n/a	.csv	Scan data in CSV format	First download replay data (.gprec) in GoPxL, and then use the Replay Utils tool to convert to the CSV format, for use with third-party tools.
.pcd / .sur / .srf / .pro	n/a	n/a	Scan data in various formats	First download replay data in GoPxL, and then use the Replay Utils tool to convert to the desired format, for use with third-party

Format	GoPxL extension	Gocator firmware 6.x and earlier	Content	Description
				tools.

You can also use the Data Export tool to export scan data in various formats. For more information, see *Data Export Tool* on page 247.

File Name Conventions

When you export or download files, GoPxL uses the following conventions:

Exported date/time: `YYYYMMDD-HHMMSS`

Platform: `Sensor`, `PC` (if accelerated by a PC instance of GoPxL), or `GoMax` (if accelerated by GoMax)

Model and serial number: `G{model_number}_{serial_number}`

GoPxL version: `{major_minor_patch_build}`

For example:

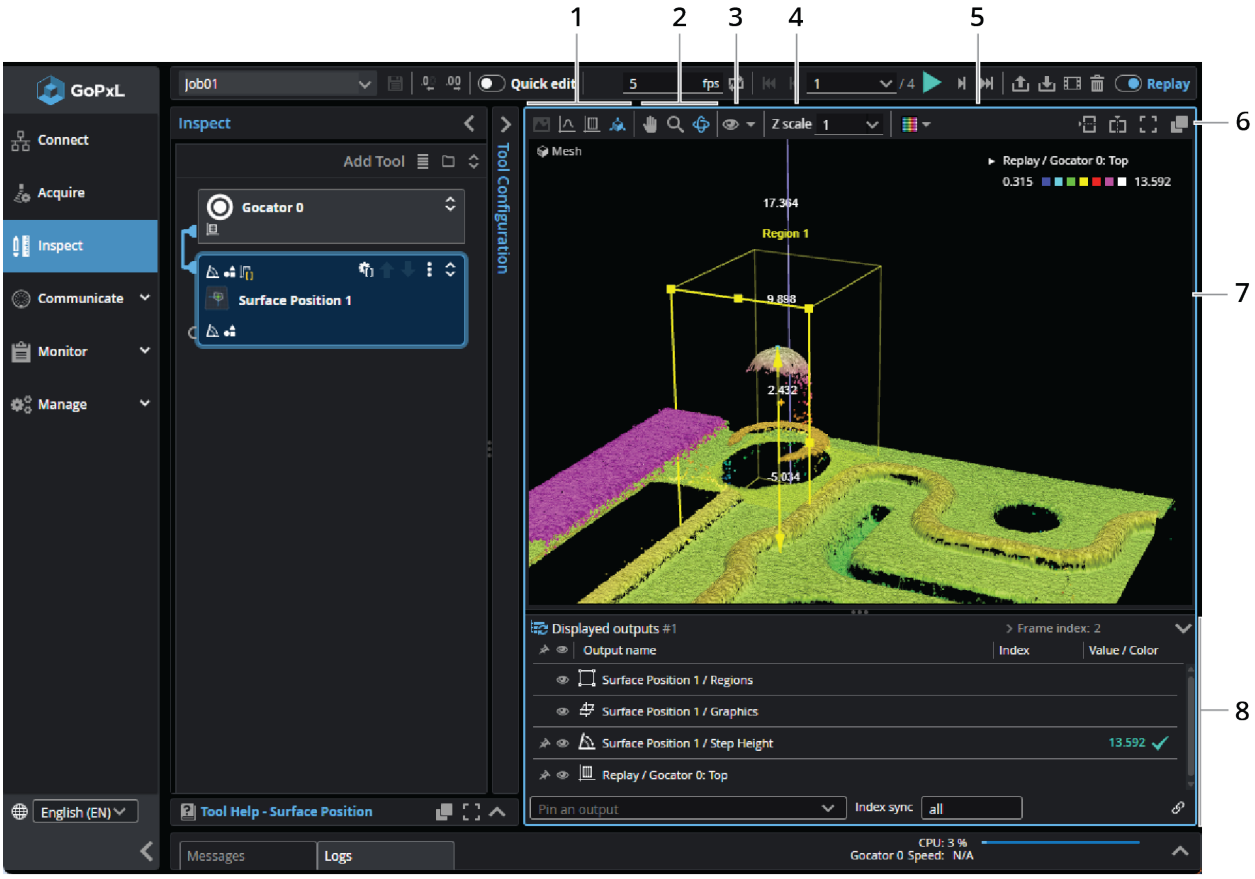
`20261020_134550-Sensor-G2520_65656-1_6_33_75.gprec`

Job files names (.gpjob) only use the name you provide in GoPxL, for example:

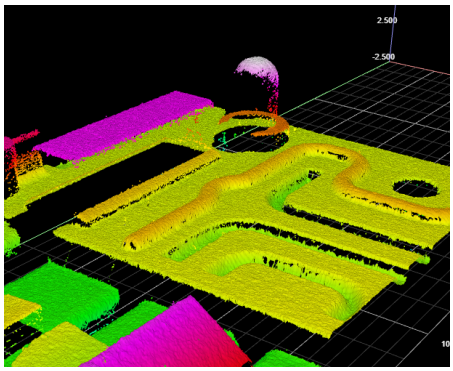
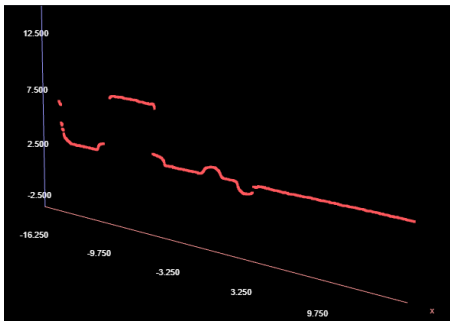
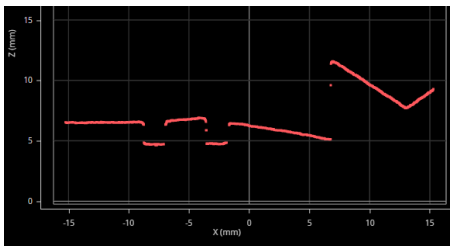
`morning_production_run.gpjob`

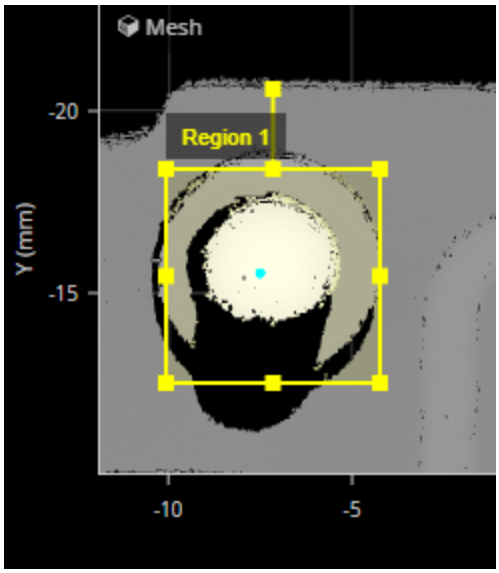
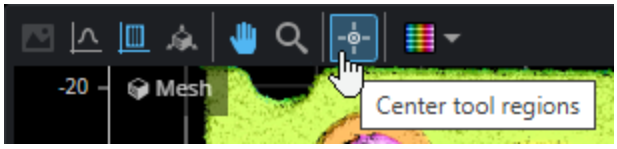
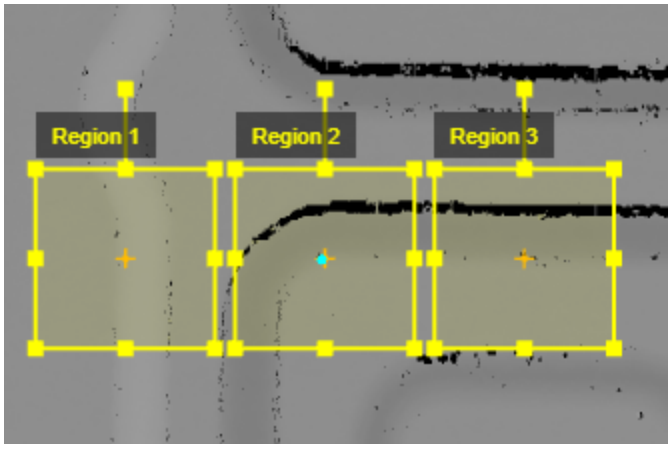
Data Viewer

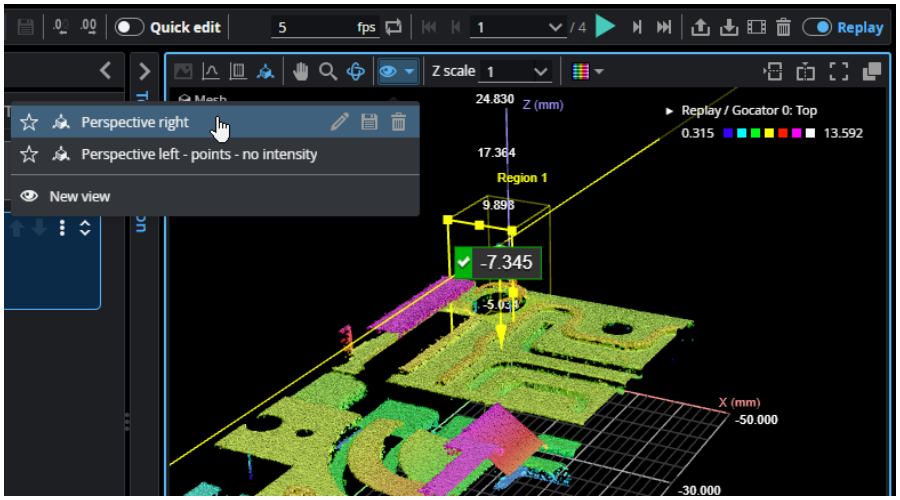
You use the data viewer to observe scan data (such as Profile and Surface data), as well as video images and intensity images. You can also use it to configure measurement tools (see *Inspect - Measurement and Processing Tools* on page 162) and a sensor's active area (see *Active Area* on page 154). GoPxL supports multiple data viewers (via both splitting and pop-out buttons). You can save "views" that preserve the data viewer's mode (perspective or other) as well as the orientation of the scan data in a perspective view.

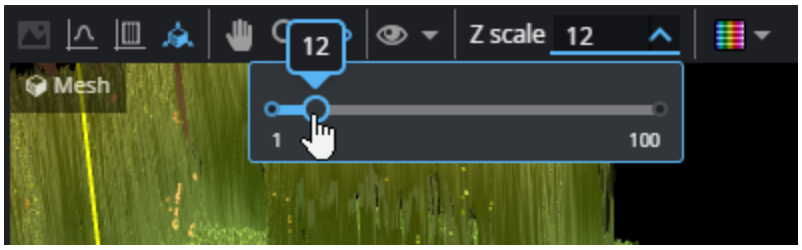
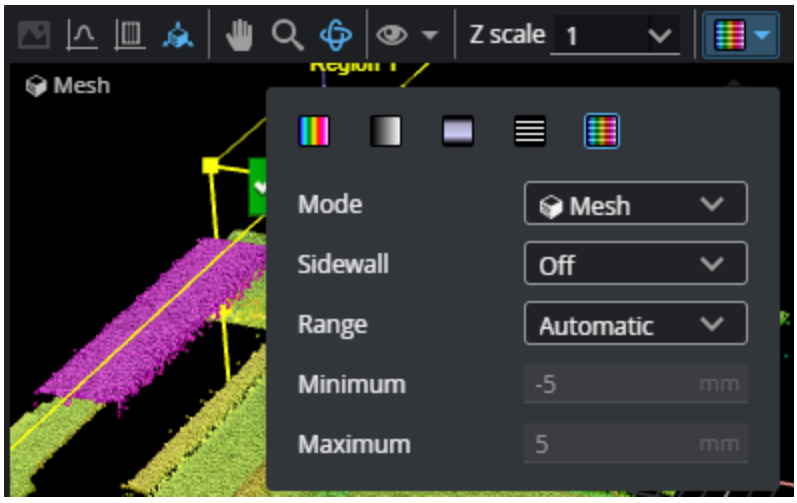


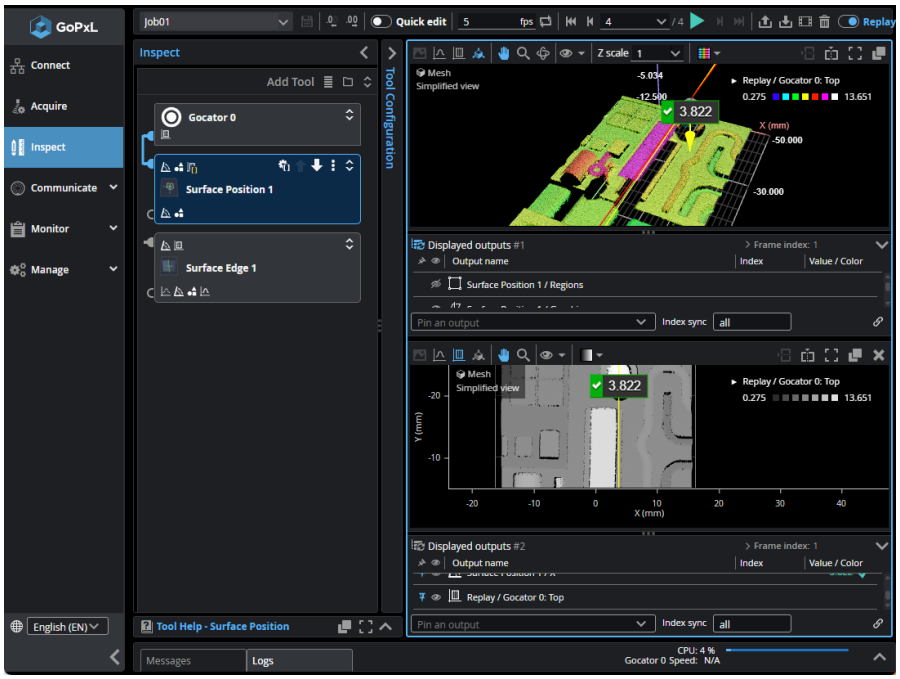
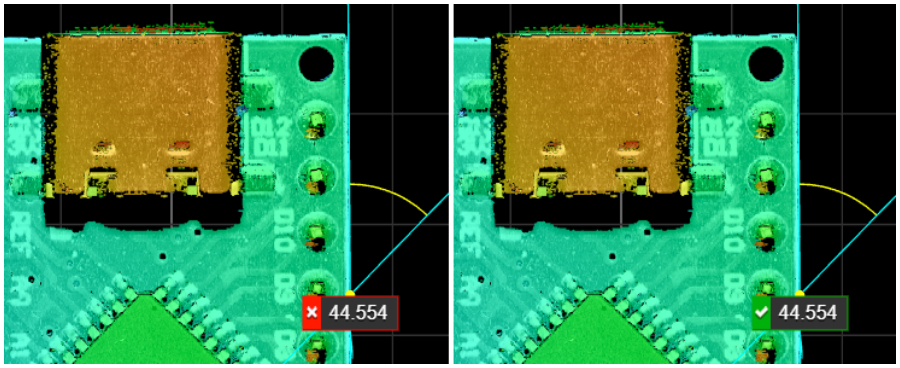
	Element	Description
1	Display modes	Use these to change the data viewer's display mode. One of the following: **Image** Image mode ( ) lets you view pixel data that the system has acquired in Image mode. You should use Image mode to adjust exposure as an initial step and for diagnostics. For more information, see *Image Mode* on page 85.

Element	Description
	Perspective Perspective mode ( ) lets you view data in a 3D perspective representation. You can move and turn the data using a mouse and the pan, orbit, and zoom buttons ( ). When in orbit mode, hold Shift to temporarily switch to pan mode. The data viewer can display both Surface and Profile data in Perspective display mode. For information on controlling how the data viewer displays Surface data in Perspective mode, see *Surface and Perspective Display Options* on page 89.   **Profile / Front** Profile / Front mode ( ) displays profiles in a 2D representation when Profile data is available.  **Surface / Top** Surface / Top mode ( ) lets you display surface data in a 2D representation when Surface data is available. For information on

	Element	Description
		controlling how the data viewer displays Surface data in Perspective mode, see *Surface and Perspective Display Options* on page 89.  Note that In Surface / Top display mode (and in Profile / Front mode), a "Center tool regions" button is available at the top of the data viewer when a tool is selected in the Tool Diagram panel.  Clicking this centers all regions in the middle of the data viewer, resizing them to fit. This can be useful when you have zoomed the data viewer in a lot, and need to quickly bring the regions into view.  For more information on regions, see *Regions* on page 214.
2	Pan, zoom, and	Use these buttons ( ) to move scan data in the data viewer with your mouse. (Orbit is only available in the perspective view mode; see

	Element	Description
	orbit	above.) Various shortcuts are available to make working with the data viewer easier. For more information, see *Keyboard and Mouse Shortcuts* on page 70.
3	View drop-down	GoPxL lets you define "preset views" for the data viewer, which let you display a specific 3D viewing angle and data viewer setup to make configuring measurement tools easier. This can also be useful for demos. Click the eye icon ( ) to create or choose a preset view for the data viewer. In the preset view drop-down, you can also rename the view the pointer is hovering over, save the current state to it, or delete it.  Views contain the following: • zoom level and orientation of the scan data in the data viewer • the data viewer mode (perspective, top, and so on) • other settings, such as mesh versus points display Currently, Range can't be included in a view. You can select a perspective view to be the default view. The default view's "star" icon is filled. Views are not part of a job file; they are available at all times once added in GoPxL. You can get a link for the current view by clicking the link icon at the bottom right corner of the Displayed Outputs panel; you can then use this link in an HMI application. For more information, see *GoHMI and GoHMI Designer* on page 1210.
4	Z scale	When the data viewer is displaying Surface data (in perspective view only) or Profile data, you can adjust the Z scaling to make it easier to distinguish minor height variations in the scan data.

	Element	Description
		
5	Surface display options	Lets you control how scan data is displayed, such as whether Surface scan data is displayed using a color heightmap or a grayscale heightmap overlay. For information on changing how the data viewer displays Surface data, see *Surface and Perspective Display Options* on page 89. For information on how the data viewer displays Profile data, see *Profile Mode* on page 87. 
6	Split, zoom, and pop-out	Use the split buttons (and) to create additional data viewers. You can create as many as four data viewers in the main browser window. To close a split window, click its Close icon ( ).

	Element	Description
		 Use the Pop-out button ( ) to create a new window with a single data viewer. For more information, see *Using Multiple Data Viewers* on page 100. Use the Zoom to fit button ( ) to zoom the scan data so that it fits in the data viewer window.
7	Data viewer window	This is the main window in the GoPxL interface that shows scan data. When you configure tools or set the active area, you can graphically configure the regions in the data viewer window. The values of pinned or currently selected tool measurements are displayed in the data viewer. For more information, see *Measurement Values and Labels* on page 194.  When the view is zoomed out, to reduce processing, the number of data points that are displayed is reduced, as indicated by "simplified view" in the data viewer. When you zoom in enough, the actual data points are

	Element	Description
		all displayed. The legend at the upper right is collapsible. 
8	Displayed Outputs pane	Lets you pin and show / hide outputs. When you pin a measurement output, how the value is displayed in the data viewer may change. For more information, see *Displaying and Pinning Outputs* on page 94. When working with arrays, you can also select an array element (or all of the elements) across all arrayed tool outputs. Use the link icon (🔗) in the lower right of the Displayed Outputs pane to copy the path to the data viewer for use in GoHMI Designer to display a data viewer in your HMI. For more information, see *Using a Data Viewer in an HMI* on page 1228.

Image Mode

In Image mode, the data viewer displays images directly from a sensor's camera or cameras, as well as other information.

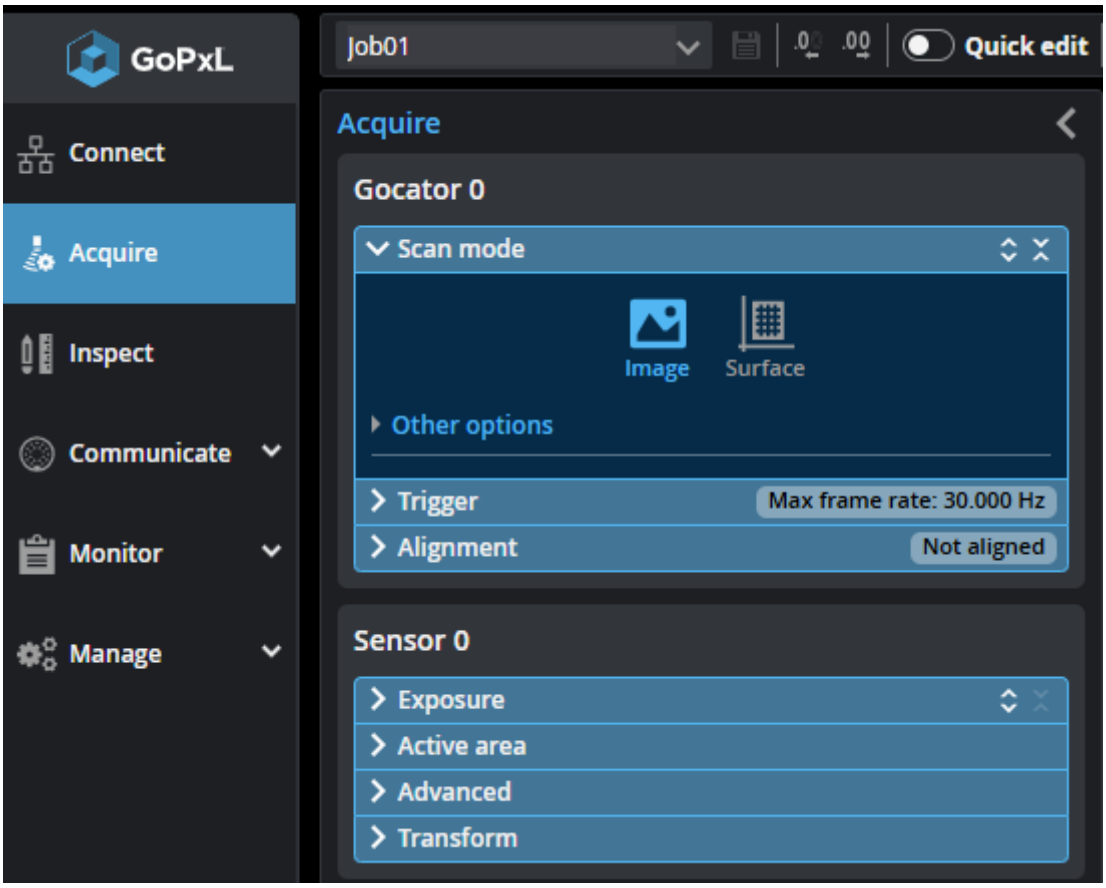
In this mode, depending on your sensor model, you can use the data viewer to display exposure information (see *Exposure Information* on the next page) that can be useful in properly setting system exposure for scanning, and for troubleshooting stray light or ambient light problems.

Gocator sensors can't 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 with minimal red or blue areas appearing in Image mode, you can try enabling the **Multiple** exposure feature; the sensor will then combine the exposures to get the best scan data possible. Choose the exposure you want to examine in the Displayed Outputs panel 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.

> Not all sensor models indicate over-exposed areas (red).

The correct tuning of exposure depends on the reflective properties of the target material and on the requirements of the application. You should carefully evaluate the exposure settings for each application.

To view Image data, you must set the system to Image mode before scanning, in the **Acquire** > **Scan mode** section.



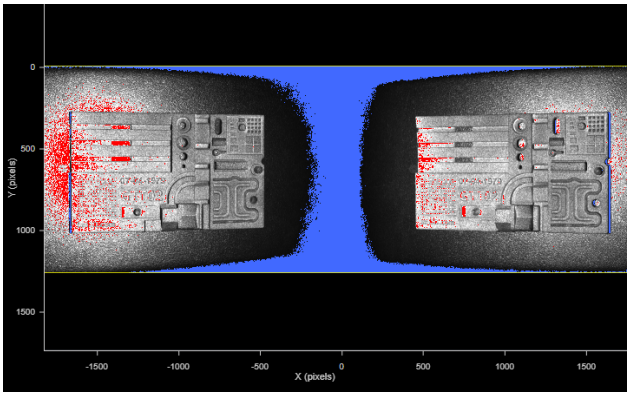


Image data showing left and right camera views.

Exposure Information

In Image mode, you can display exposure-related information. This information can help you correctly adjust the exposure. (For information on setting exposure, see *Exposure* on page 148.)

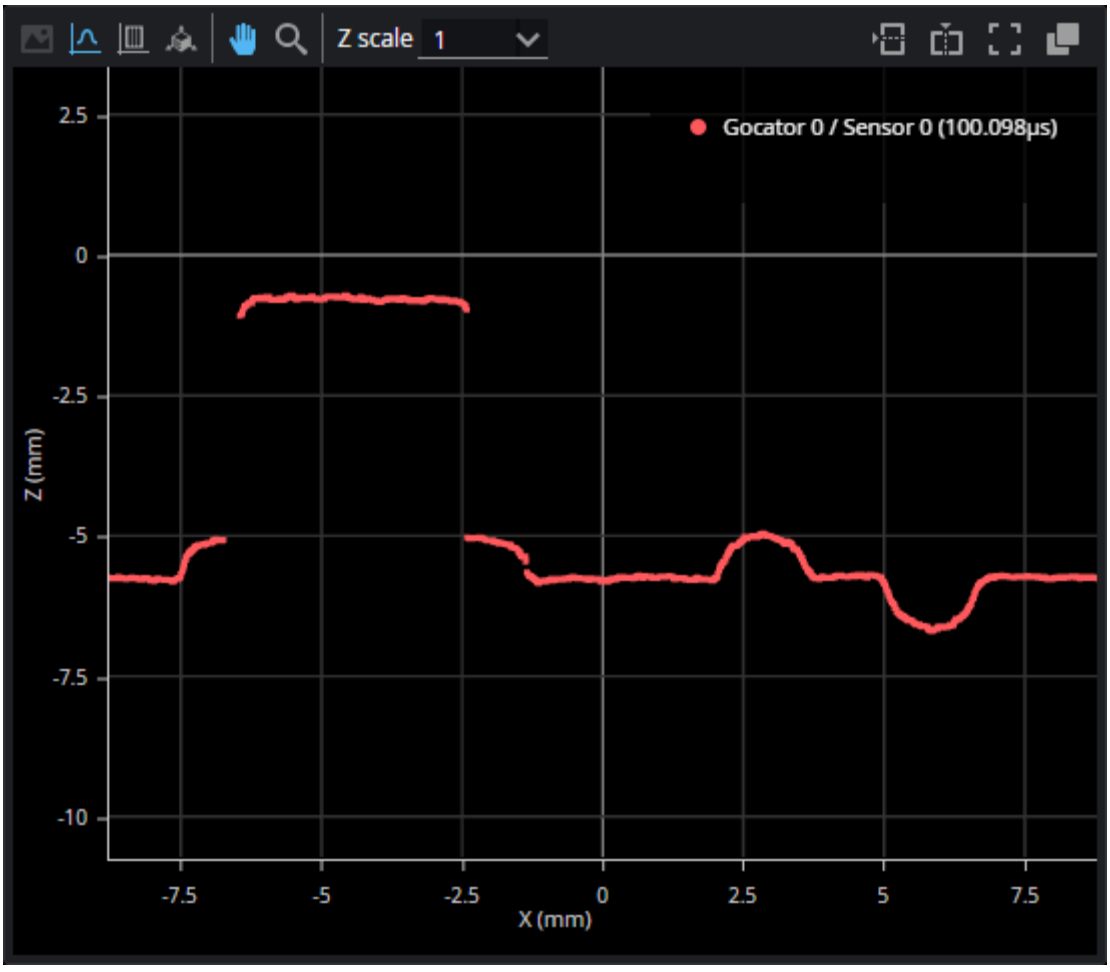
To display exposure information, click the Exposure button ( ) at the upper left of the data viewer.

Exposure information is listed in the **Displayed Outputs** panel at the bottom of the data viewer. If you have set the sensor's exposure to **Multiple** on the **Acquire** page, and have set more than one exposure, each exposure is listed individually (G2) or as an element available in a drop-down (other sensors). Exposure information is displayed in Image ( ) outputs.

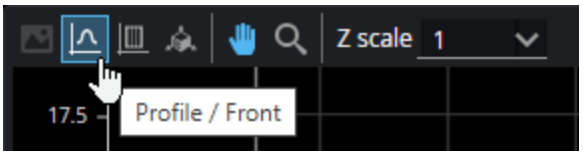
To display this information, the output must be set to visible. That is, the "eye" icon must be enabled ( ) instead of disabled ( ). With multiple exposures, only the "top" one is displayed in the data viewer. For example, to see the exposure information for Exposure 3, below, you must hide the Image outputs of Exposure 1 and 2.

Profile Mode

When Profile data is available, either because GoPxL is in Profile scan mode or because a Profile output of a tool like Surface Section is providing Profile data, the data viewer can displays profile plots.

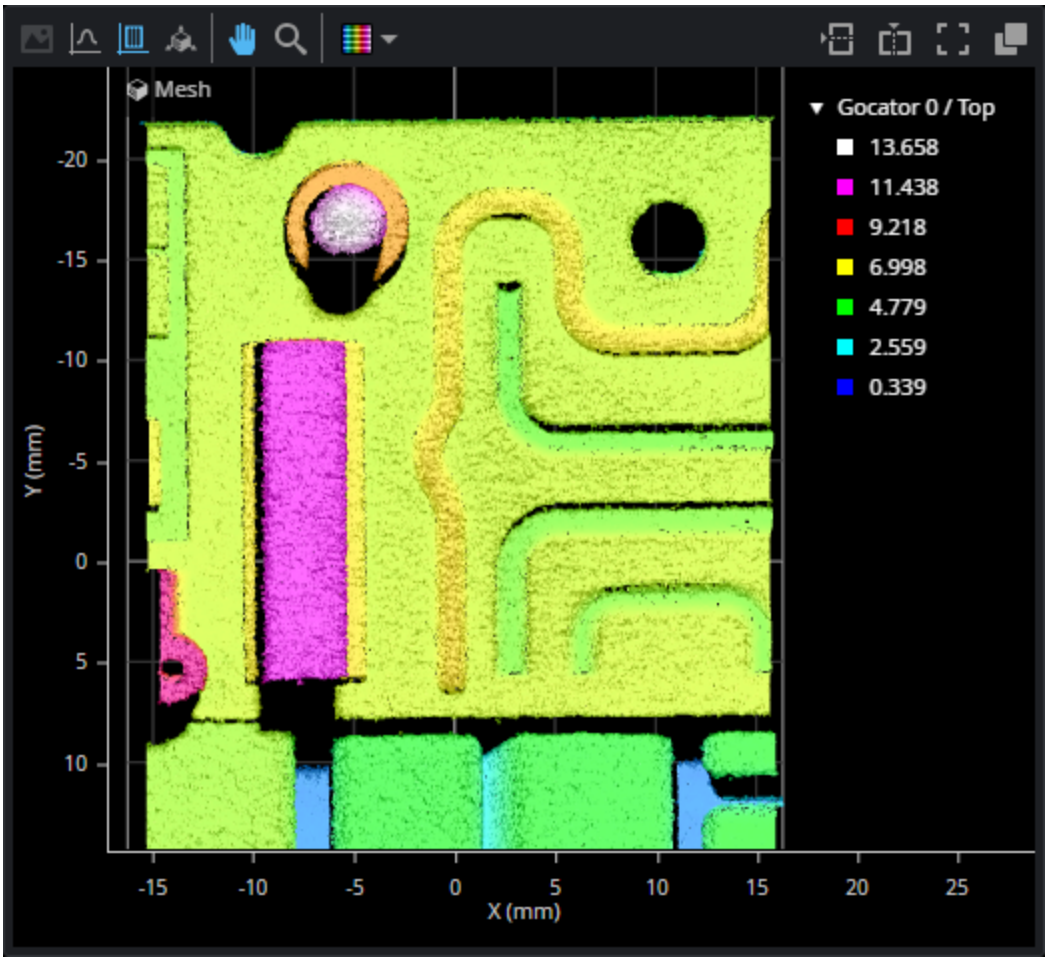


To set the data viewer to Profile mode, click the Profile mode button above the data viewer window:

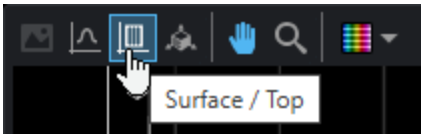


Surface Mode

When the data viewer is in Surface / Top mode and Surface data is available, the data viewer can display Surface data.



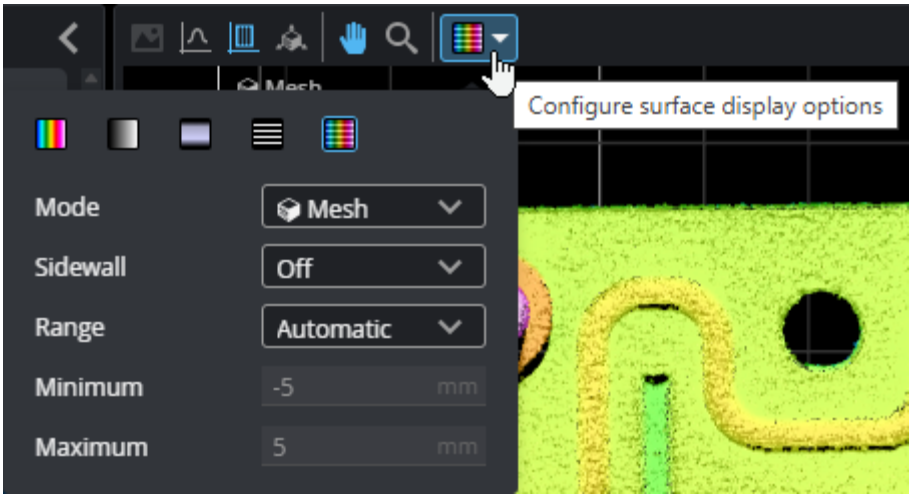
To set the data viewer to Surface / Top mode, click the mode button above the data viewer window:



GoPxL provides several ways of controlling how Surface data is displayed. For more information, see *Surface and Perspective Display Options* on the next page.

Surface and Perspective Display Options

You control how Surface data displays when the data viewer is in Surface or Perspective mode using the display options available in the Surface display options drop-down at the top-left of the data viewer.

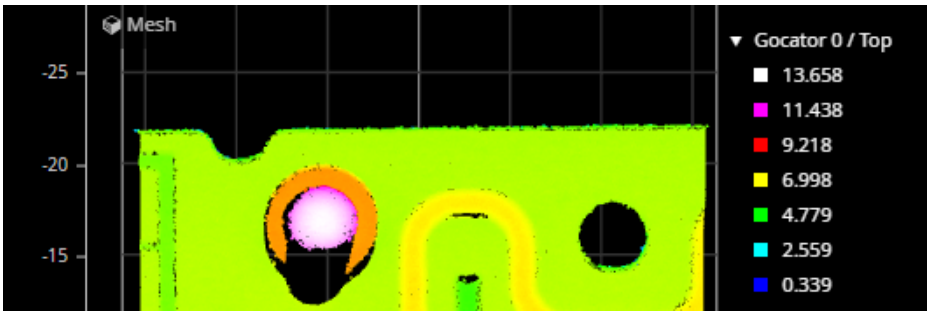


Surface display options.

Display options

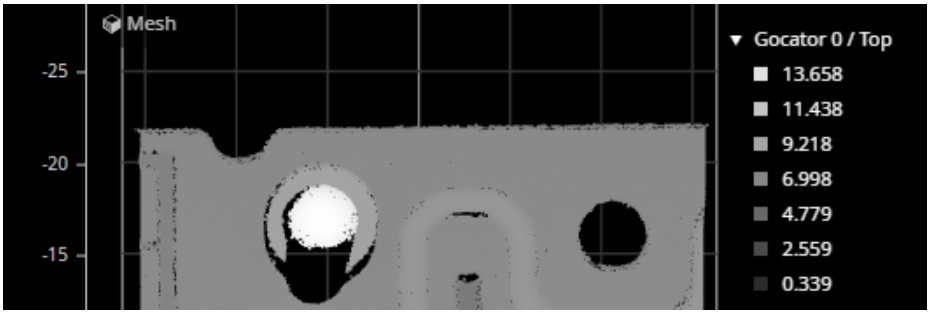
Option	Description
Display options	The output display modes let you choose how GoPxL shades the Surface data when the data viewer is in Surface mode or in Perspective mode.  For Heightmap, Grayscale, and Heightmap + Intensity settings, you can set the range values that map to the heightmap colors (using the **Range** drop-down and the related settings) to make features easier to see. The following output display modes are available: **Heightmap** Heightmap mode () displays a pseudo-color heightmap over the Surface data. A heightmap legend is displayed in the data viewer.

Option	Description



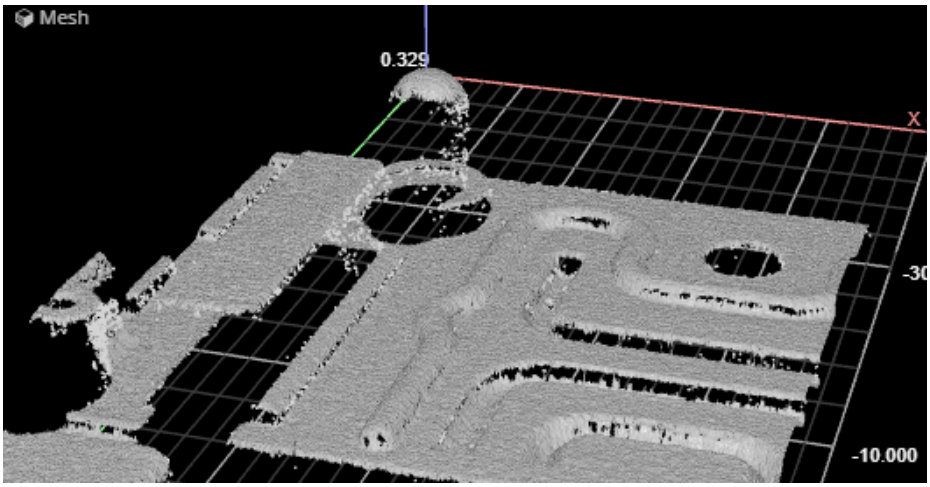
Grayscale

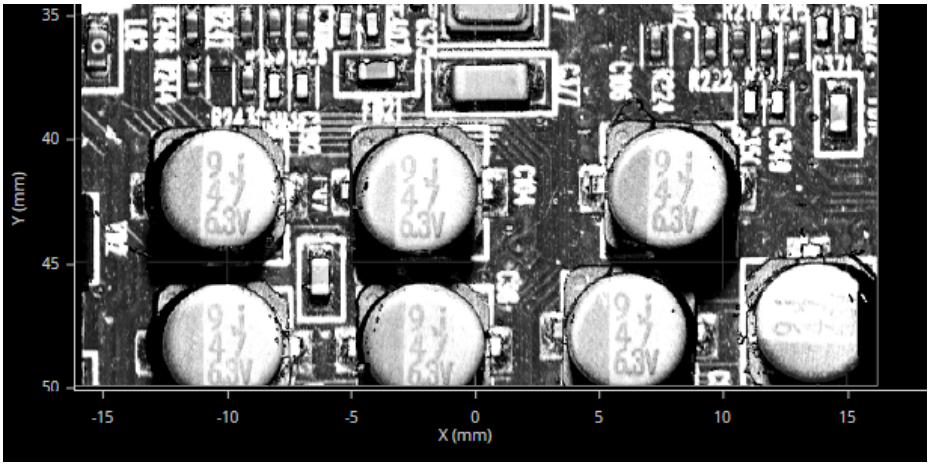
Grayscale mode ( ) displays a grayscale heightmap over the Surface data. Working in grayscale mode can make it easier to distinguish the scan data from tool visualizations. A heightmap legend is displayed in the data viewer.

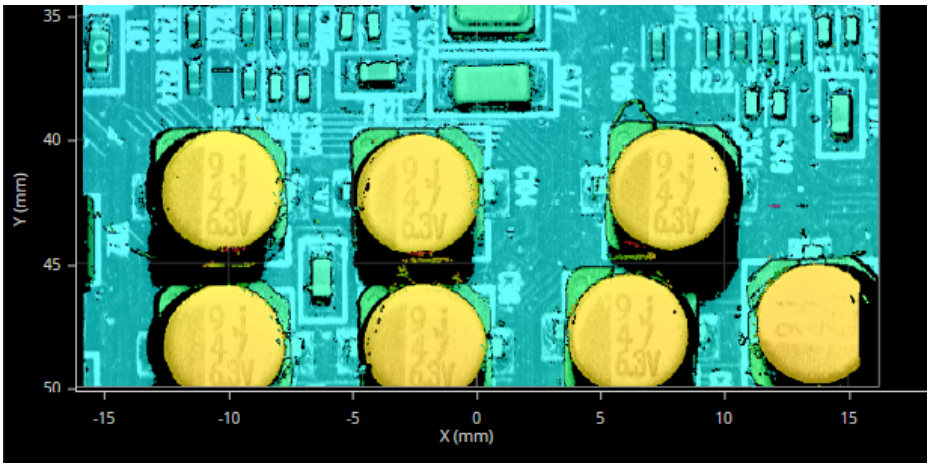
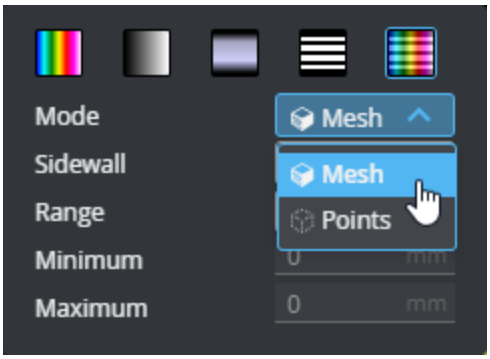
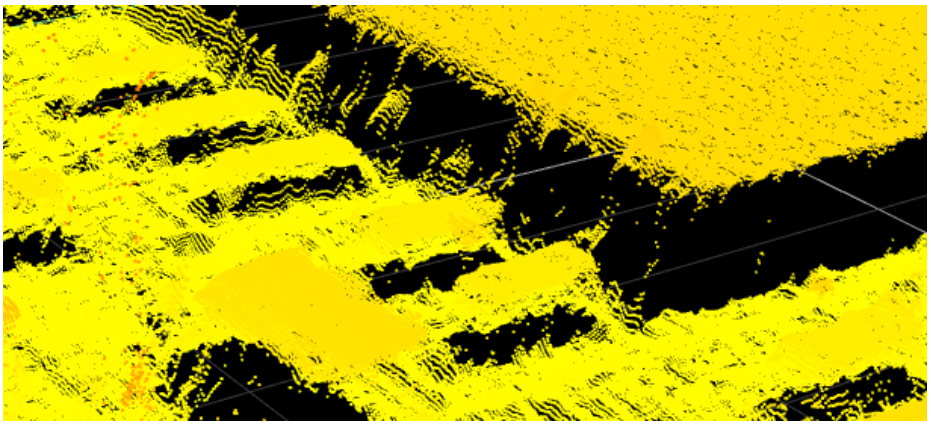


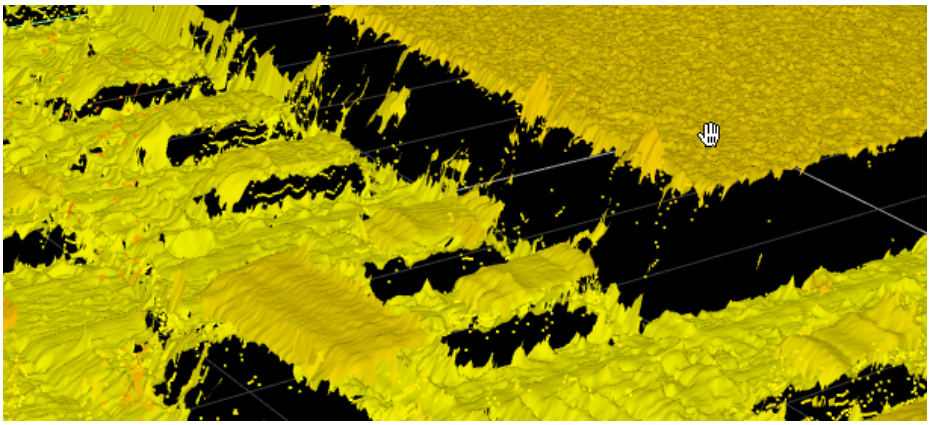
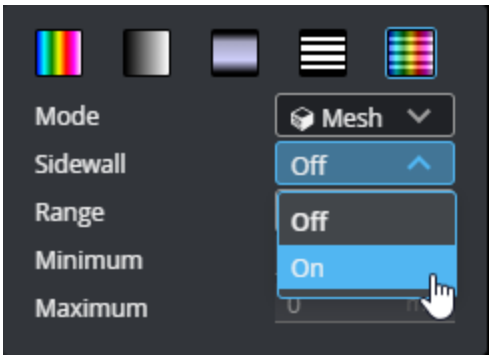
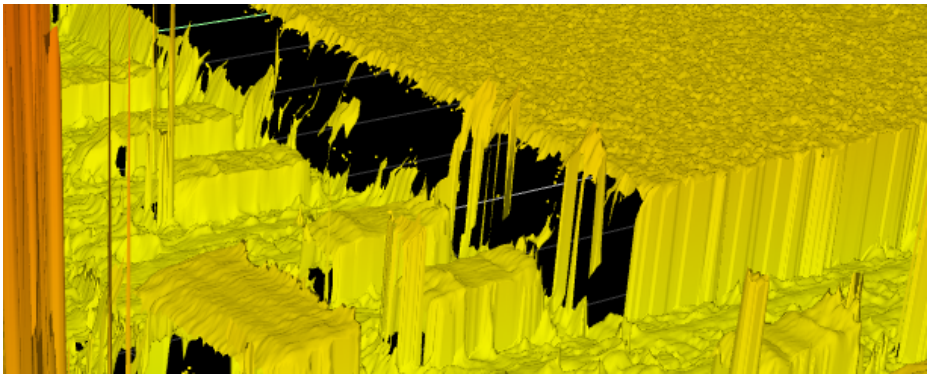
Uniform

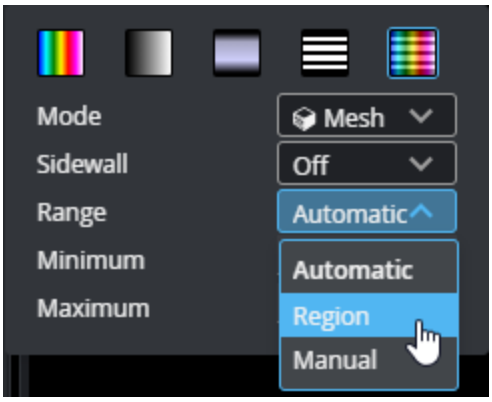
Uniform mode ( ) displays a uniform gray color over the Surface data. This mode is mostly useful when you want to focus on shape or geometry.



Option	Description
	You normally need to set **Mode** to Mesh in the Surface display options drop-down to use this display option. **Intensity** Intensity mode ( ) displays intensity data over the Surface data if it is available. When intensity data is enabled in the **Acquire** page > **Scan mode** section, sensors produce intensity images that measure the amount of light reflected by an object. An 8-bit intensity value is output for each data point in the scan data, where a darker shade represents a smaller value and lighter shade represents a larger value. The sensor applies the same coordinate system and resampling logic as the ranges to the intensity values. For more information, see *Scan Modes and Intensity* on page 139.  **Heightmap + Intensity** Heightmap + Intensity mode ( ) displays a color heightmap combined with intensity data on the Surface data. You must enable intensity data before scanning for this mode to be available. For more information, see *Scan Modes and Intensity* on page 139.

Option	Description
	
Mesh and Points modes	You can choose whether the data viewer displays Surface data as a mesh or points. Note that the data viewer Mesh mode is unrelated to the Mesh data type. (For information on the Mesh data type, see *Mesh Tools* on page 493.)  Points mode displays the data as discrete data points. This mode is useful in scan data that contains noise around edges, and can show hidden structure.  Mesh mode displays Surface data by connecting the data points with polygons.

Option	Description
	
Sidewall mode	You can toggle the data viewer between hiding and showing polygons which are vertical or nearly vertically oriented. This can produce a more realistic-looking part for demonstrations.  For example, in the following, the sidewalls are enabled, resulting in the lines shown at the edges of the PCB components.  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.
Heightmap range scaling	The data viewer displays heightmap information in pseudo-color or grayscale. The height axis (Z) is color-coded in the legend to the right in the data viewer. You can adjust the scaling of the height map using **Range** and its related settings.

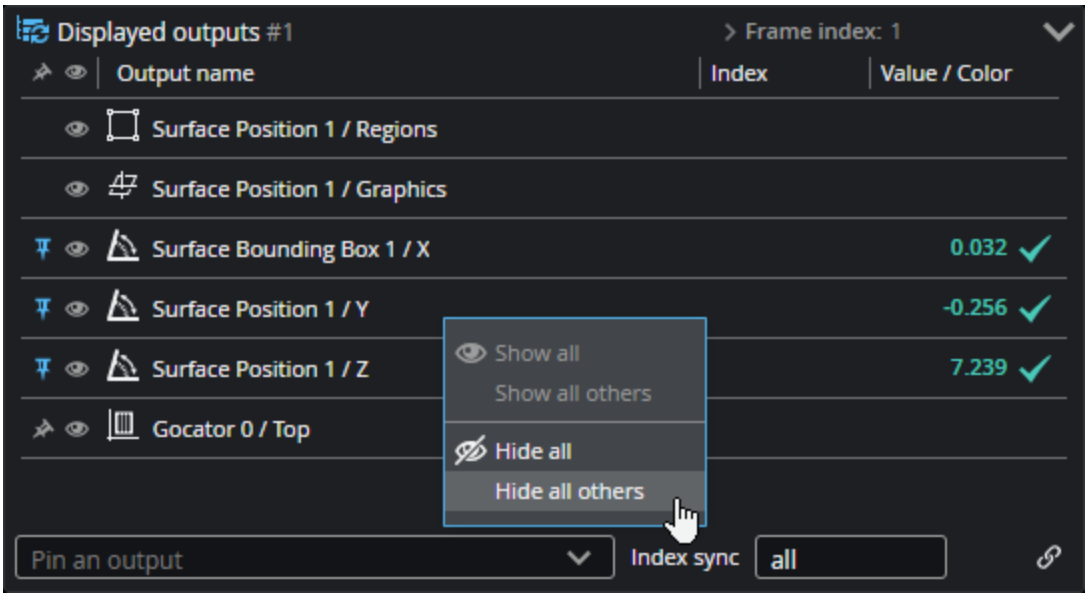
Option	Description
	 Do one of the following: • To automatically set the scale, choose **Automatic** in the **Range** drop-down. • To automatically set the scale based on sub-region you select on the heightmap, choose **Region** in the **Range** drop-down and adjust the yellow region box in the data 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.

Displaying and Pinning Outputs

You can interact with the inputs, outputs, and other elements listed in the **Displayed outputs** panel to control what you see in the data viewer window, by pinning or hiding outputs. This makes it easier to configure and monitor your application.

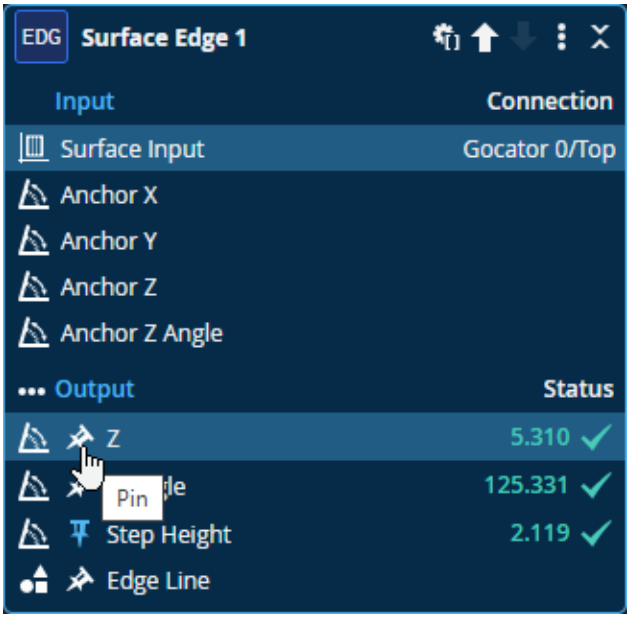
Pinning outputs affects how measurement values are displayed in the data viewer. For more information, see *Measurement Values and Labels* on page 194.

When you select a block (a tool or the sensor, device, or source at the top) in the Tool Diagram panel, the **Displayed outputs** panel lists items related to the currently selected tool. By default, the panel lists *all* of the selected tool's inputs, the currently selected output of that tool, any pinned outputs, and tool-related graphics. In addition to selectively pinning or hiding outputs, you can right-click on an output in the panel to access "show / hide all" options.

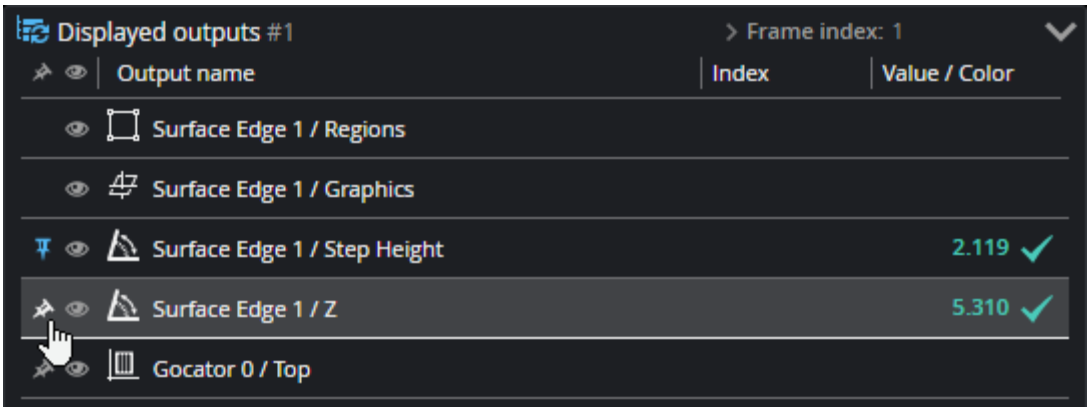


You can “pin” items in the **Displayed outputs** panel so that it they are always displayed in the data viewer, even if you select a different tool, or another input or output in a tool block. Pinning items is useful for example if you want to monitor multiple, independent measurements while GoPxL is 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 change has on another tool (whose output is pinned) later in the chain. This minimizes toggling and clicking between tools and measurements.

You pin items in the **Displayed outputs** panel, in the tool's block in the Tool Diagram, or in the tool's Outputs section in the **Tool Configuration** panel. Pinned outputs are identified in GoPxL by a vertical blue "thumbtack" icon. Unpinned outputs are identified by a grey thumbtack icon. (See below.)

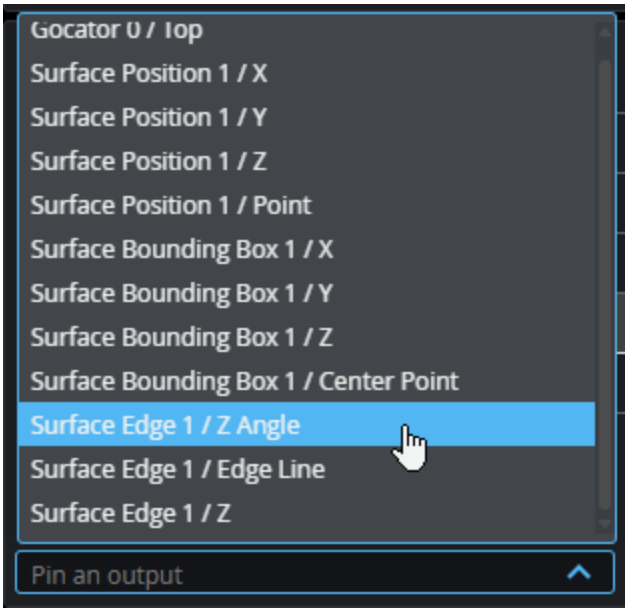


Pinning an output in the Tool Diagram.



Pinning an output in the Displayed Outputs panel.

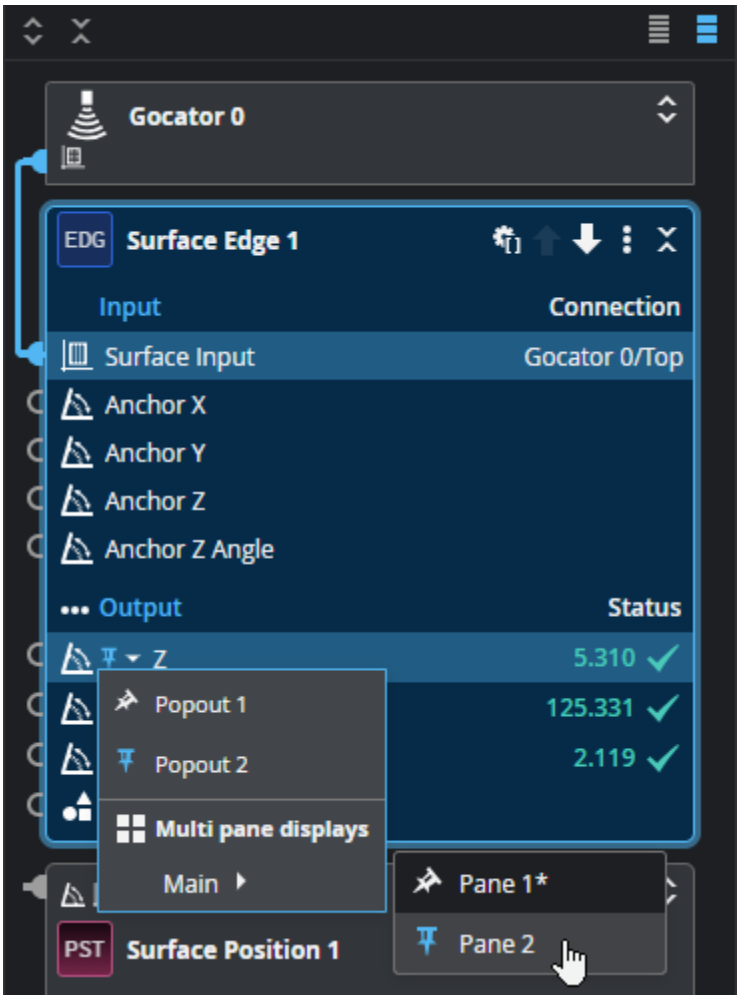
You can also choose an item from the "Pin an output" menu at the bottom of the panel.



If you have created additional data viewers, either by splitting a data viewer in the main browser window, or by popping out a data viewer in a new window, you can choose which data viewer to pin an output to in the Tool Diagram panel. (For information on creating and using new data viewers, see *Using Multiple Data Viewers* on page 100.)

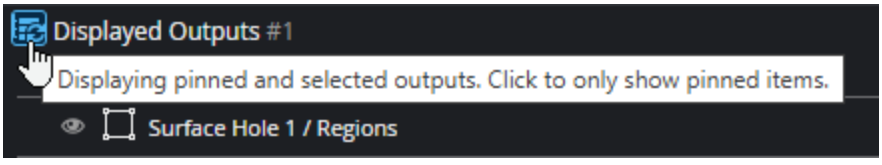
"Popout" displays are data viewers you've created using the "Pop out" button ( ), which display in a new browser window.

"Multi-pane displays" are data viewers you've created using one of the "split" buttons (or). In a multi-pane data viewer, the pane identified by an asterisk is the original data viewer, the one from which you created others.



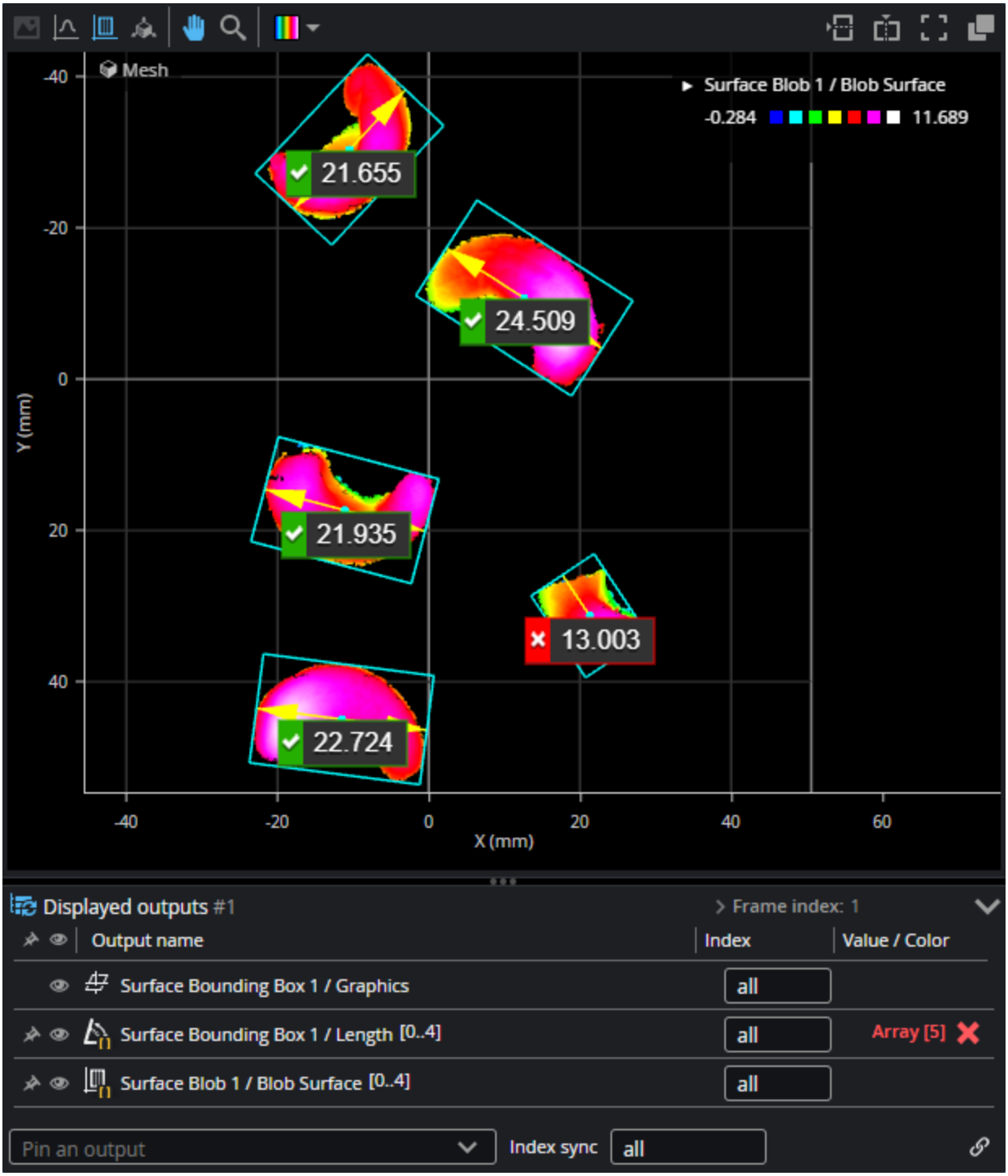
In both cases, the input of the currently selected tool and the currently selected tool output are added to the new data viewer and are pinned. You can add more outputs from the "Pin an output" menu.

You can control whether a data viewer displays both pinned and selected inputs / outputs, or instead *only* pinned outputs. Displaying only pinned outputs lets you focus on those inputs and outputs in the data viewer while configuring or monitoring your application. You toggle between these modes by clicking the Synchronize pane button ( ) at the top of the **Displayed Outputs** panel.

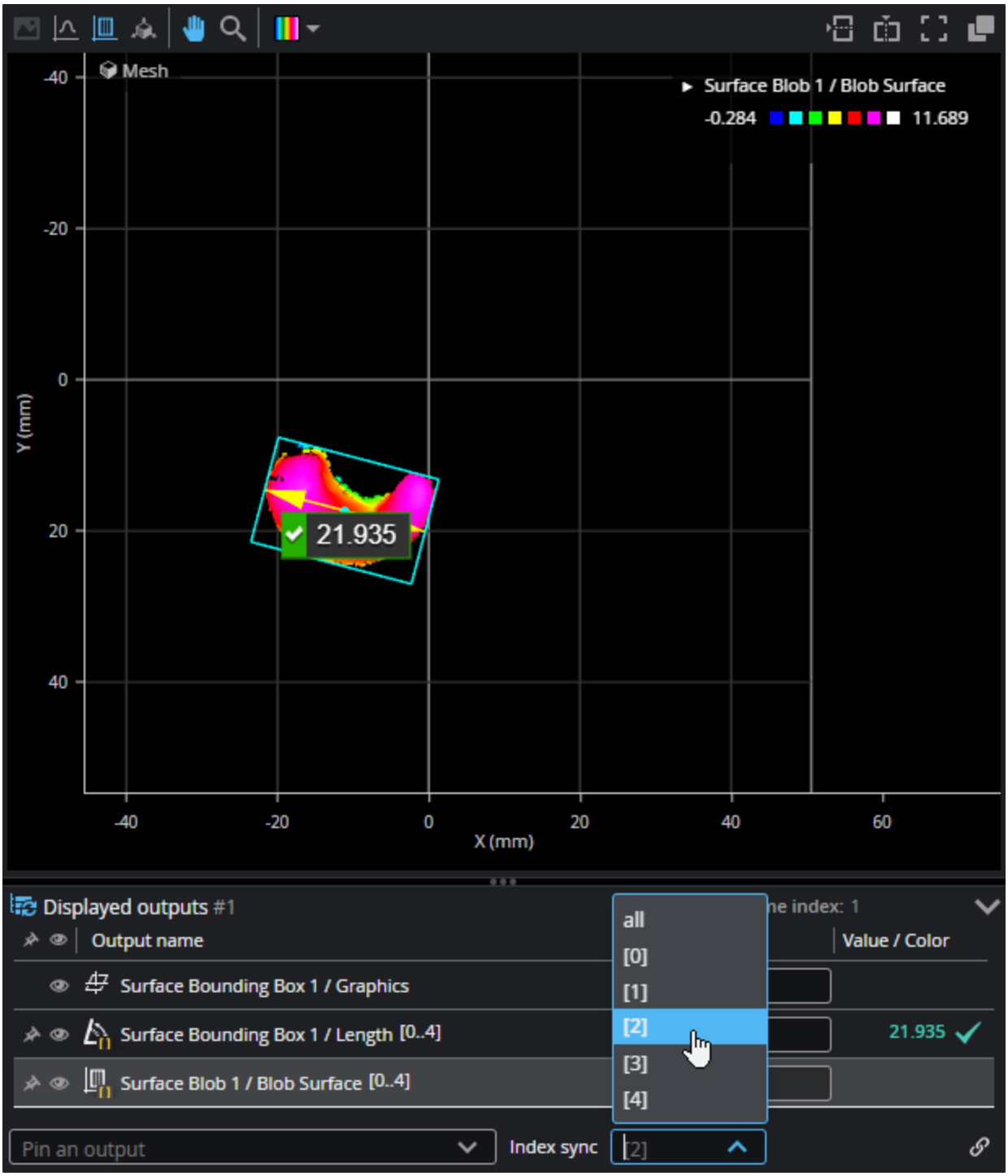


By default, this is enabled in the main data viewer, and disabled in any pop-out or split data viewers you create. Note that if there are no pinned outputs in any of the tools in your job, and you disable pane synchronization, the **Displayed Outputs** panel will be empty and nothing will be displayed in the data viewer.

When you are working with arrayed outputs, you can select an array element for display across all arrayed outputs listed in the **Displayed Outputs** panel. For example, in the following, we see all five objects in an arrayed output displayed with Bounding Box tool graphics and Length measurements for *each* element, as well as the Surface Blob output.



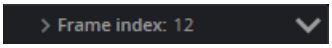
If you want to focus on a single element, you can choose or type that element's index in the **Index sync** drop-down.



Note that the default is to display all elements.

Frame Information

The area at the top right of the **Displayed outputs** panel provides useful frame information that you can cycle through by clicking the chevron symbol.



The following information is available:

Frame index: Displays the index in the data buffer of the current frame. The value resets to 1 when the sensor is restarted or when recording is enabled.

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

Timetick is a 64-bit counter that increases with time, so each scan from the same device has a unique timetick value.

On Gocator 3D sensors, to convert this value to microseconds, use the following:

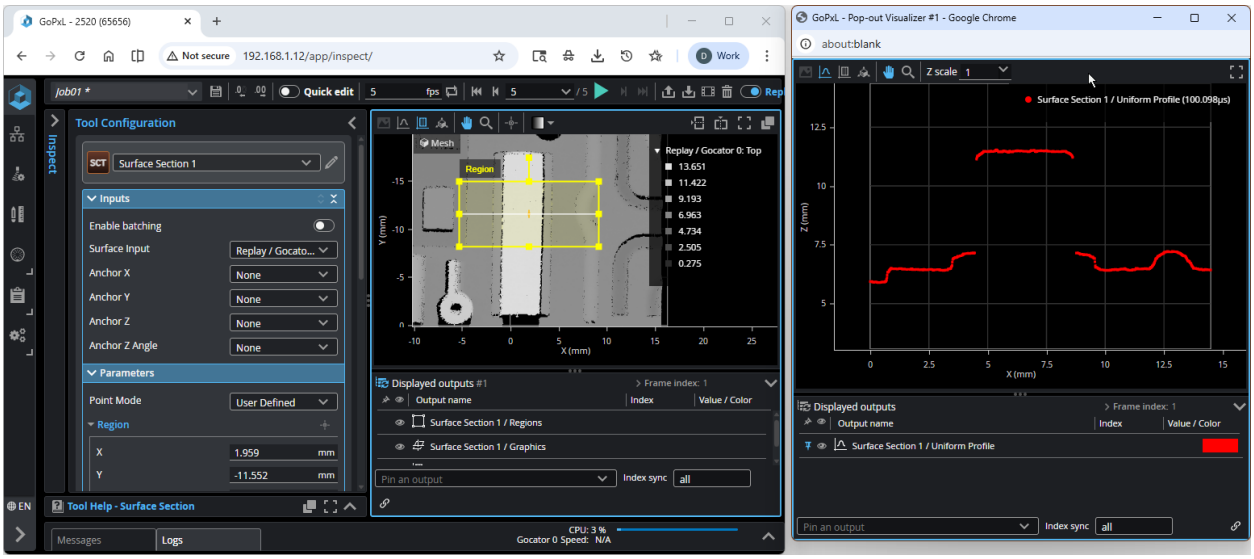
```
time (µs) = timetick / 1.024.
```

Encoder index: Displays the encoder value at the time of the frame's capture.

Encoder at Z: Displays the encoder value when the encoder Z signal is triggered. This is especially useful in rotational applications.

Using Multiple Data Viewers

You can more easily monitor or set up complex applications by popping out multiple data viewers outside of the main browser window, or by splitting a data viewer into two data viewers in the same browser window. You can configure each data viewer to different views, different display modes, and different sets of pinned outputs. This gives you complete flexibility.



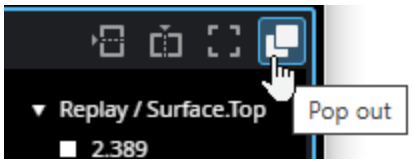
Main view in original browser window showing surface data and a defined section, and a second window showing a Profile tool running on the section.

External data viewer windows provide the same functions as the main 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 *Displaying and Pinning Outputs* on page 94.

To create a new data viewer window

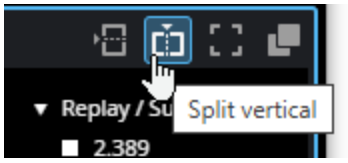
Do one of the following:

- In the toolbar of the main view data viewer, click the Pop out button ( ).



A new browser window opens containing a separate data viewer.

- In the toolbar of any data viewer, click one of the "split" buttons (or).



A new data viewer appears in the same browser window.

Use the tool bar at the top of the new data viewer to choose and modify the view. For more information, see *Data Viewer* on page 79.

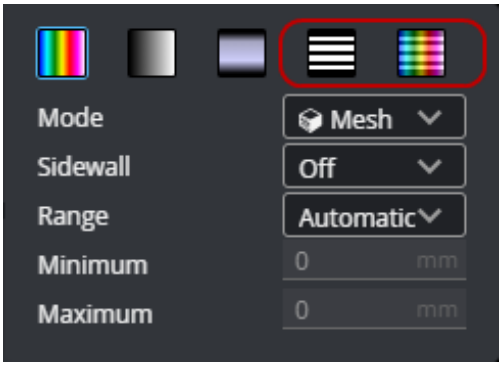
Pin outputs to the new data viewer as in the main view data viewer. For more information, see Displaying Multiple Measurements and Geometric Features (Pinning). Any outputs pinned in the

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.

Intensity Output

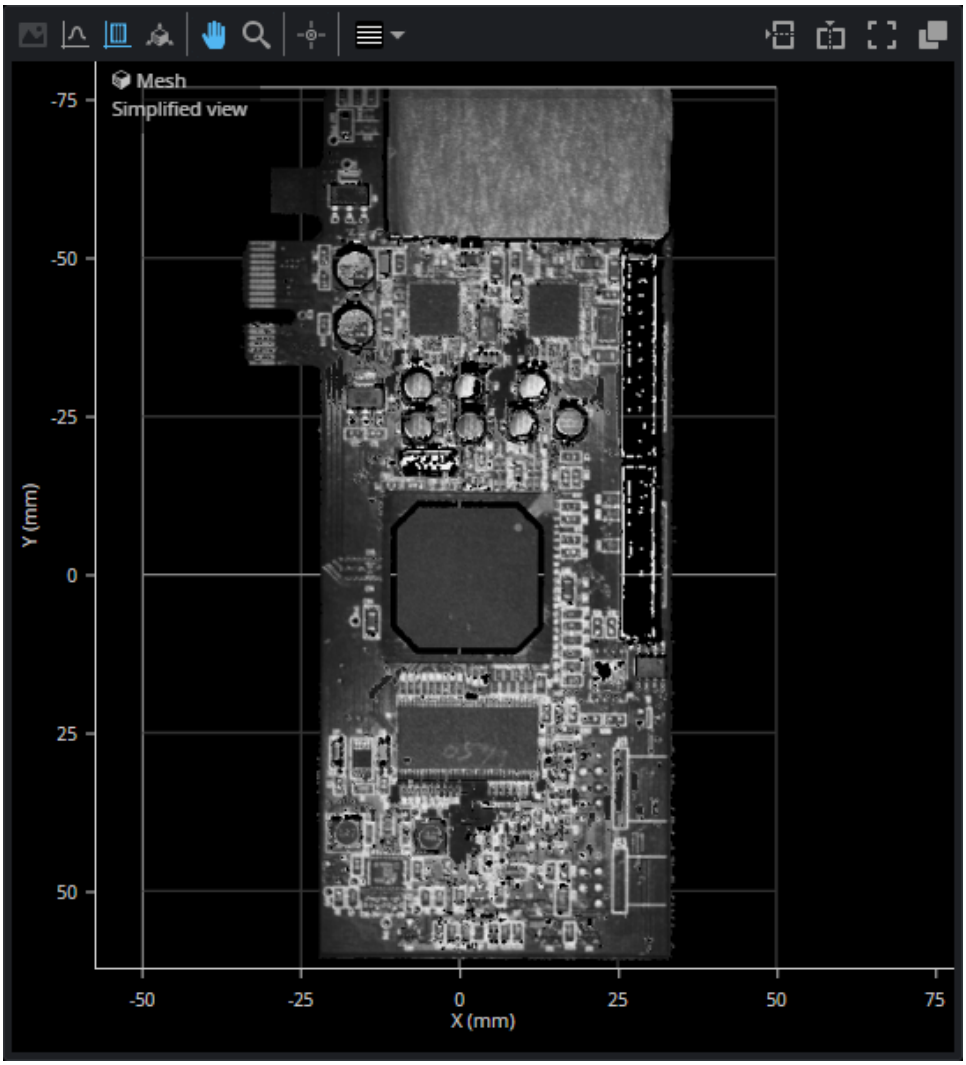
Sensors can produce intensity images that measure the amount of light reflected by an object. An 8-bit intensity value is output for each data point in the scan data. GoPxL uses the same coordinate system as the data points.

To display intensity data, choose either of the last two display options in the data viewer display options. If intensity data is not available, the icons are grayed out. Intensity data can only be shown with Surface data.



Intensity-only, and Surface with intensity buttons

> To be able to display intensity data, you must enable **Acquire Intensity** in the **Scan Mode** panel before scanning. For more information, see *Scan Modes and Intensity* on page 139.


Mesh
Simplified view
-75
-50
-25
0
25
50
Y (mm)
-50
-25
0
25
50
75
X (mm)

Recording and Working with Scan Data

You use the buttons in the right half of the global toolbar, at the top of the interface, to do the following:

- start and stop acquisition
- record replay data
- enable replay mode and play back replay data
- upload, download, and clear replay data
- use the Replay Editor

> When switching between scan modes, if you have previously recorded data, you must clear it before recording replay data of a different type.



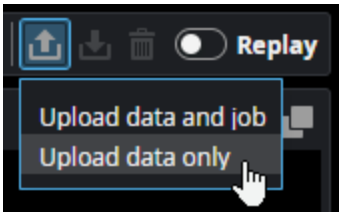
Depending on whether Replay mode (4, below) is disabled or enabled, the control buttons (2, below) change. When Replay is off, the controls let you start and stop data acquisition. When Replay is on, the controls let you play back recorded data.



The global toolbar with Replay disabled (top) and enabled (bottom)

Note that when GoPxL is in Replay mode, the data viewer is displayed with a blue border to help

	Element	Description
1	Frames per second and Repeat frames (looping)	These controls let you set the number of frames per second (fps) during playback, and also whether GoPxL repeats frames, by toggling the Repeat frames button ( ). For more information, see *Playing Back Recorded Data* on page 107.
2	Data playback and recording controls	These controls let you start and stop scanning, start recording and also play back replay data. The number of frames is also indicated. The controls that are available depends on whether Replay mode is disabled or enabled.

	Element	Description
		For information on starting and stopping scans, see *Starting, Stopping, and Recording* below. For information on data playback, see *Playing Back Recorded Data* on page 107.
3	Replay actions	Lets you upload, download, and clear recorded data. When you download replay data from the main toolbar, GoPxL uses the file name convention described in *File Name Conventions* on page 78. You can also add and remove frames with the Replay Editor. For information on the Replay Editor, see *Recording and Working with Scan Data* on the previous page. When uploading data, you can decide whether to only upload data, or to upload both scan data and the associated configuration (tools and other GoPxL settings).  The file type of replay data is .gprec.
4	Replay toggle	Toggles between data acquisition (toggle is to the left) and data playback (toggle is to the right).

Starting, Stopping, and Recording

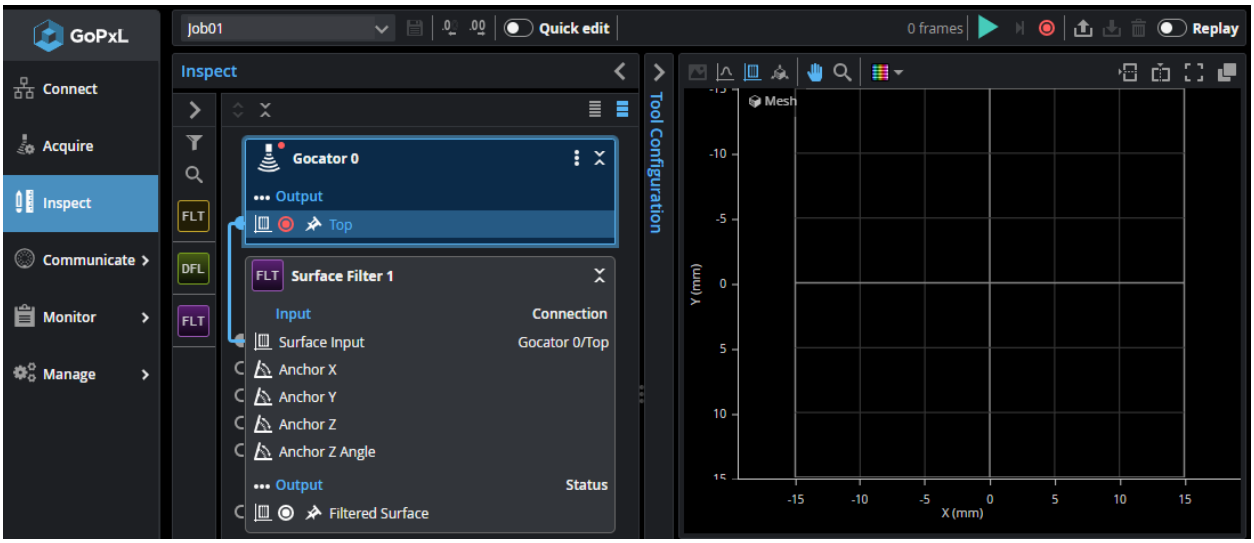
You start and stop data acquisition (and toggle recording) using the data control buttons when Replay is disabled.



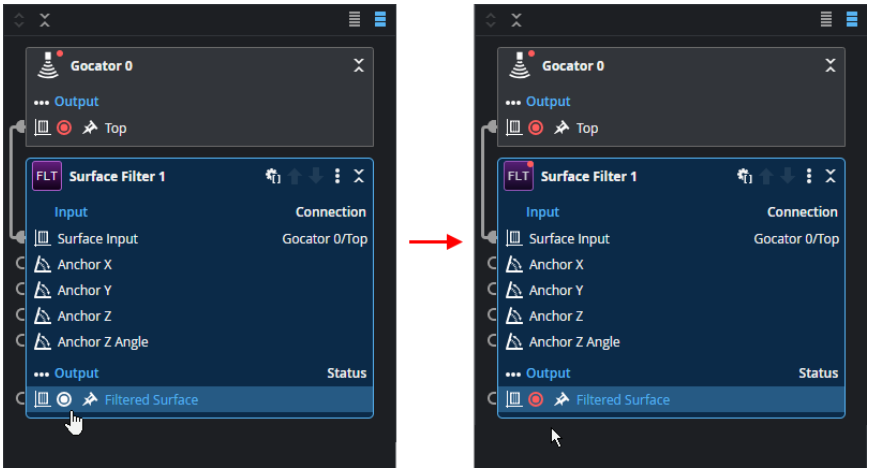
	Element	Description
1	Start or stop scanning	The sensor starts scanning continually when you click the Start button ( ), and stops when you click the Stop button ( ).
2	Single frame (Snapshot)	The sensor acquires a single frame of data. For line profile sensors, you can't acquire a single frame of data if the sensor is in Surface mode, only in Profile or Image mode.

	Element	Description
3	Record data	Toggles recording scan data as it is acquired. The number of frames recorded is displayed next to the Start button. When recording is enabled, the Record data button turns red ( ).

Normally, when you have enabled recording using the Record data button above the data viewer, GoPxL records scan data coming directly from the sensor or group of sensors (in the Tool Diagram, this is called the "sensor block").

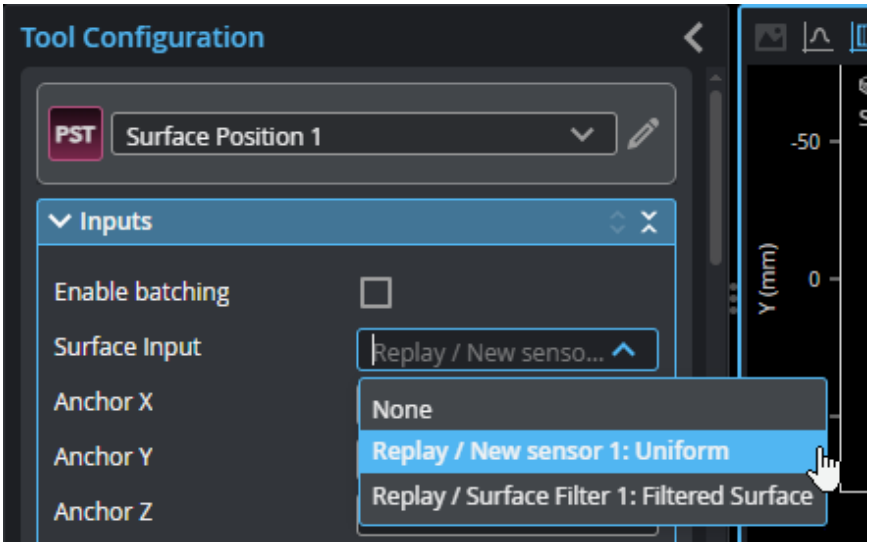


You can also record a tool's output. An example of this is recording the "Filtered Surface" output of a Surface Filter tool. To do this, first enable recording above the data viewer as you would normally do. Then, expand a tool in the Tool Diagram, and click the white Record button next to an output.



When you record data with output recording enabled for any of the tool's outputs, the resulting replay data will contain the scan data coming from the sensors, but also the tool's output data. Once

you've downloaded this replay data and uploaded it to a sensor or PC instance of GoPxL, both sources are available in a tool's input.



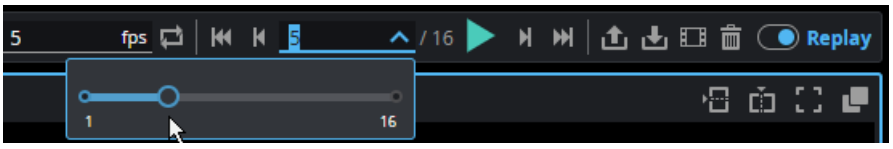
You can choose to disable recording of the data coming from the sensor block to only record a tool's output. This can be useful when you have a long chain of pre-processing tools that performs filtering, transformations, or masking to isolate scan data, and so on. By recording the output of the last tool in the pre-processing chain, uploading that to a PC instance of GoPxL, you can then develop the decision-making part of your application separately, without the overhead of the pre-processing tool chain.

Playing Back Recorded Data

When Replay mode is enabled, you play recorded data back using the data control buttons.



	Element	Description
1	Playback rate and repeating frames	When "Repeat frames" ( ) is enabled, scan data is played back in a loop. This is useful when inspecting data or configuring tools using recorded data, and also for demos.
2	Go to previous / first frame	Displays the previous frame or jumps to the first frame of the recorded data and displays it.
3	Frame position	Lets you go to a specific frame, either by entering a value in the field, or by using a slider.

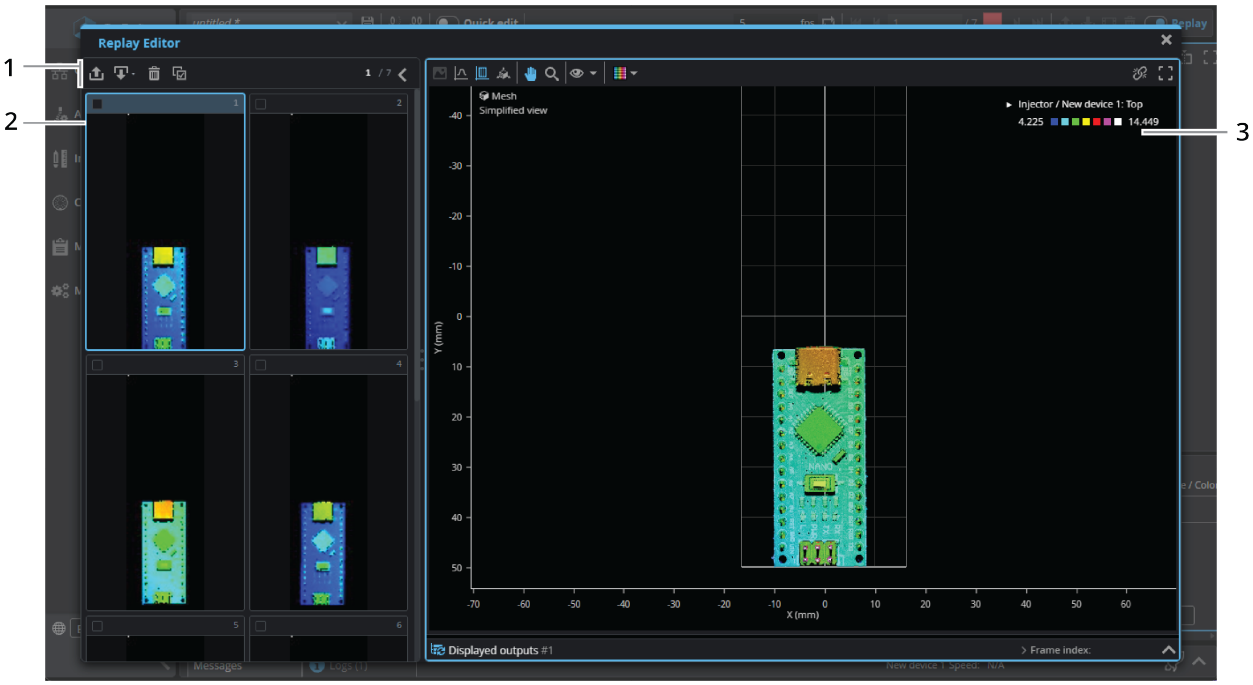
	Element	Description
		
4	Start playback	Starts playing back frames of the recorded data, at the specified number of frames per second.
5	Go to next / last frame	Displays the next frame or jumps to the last frame of the recorded data.

Using the Replay Editor

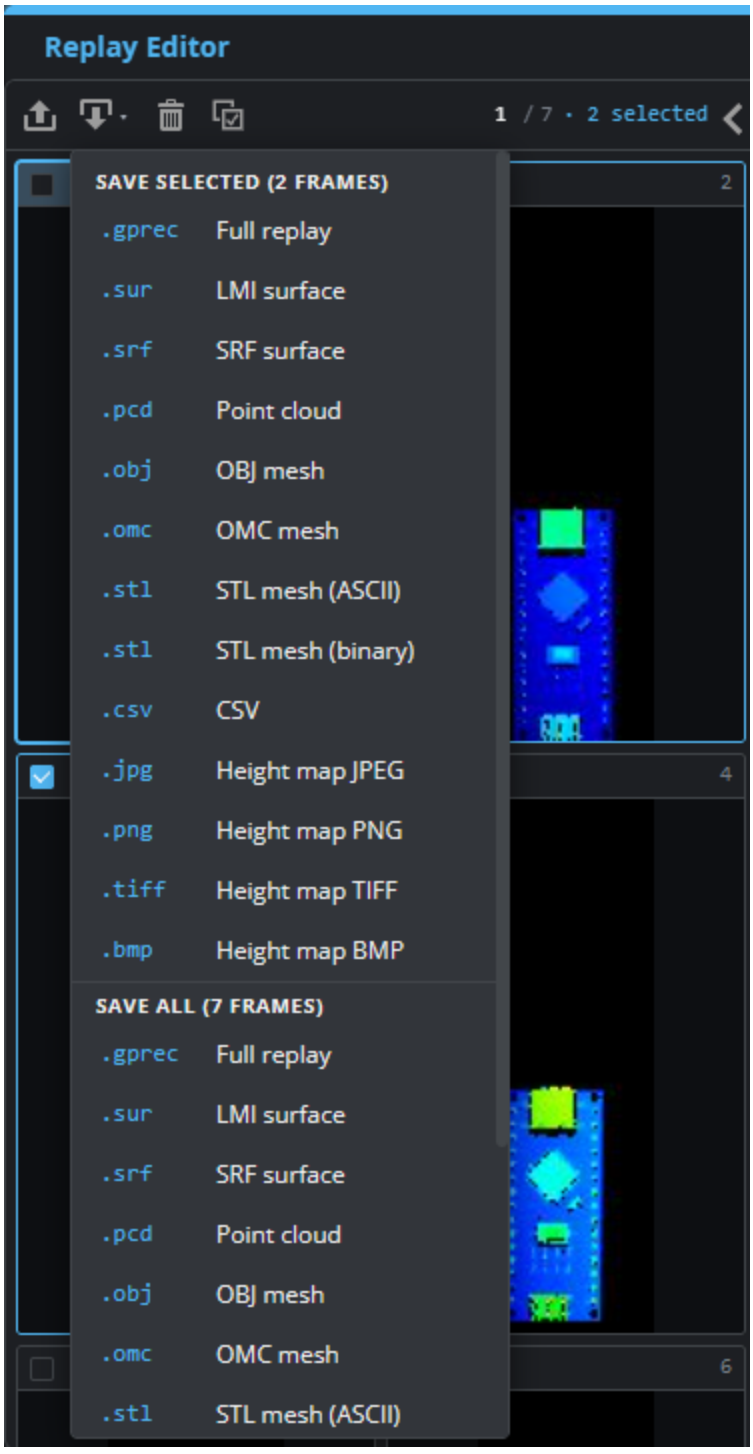
The replay editor lets you add or remove frames of recorded data in the sensor's buffer. For a list of keyboard shortcuts available in the replay editor, see *Replay Editor shortcuts* on page 74.

Some examples of using the Replay Editor:

- Removing unwanted frames to produce a set frames of representative good parts (sometimes called "golden parts") or bad parts for evaluation, or for training with tools like GoPxL Anomaly Detector.
- Adding new frames.
- Preparing a test, training, or demo dataset.



	Element	Description
1	Editor tools	The replay editor tool bar lets perform operations on the frames of data in the sensor's buffer.  **Download** - Lets you download the currently selected frame or frames. You can select multiple frames by clicking the checkbox next to each frame you want to download. Clicking the Download button displays a list of all the available types in two sections (one for the selected frames, another for all frames). Clicking a type downloads the frames (for example, choosing ".gprec" downloads `replayEditor.gprec`).

Element	Description
	 **Upload** - Lets you upload either a .gprec file or individual image files. Frames are added to the end of the list of frames in the replay editor.

	Element	Description
		 **Delete frames** - Deletes the currently selected frames.  Select all / Deselect all - Toggles between selecting all frames or deselecting all frames.
2	Frame list	The list of frames in the sensor's buffer. You can collapse the list to make more room in the interface.
3	Data viewer	The currently selected frame is displayed in the data viewer. For more information on the data viewer, see *Data Viewer* on page 79.

Before using the replay editor, you must either record some frames of data, or upload previously recorded data.

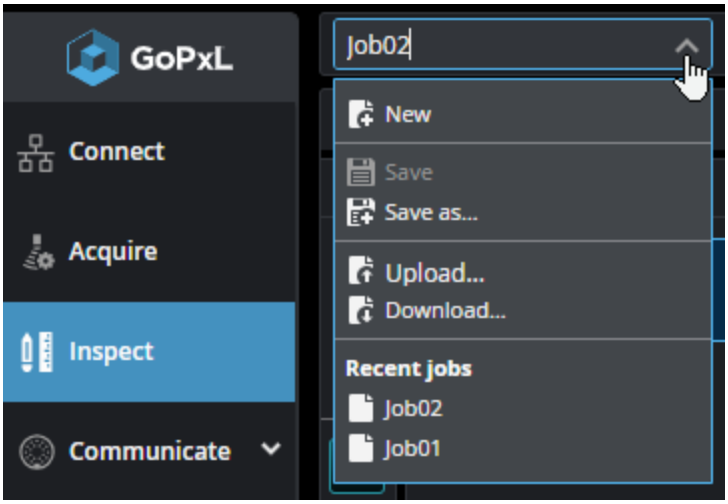
Creating, Saving and Loading Jobs

Jobs contain sensor, tool, and control configurations for a particular inspection or quality control application. (For more information, see *Configuring GoPxL* on page 118.)

A sensor can store many jobs, subject to sensor storage limitations. Running GoPxL on a PC or using GoMax NX provides more storage. 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 you can change in GoPxL's web interface are temporary until saved in a job file. If there is a job file that is set as the default, it will be loaded automatically when GoPxL starts.

GoPxL lets you perform several job-related operations from the global toolbar at the top of the interface regardless of the currently selected page:



Other operations are available on the **Manage** > **Jobs** page. For more information on jobs and working with them, see *Jobs* on page 1179.

Pro Tools

The GoPxL Pro Tools inspection toolsets address specific application challenges in industrial environments. Each toolset combines a selection of AI-based or Image tools and rule-based methods deployable on GoMax® or an industrial PC.

Tools Included by Package

The following table lists the tools available with the different Pro Tools licenses. You can use these tools without a license on recorded data to evaluate them, but to use them on live data, you need one of these licenses.

Note that you can use all other tools without restrictions.

	Pro Tools License		
	GoPxL Pro AI (301295)	**GoPxL Pro Image Inspection (301382)**	**GoPxL Pro AI and Image Tools (301384)**
Image Classifier	✓		✓
Image Object Detector	✓		✓
Surface Anomaly Detector	✓		✓
Image Anomaly Detector	✓		✓
Surface Image Converter[1]	✓	✓	✓
Image Arithmetic		✓	✓
Image Barcode		✓	✓
Image Blob		✓	✓
Image Circle		✓	✓
Image Converter		✓	✓
Image Feature Mask		✓	✓
Image Filter		✓	✓
Image Intensity		✓	✓
Image Line		✓	✓
Image Line Count		✓	✓
Image Matrix Code		✓	✓
Image OCR		✓	✓
Image Pattern		✓	✓
Image Pixel Count		✓	✓

	Pro Tools License		
	GoPxL Pro AI (301295)	**GoPxL Pro Image Inspection (301382)**	**GoPxL Pro AI and Image Tools (301384)**
Image QR Code		✓	✓
Image Track		✓	✓
Image Transform		✓	✓

[1]Surface Image Converter is available free of charge on all sensors.

Installing CodeMeter Runtime for GoPxL for PC

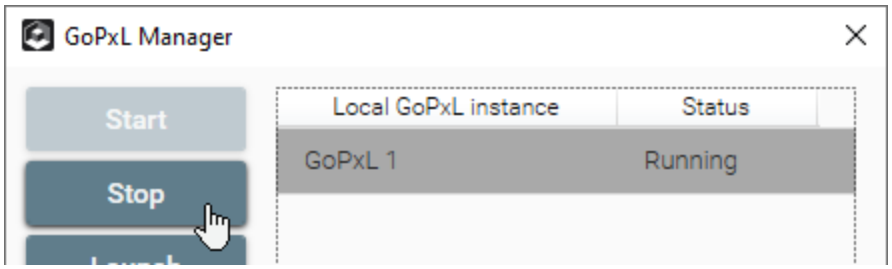
If you are using a Pro Tools tool in GoPxL for Windows while accelerating a sensor and on live data, you must install the CodeMeter runtime, which is required by the Pro Tools license and dongle. You install the runtime from GoPxL Manager.

To install the CodeMeter runtime

1. If GoPxL it is not already running, start GoPxL Manager.

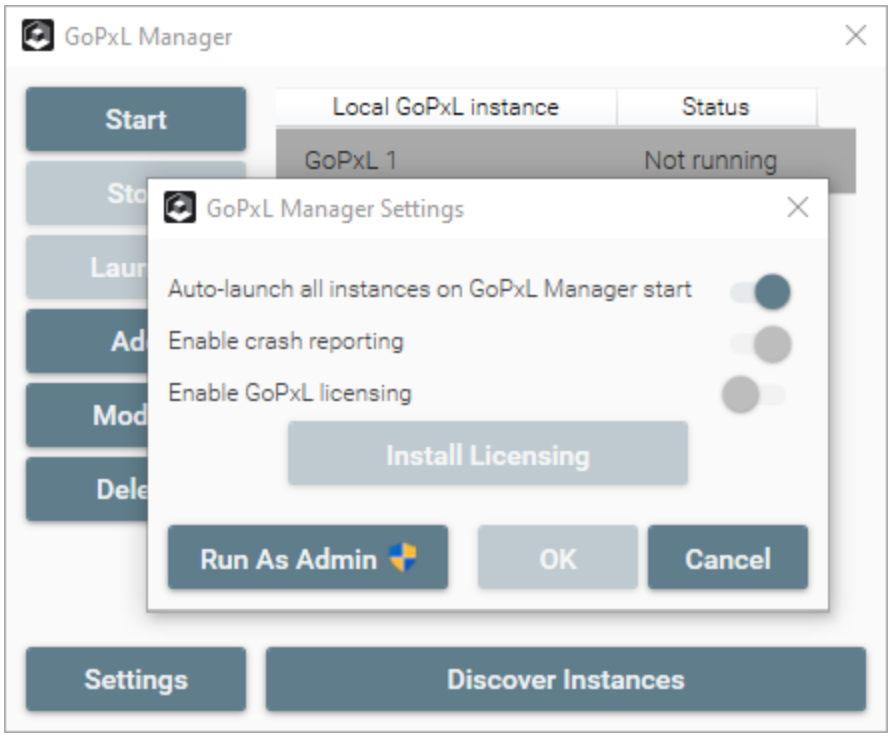
   For more information, see *Running GoPxL on a Windows PC* on page 1188.

2. Stop any running instances of GoPxL.

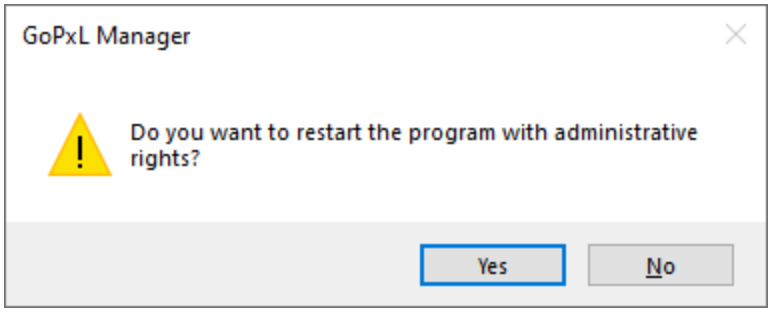


   When an instance stops, its status becomes "Not running."

3. Click **Settings**, and in the **GoPxL Manager Settings** dialog, click **Run As Admin**.



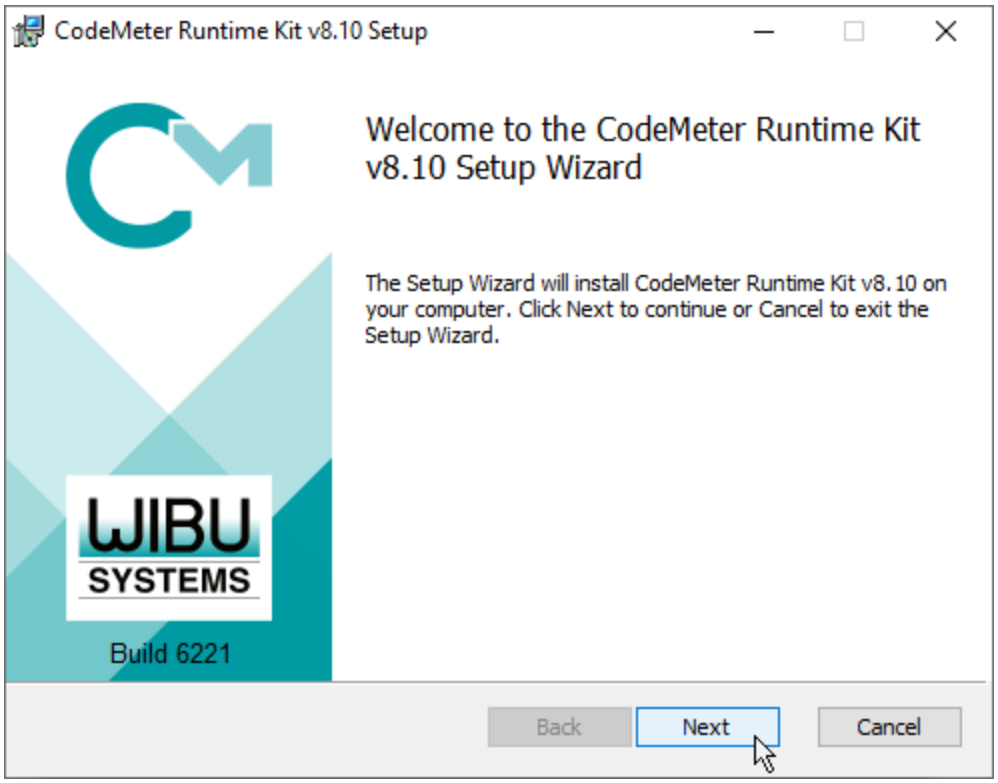
4. Answer Yes in the dialog that appears.



You will need to click Yes in the User Account Control dialog.

5. Click **Settings** again, and in the dialog, toggle on **Enable GoPxL licensing** and click **Install Licensing**.

The CodeMeter Runtime setup wizard launches.

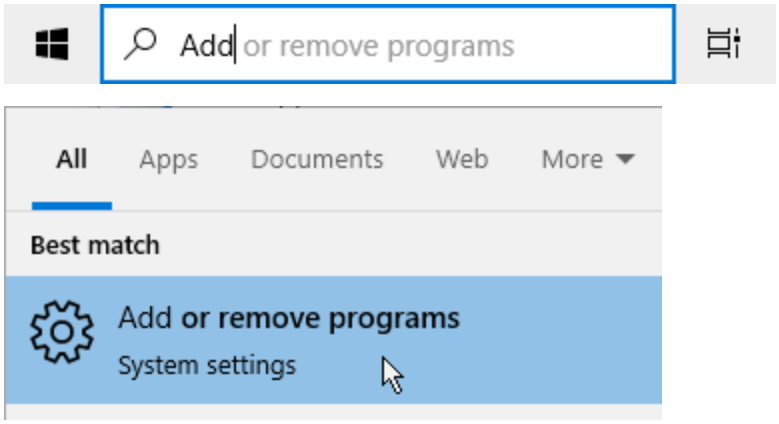


6. Go through the steps in the wizard to install the runtime.

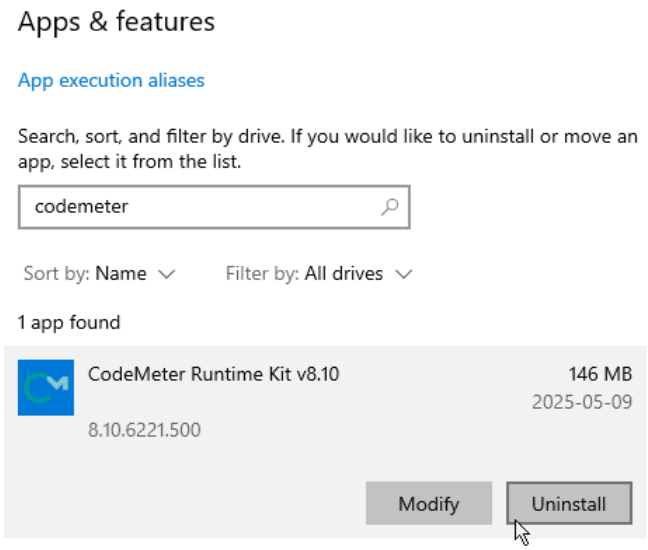
   You will need to accept the license agreement before continuing with the installation, and choose whether CodeMeter will be installed for all users on the PC. Leave the defaults in the Custom Setup step.

If you no longer need to use Anomaly Detector on your PC, remove the licensing software:

1. On your PC desktop, in the Search field, type "Add" and click "Add or remove programs" in the "Best match" list.



2. In the "Search this list" field, type "codemeter", click CodeMeter Runtime Kit to expand it, click **Uninstall**, and follow the instructions.


Apps & features
App execution aliases
Search, sort, and filter by drive. If you would like to uninstall or move an app, select it from the list.
codemeter
Sort by: Name
Filter by: All drives
1 app found
CodeMeter Runtime Kit v8.10
146 MB
2025-05-09
8.10.6221.500
Modify
Uninstall

Configuring GoPxL

The following sections describe how to configure GoPxL using its web interface.

GoPxL can run on a sensor, or you can run GoPxL on a Windows PC and connect to a sensor running GoPxL from the application (in order to increase performance, for example). When GoPxL runs on a PC, we call this a local or PC instance of GoPxL. For applications where performance is critical, you should use the PC version of GoPxL to improve sensor performance. With some sensor models, such as Gocator 4000 and Gocator 5500 series sensors, LMI *strongly* recommends using the PC version of GoPxL (or a GoMax accelerator).

To configure GoPxL when it is running on-sensor, you connect directly to the sensor's IP address (by default, 192.168.1.10). To configure GoPxL when it is running on a PC, you connect to the IP address of the PC instance. If you can't connect to GoPxL, use the GoPxL Discovery tool to find it and launch the GoPxL interface (for more information, see *GoPxL Discovery Tool* on page 1290).

The interface of the sensor and the PC versions of GoPxL are nearly identical. For this reason, you can use this section to configure sensors whether you are using GoPxL on a sensor or on a PC. If you need to run the PC version of GoPxL, see *Running GoPxL on a Windows PC* on page 1188 first.

For information on configuring GoPxL programmatically, see *GoPxL SDK and REST API* on page 1234.

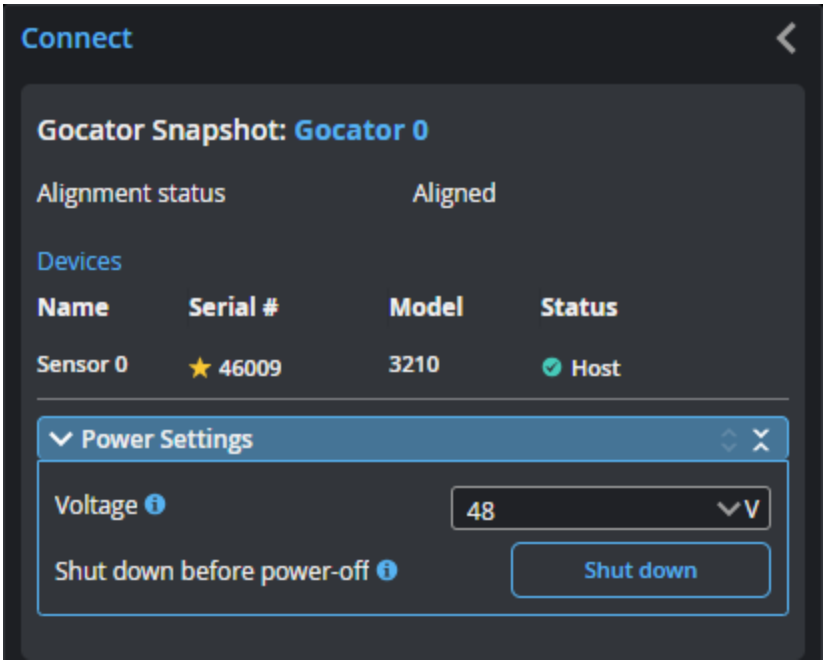
Connect - Creating a System & Networking

You use the **Connect** page to see available devices, create a sensor system containing two or more sensors, and also configure network settings. You can also identify devices that are unavailable (either because of a GoPxL version mismatch, or because a sensor is already running through another PC instance of GoPxL). For information on running a sensor through a PC instance of GoPxL, see *Running GoPxL on a Windows PC* on page 1188.

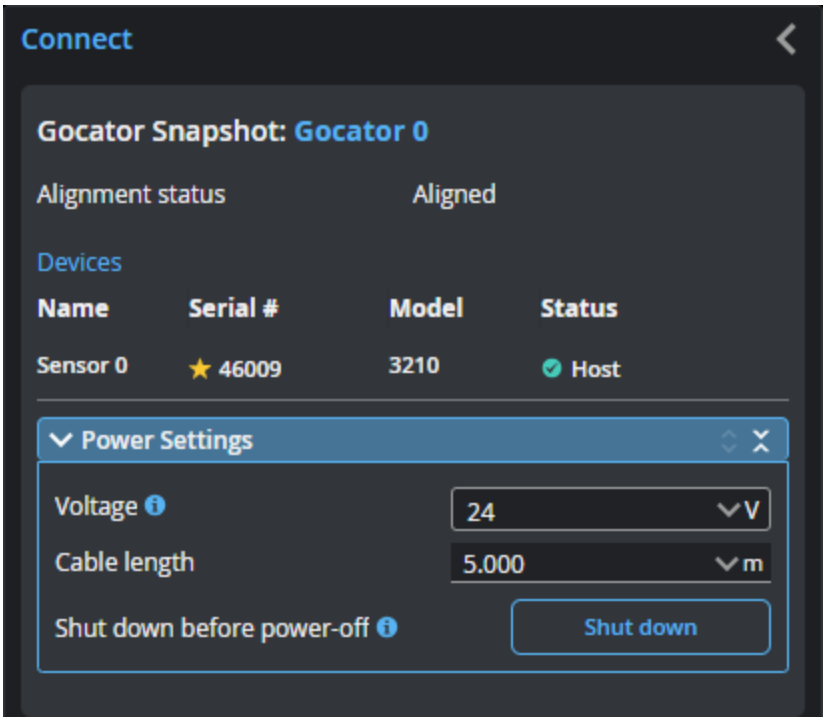
GoPxL does not currently support mixed-model multi-sensor systems.

Configuring Power

LMI recommends running Gocator 3210A, 3210B, and 3506B sensors at 48 VDC; when running these sensors at 48 VDC, you should make sure to set the **Voltage** setting in GoPxL, on the **Design** page to 48 V.



Although you can run Gocator 3210A, 3210B, and 3506B at 24 VDC, the sensor's sensitivity is reduced because the maximum intensity of the projector is reduced; this effect is increased as the cable length increases. If you decide to run one of these sensors at 24 VDC, you must set **Voltage** to 24 V, and then set the cable length to the length of the Power and Ethernet cable you are using with the sensor.



Note that setting the voltage to 48V when only supplying 24V can result in the sensor failing to scan.

Projector Shutdown

Gocator 3200 and 3500 sensors with B revision hardware and later provide a projector mirror parking feature. Use this feature to help reduce mechanical wear during planned shutdowns and frequent power cycling (for example, more than once a day).

In these cases, before removing power from the sensor, click **Shut down...**, after which the connection to the sensor will drop.

To restart the sensor after performing a shutdown, you must power cycle the sensor.

	Shutdown is not available or required on laser line profile sensors.

Network Settings

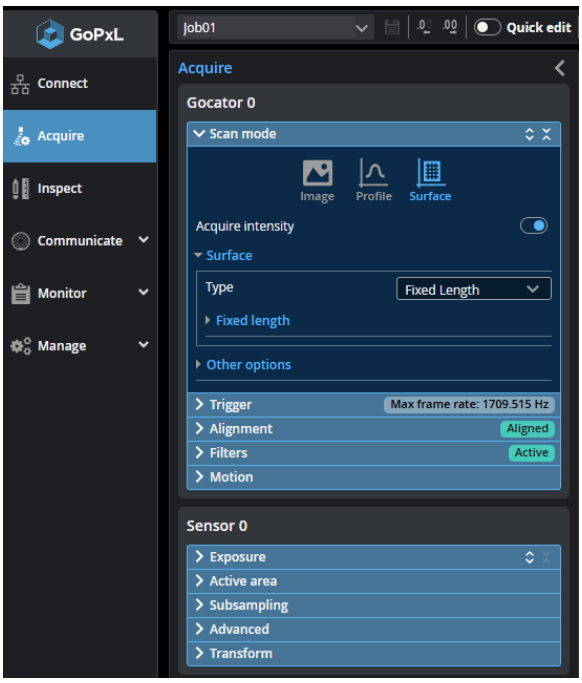
To configure the network settings:

1. Go to the **Connect** page.
2. Specify the type (Manual or DHCP), 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. Click **Save Changes...**.
   You will be prompted to confirm your selection.

The sensor restarts, using the *new* IP address. It may take several seconds for the sensor to become available.

Acquire - Alignment and Acquisition

You configure acquisition and align sensors on the **Acquire** page.



Scan settings are displayed in two sections: the top section, by default called **Gocator 0**, and another section called **Sensor 0** section by default.

Settings such as the scan mode, X spacing intervals, and triggering, which must apply to *all* sensors in a multi-sensor system (that is, as a sensor group), are in the top section. For consistency, in single-sensor systems and with models such as Gocator 3x00 that don't support multi-sensor grouping, these settings are separated from the others in this section.

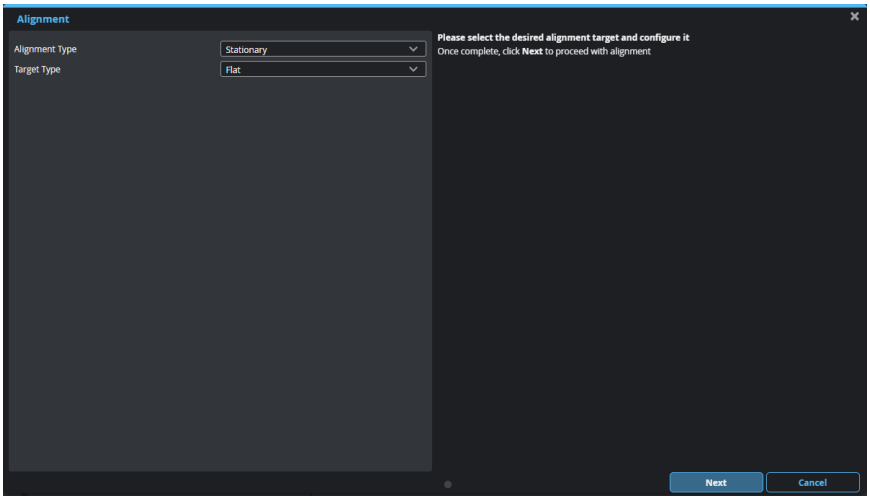
Note that the sensor system's maximum possible frame rate is displayed in this section at the top of the Trigger section. This value changes depending on some of the settings in the **Sensor {n}** section (see below).

Aligning Sensors

Alignment is the process you use to automatically calculate transformations (rotations and translations / offsets) that are applied to a sensor's scan data while it is scanning targets. If you do not correct for these rotations, scan data may be too distorted for your application, and your measurements may therefore be inaccurate. Alignment is often required for various reasons:

- To compensate for sensor mounting inaccuracies relative to the intended scanning surface.
- To set a Z (height) reference plane, using a flat surface or an alignment target.
- To accommodate intentional rotation of sensors.

Once you click the **Align...** button, an alignment "wizard" goes through the steps required.



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

Aligning Sensors with up to 5 Degrees of Freedom

Alignment is configured and performed from the **Acquire** page. If you use a plate alignment target with two holes, alignment will include all 6 degrees of freedom; otherwise, sensors are aligned to only 5 degrees of freedom.

For information on coordinate systems, see *Coordinate Systems* on page 51.

Snapshot sensors only support Stationary alignment.

A sensor can be in one of two alignment states: Unaligned and Aligned. An indicator on the **Alignment** panel displays UNALIGNED or ALIGNED, depending on the sensor's state. A sensor's alignment state determines its coordinate system; for more information on coordinate systems, see *Coordinate Systems* on page 51.

If you perform a high-accuracy tool-based sensor alignment, the **Alignment** panel will still display UNALIGNED. *This is normal.*

Alignment State

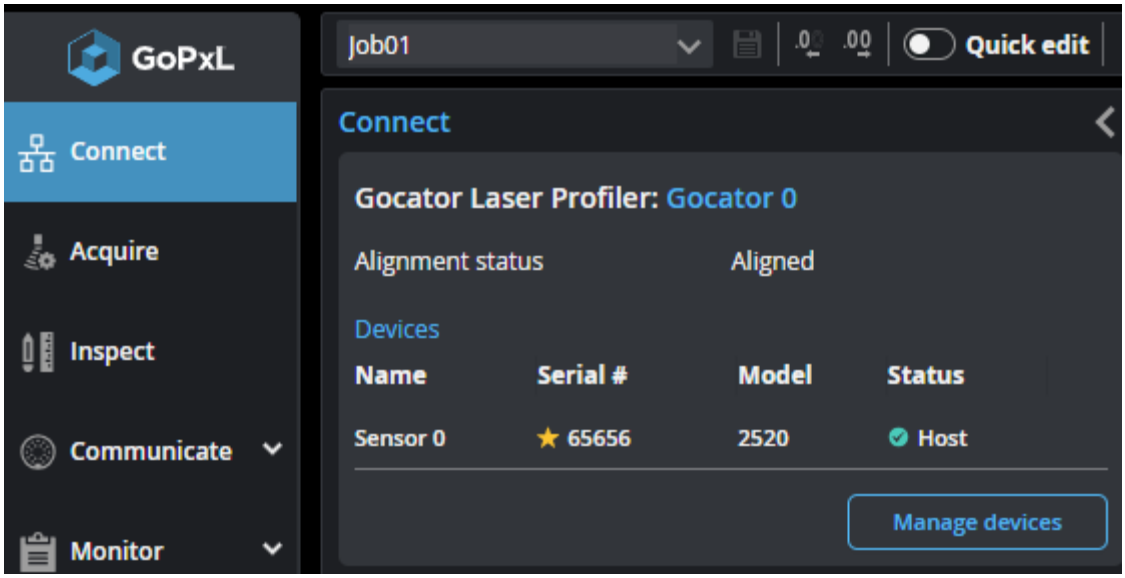
State	Explanation
Unaligned	The sensor or sensor system is not aligned. Data points are reported in sensor coordinates.
Aligned	The sensor is aligned using the alignment procedure (described below) or by manually modifying the values under **Transform** on the **Acquire** page (for more information, see *Transformations* on page 137). Data points are reported in system coordinates.

Once you have performed the alignment procedure on the **Acquire** page, the calculated transformation values are displayed under **Transform** on the **Acquire** page. For more information on transformations, see *Transformations* on page 137.

To prepare for alignment

1. For dual- or multi-sensor systems, make sure you have done the following:

   On the **Connect** page, add sensors to the system. For information, see *Connect - Creating a System & Networking* on page 119.



   Configure the system's layout, if necessary. For more information, see *Configuring Power* on page 119 and *Connect - Creating a System & Networking* on page 119.

   If the profiling lines of the sensors overlap, make sure to check the **Exposure Multiplexing** option. Otherwise, the profiling line from one sensor will be detected by other sensors and cause the alignment procedure to fail or be inaccurate; for more information, see *Connect - Creating a System & Networking* on page 119.

2. Go to the **Acquire** page.
3. Ensure that all sensors will have a clear view of the alignment target surface.
4. Perform a preliminary scan of the alignment target to evaluate the quality of the scan data.

   Doing this will help ensure that the alignment process succeeds. In the next step, adjust the settings based on the scan data of the alignment target.

5. If necessary, adjust the settings on the **Acquire** page to get the best data possible from the scans of the alignment target.

   Some examples of the settings you may need to adjust are:

   - Exposure duration (to make sure the target is clearly represented in the scan data). Typically, only a single exposure is needed. For more information, see *Single Exposure* on page 151.
   - Active area. For more information, see *Active Area* on page 154.

- Spacing: Make sure to use the sensor's full X resolution (spacing interval is set to full resolution). For more information, see *Spacing on X and Y (Uniform Spacing)* on page 140.

Because the alignment procedure automatically uses Time triggering, regardless of the settings in the **Trigger** panel, you can leave these settings as is and configure them later. For information on triggering, see *Triggers* on page 141.

6. Perform the steps in *Performing Stationary Alignment* below.

Performing Stationary Alignment

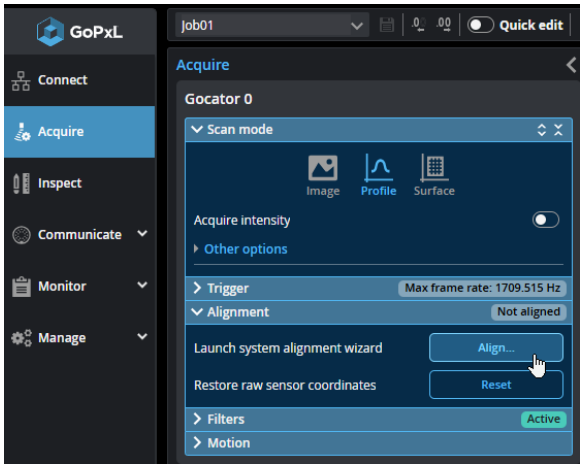
To perform stationary alignment

1. Make sure that the alignment surface (whether it's the surface of a conveyor or of an alignment target) is within the sensor's measurement range.

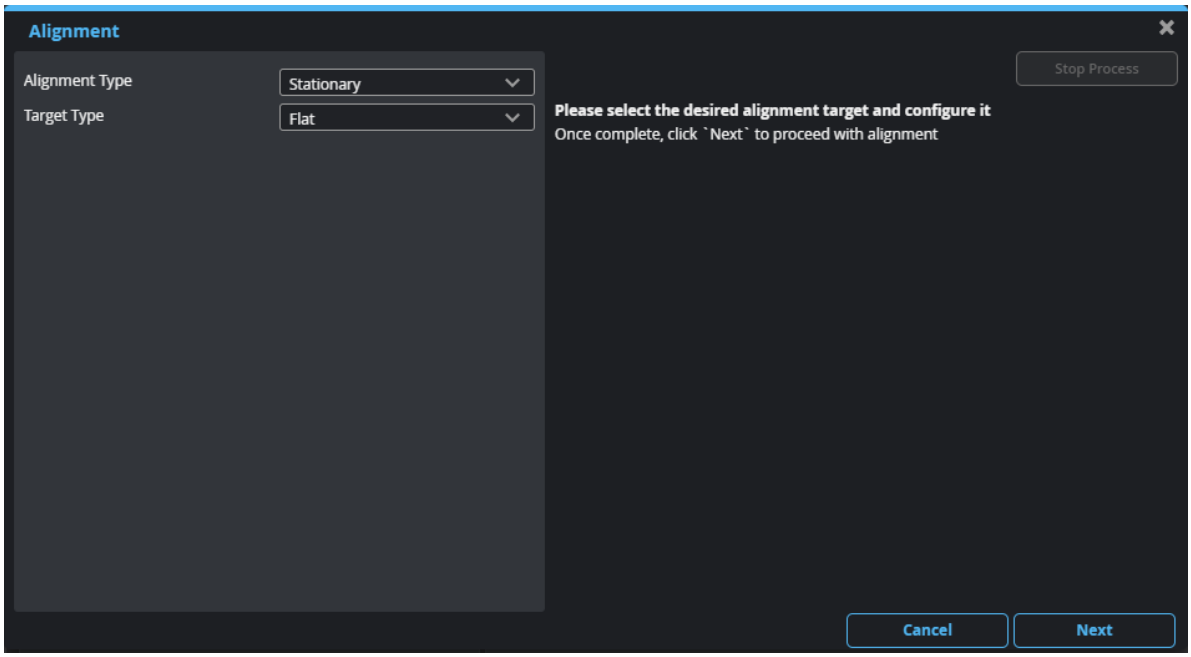
   To determine this, in the sensor's web interface, click **Start** and observe whether the Range LED on the sensor is illuminated. If you have set the trigger source to Encoder under **Acquire** > **Trigger**, temporarily set it to Time for this. Be sure to stop the sensor after confirming the target is in range by clicking the **Stop** button.

   Alternatively, you can determine the correct distance to the scan surface by consulting the sensor's measurement range specifications (see *Sensors* on page 1332), and measuring the physical distance between the scan surface and the sensor.

2. Go to the **Acquire** page.
3. (Optional) If a previous alignment is present, click the **Reset** button to clear it.
4. If you are using an alignment target, place it below the sensor or sensors.
5. Click **Align...**.



The Alignment dialog appears.



6. Set **Alignment type** to Stationary.
7. Choose an alignment type in **Alignment type**.
   - **Flat Surface**: Use this to align to a simple flat surface. For more information, see *Stationary Flat Surface* below.
   - **Stationary Plate**: Use this to align to an alignment plate you have manufactured. For more information, see *Stationary Plate* below.
8. Click the **Next** button.

   If the alignment fails, check the settings described in *To prepare for alignment* on page 124 and repeat the steps described here.
9. Inspect alignment results.

   Check the alignment results under **Transform**.

   For information on how alignment affects the coordinate system used by sensors, see *Coordinate Systems* on page 51.

Stationary Flat Surface

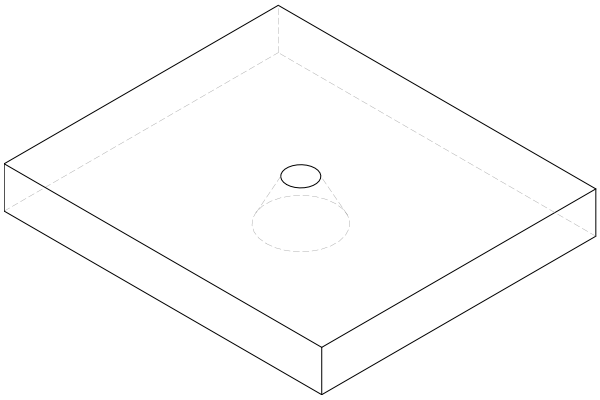
No additional settings are required for this alignment method. Note however that this type of alignment expects to receive flat scan data. Therefore, if the surface is curved, the alignment will be inaccurate. The surface should also be clear of debris and damage. The alignment results in 5 degrees of freedom (it excludes Z angle corrections).

Stationary Plate

You can align a G3 sensor using an alignment plate with one or two holes. The table below describes the transformations resulting from aligning using each kind of target.

TARGET SPECIFICATIONS

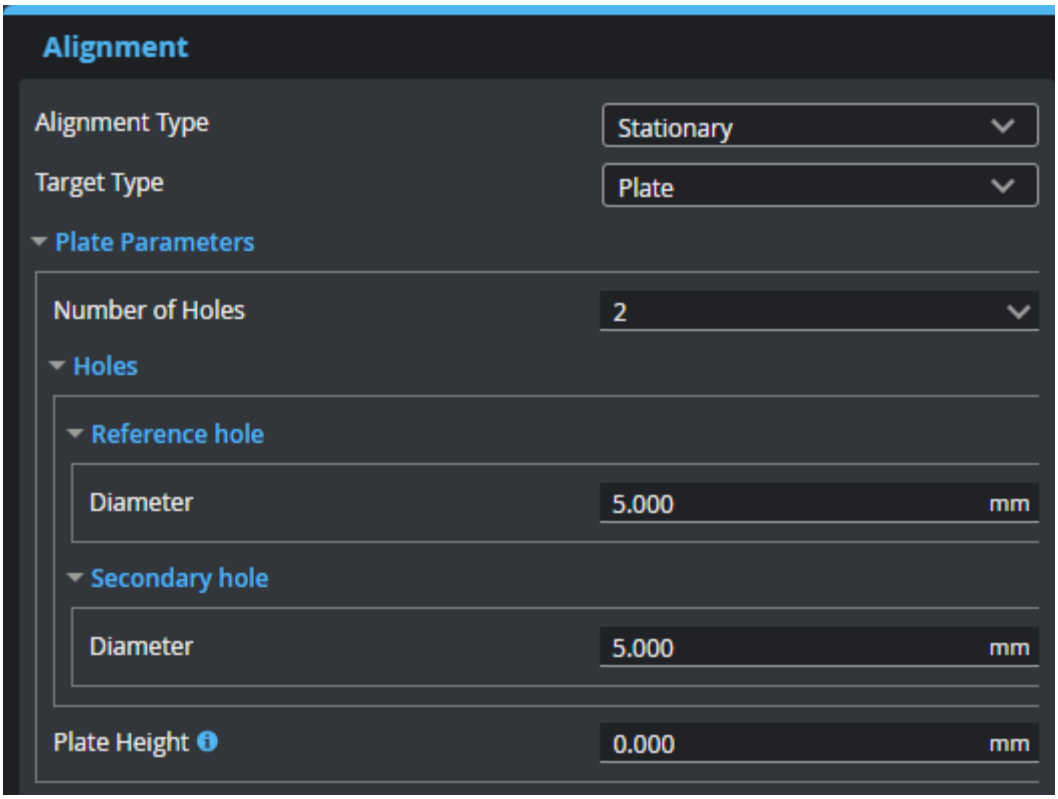
Make sure of the following when manufacturing an alignment plate:



- Holes must be as sharp as possible: avoid bevels.
- The size of the holes should be more than 10 times the XY resolution of the sensor; for the XY resolution of your sensor, see specifications of the sensor in *Sensors* on page 1332.
- Sensors must capture as little data from the inside of a hole as possible. Either countersink holes from the opposite side of the plate, or paint the insides of the holes with a flat black paint. Otherwise, although the alignment should succeed, it will not be as accurate: it may result in unwanted offsets or angles in the transformations.
- The recommended flatness of alignment targets for accurate Y angle is roughly the Z resolution rating of the sensor. If the bar target is curved, it will introduce an apparent Y angle in the sensor alignment. For sensor Z resolution, see the specifications for your sensor in *Sensors* on page 1332.
- It is not necessary to machine the plate height to a high tolerance. Height can instead be controlled during measuring rather than at manufacture. Only flatness is important. If the zero level is not critical for the measurement, then standard machining tolerances can be used. Alternatively you can machine to a low tolerance and measure the value to a high precision to save cost.
- Plates should be painted with flat light grey or white paint to improve data capture (by reducing the possibility of reflections and improving profile data of the bar surface).

CONFIGURING GOPXL FOR PLATE ALIGNMENT

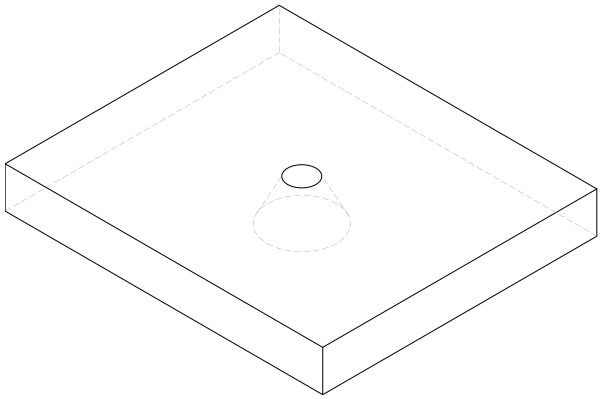
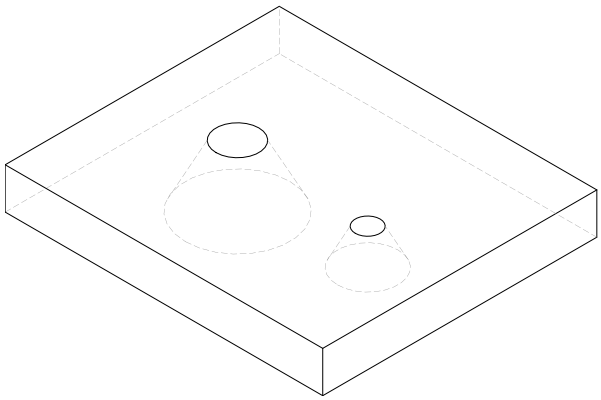
Configure the characteristics of the target; for more information on these settings, see below.



- **Number of Holes** is the number of holes in the alignment plate. For information on the transformations resulting from different types of plates, see *Transformations resulting from plates* below.
- **Diameter**: Sets the diameters of the holes. If you are using a plate with two holes, they must be of different sizes.
- **Plate Height**: The alignment procedure determines the average Z height of the alignment target's top surface and uses the value specified in **Height** to offset the coordinate system from that average Z height; in effect, the bottom of the alignment target becomes the Z origin (the zero reference level). This value can be negative.

Transformations resulting from plates

Target	Resulting Transformations	Explanation
Plate	X offset Y offset Z offset X angle Y angle	The orientation of the surface surrounding the hole determines the X angle and Y angle transformations. Z angle is not calculated. X and Y offsets are calculated from the X and Y angles.
Plate (single reference	X offset	The center of the hole defines the location of the X, Y and Z origins (the

Target	Resulting Transformations	Explanation
hole)	Y offset Z offset X angle Y angle	offsets). The orientation of the surface surrounding the hole determines the X angle and Y angle transformations. Z angle is not calculated. 
Plate (with two reference holes)	X offset Y offset Z offset X angle Y angle Z angle	The sizes of the primary and the secondary holes must be different. The center of the primary hole defines the location of the X, Y, and Z origins (the offsets). The orientation of the surface determines X Angle and Y Angle. The angular position of the secondary hole on the XY plane, relative to the primary hole, determines Z Angle. 

Clearing Alignment

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

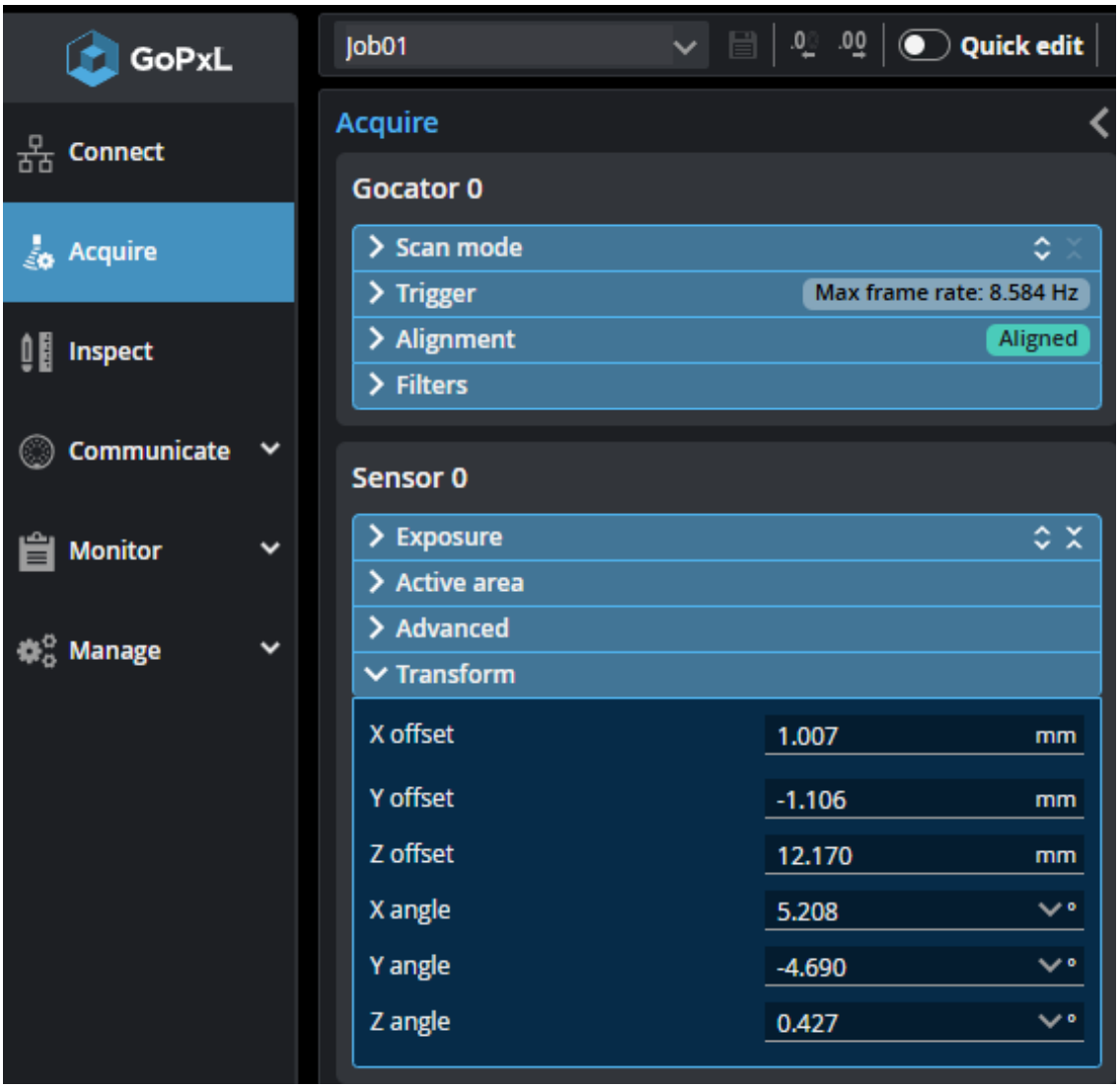
To clear alignment:

1. Go to the **Acquire** page.
2. In the **Alignment** panel, click **Reset**.

   GoPxL clears the alignment. Sensors revert to using sensor coordinates instead of aligned system coordinates.

Transformations

The transformation values determine how data is converted from unaligned coordinates to aligned coordinates (for an overview on coordinate systems, see *Coordinate Systems* on page 51). Transformation values are set automatically when you align a sensor using an alignment routine on the **Acquire** page. The resulting transformations are displayed in the expandable **Transform** section on that page.



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.
X Angle	Specifies the tilt around the X axis.
Y angle	Specifies the tilt around the Y axis.
Z angle	Specifies the tilt around the Z axis.

When applying the transformations, the data is first rotated around X (clockwise, with the X axis toward the viewer), then Y (counterclockwise), and then Z (clockwise), 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). For more information on trigger spacing, see *Trigger Settings* on page 143.

Performing the Alignment

Alignment is configured and performed from the **Acquire** page. If you use a plate alignment target with two holes, alignment will include all 6 degrees of freedom; otherwise, sensors are aligned to only 5 degrees of freedom.

For information on coordinate systems, see *Coordinate Systems* on page 51.

Snapshot sensors only support Stationary alignment.

A sensor can be in one of two alignment states: Unaligned and Aligned. An indicator on the **Alignment** panel displays UNALIGNED or ALIGNED, depending on the sensor's state. A sensor's alignment state determines its coordinate system; for more information on coordinate systems, see *Coordinate Systems* on page 51.

> If you perform a high-accuracy tool-based sensor alignment, the **Alignment** panel will still display UNALIGNED. *This is normal.*

Alignment State

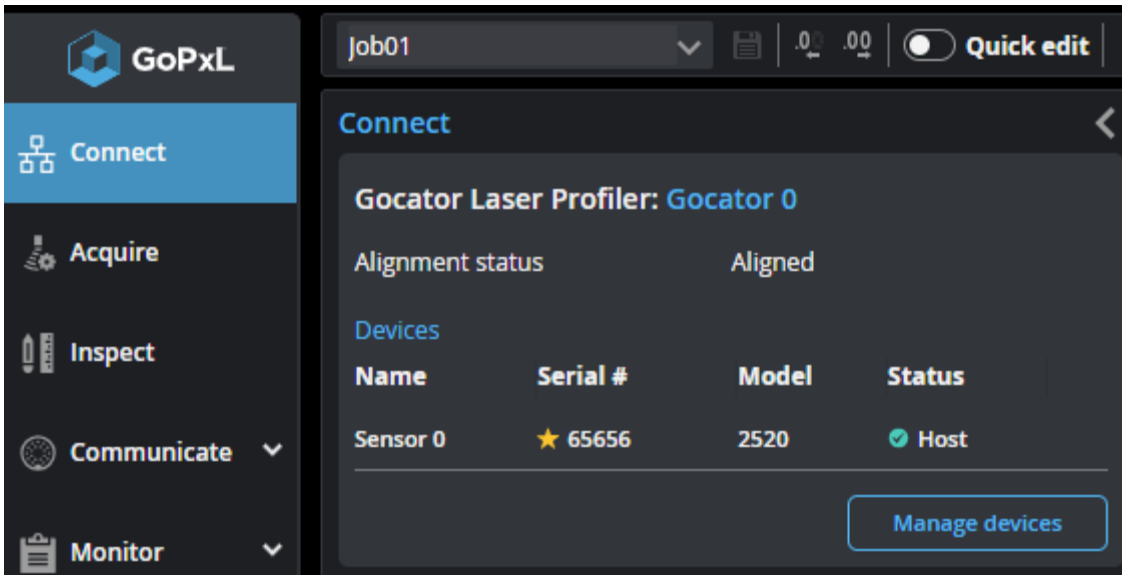
State	Explanation
Unaligned	The sensor or sensor system is not aligned. Data points are reported in sensor coordinates.
Aligned	The sensor is aligned using the alignment procedure (described below) or by manually modifying the values under **Transform** on the **Acquire** page (for more information, see *Transformations* on page 137). Data points are reported in system coordinates.

Once you have performed the alignment procedure on the **Acquire** page, the calculated transformation values are displayed under **Transform** on the **Acquire** page. For more information on transformations, see *Transformations* on page 137.

To prepare for alignment

1. For dual- or multi-sensor systems, make sure you have done the following:

   On the **Connect** page, add sensors to the system. For information, see *Connect - Creating a System & Networking* on page 119.



Configure the system's layout, if necessary. For more information, see *Configuring Power* on page 119 and *Connect - Creating a System & Networking* on page 119.

If the profiling lines of the sensors overlap, make sure to check the **Exposure Multiplexing** option. Otherwise, the profiling line from one sensor will be detected by other sensors and cause the alignment procedure to fail or be inaccurate; for more information, see *Connect - Creating a System & Networking* on page 119.

2. Go to the **Acquire** page.
3. Ensure that all sensors will have a clear view of the alignment target surface.
4. Perform a preliminary scan of the alignment target to evaluate the quality of the scan data.

   Doing this will help ensure that the alignment process succeeds. In the next step, adjust the settings based on the scan data of the alignment target.
5. If necessary, adjust the settings on the **Acquire** page to get the best data possible from the scans of the alignment target.

   Some examples of the settings you may need to adjust are:

   - Exposure duration (to make sure the target is clearly represented in the scan data). Typically, only a single exposure is needed. For more information, see *Single Exposure* on page 151.
   - Active area. For more information, see *Active Area* on page 154.
   - Spacing: Make sure to use the sensor's full X resolution (spacing interval is set to full resolution). For more information, see *Spacing on X and Y (Uniform Spacing)* on page 140.

   Because the alignment procedure automatically uses Time triggering, regardless of the settings in the **Trigger** panel, you can leave these settings as is and configure them later. For information on triggering, see *Triggers* on page 141.
6. Perform the steps in *Performing Stationary Alignment* on the next page.

Performing Stationary Alignment

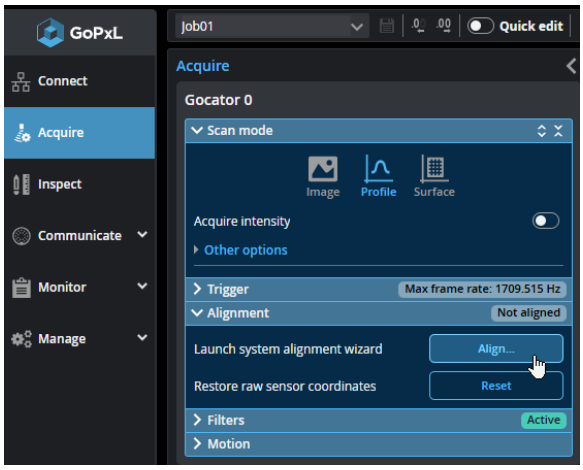
To perform stationary alignment

1. Make sure that the alignment surface (whether it's the surface of a conveyor or of an alignment target) is within the sensor's measurement range.

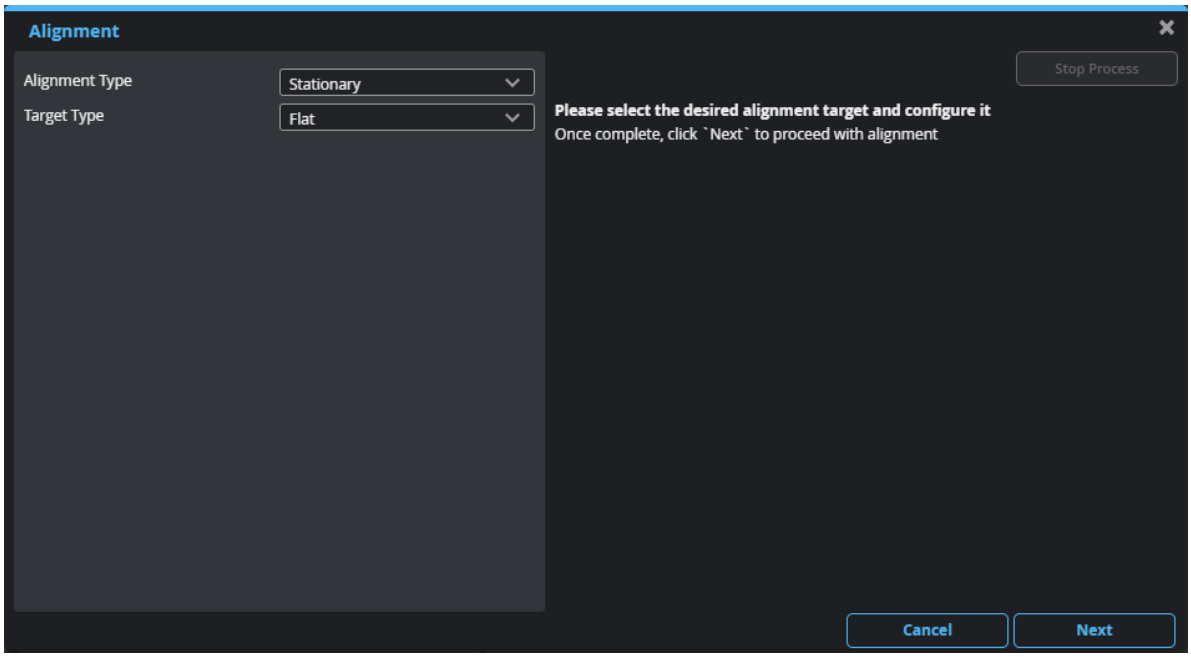
   To determine this, in the sensor's web interface, click **Start** and observe whether the Range LED on the sensor is illuminated. If you have set the trigger source to Encoder under **Acquire** > **Trigger**, temporarily set it to Time for this. Be sure to stop the sensor after confirming the target is in range by clicking the **Stop** button.

   Alternatively, you can determine the correct distance to the scan surface by consulting the sensor's measurement range specifications (see *Sensors* on page 1332), and measuring the physical distance between the scan surface and the sensor.

2. Go to the **Acquire** page.
3. (Optional) If a previous alignment is present, click the **Reset** button to clear it.
4. If you are using an alignment target, place it below the sensor or sensors.
5. Click **Align...**.



   The Alignment dialog appears.



6. Set **Alignment type** to Stationary.
7. Choose an alignment type in **Alignment type**.
   - **Flat Surface**: Use this to align to a simple flat surface. For more information, see *Stationary Flat Surface* below.
   - **Stationary Plate**: Use this to align to an alignment plate you have manufactured. For more information, see *Stationary Plate* below.
8. Click the **Next** button.

   If the alignment fails, check the settings described in *To prepare for alignment* on page 131 and repeat the steps described here.
9. Inspect alignment results.

   Check the alignment results under **Transform**.

   For information on how alignment affects the coordinate system used by sensors, see *Coordinate Systems* on page 51.

Stationary Flat Surface

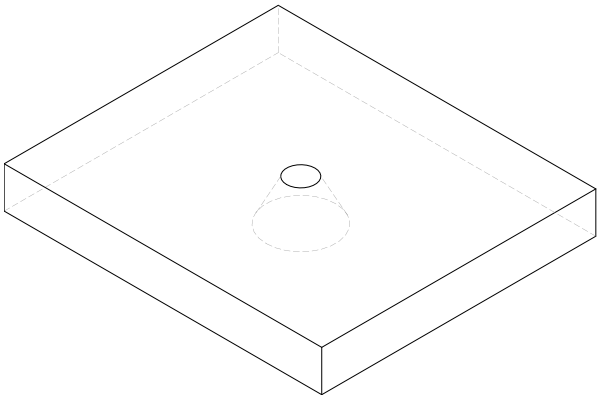
No additional settings are required for this alignment method. Note however that this type of alignment expects to receive flat scan data. Therefore, if the surface is curved, the alignment will be inaccurate. The surface should also be clear of debris and damage. The alignment results in 5 degrees of freedom (it excludes Z angle corrections).

Stationary Plate

You can align a G3 sensor using an alignment plate with one or two holes. The table below describes the transformations resulting from aligning using each kind of target.

TARGET SPECIFICATIONS

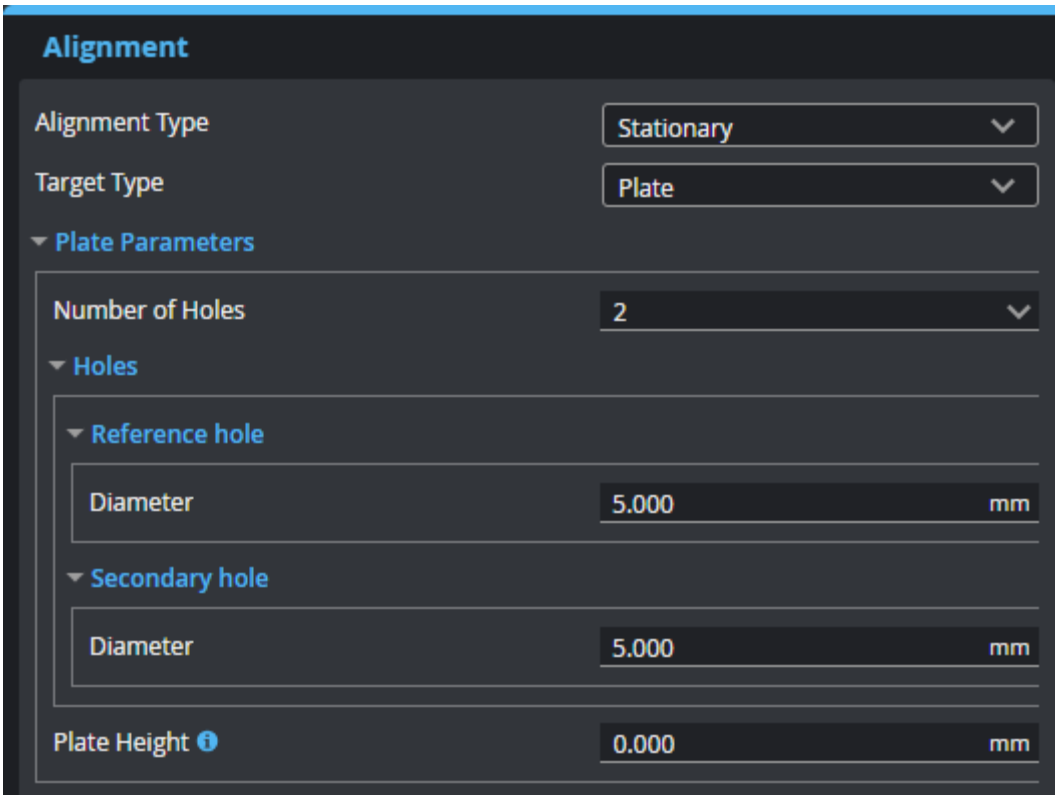
Make sure of the following when manufacturing an alignment plate:



- Holes must be as sharp as possible: avoid bevels.
- The size of the holes should be more than 10 times the XY resolution of the sensor; for the XY resolution of your sensor, see specifications of the sensor in *Sensors* on page 1332.
- Sensors must capture as little data from the inside of a hole as possible. Either countersink holes from the opposite side of the plate, or paint the insides of the holes with a flat black paint. Otherwise, although the alignment should succeed, it will not be as accurate: it may result in unwanted offsets or angles in the transformations.
- The recommended flatness of alignment targets for accurate Y angle is roughly the Z resolution rating of the sensor. If the bar target is curved, it will introduce an apparent Y angle in the sensor alignment. For sensor Z resolution, see the specifications for your sensor in *Sensors* on page 1332.
- It is not necessary to machine the plate height to a high tolerance. Height can instead be controlled during measuring rather than at manufacture. Only flatness is important. If the zero level is not critical for the measurement, then standard machining tolerances can be used. Alternatively you can machine to a low tolerance and measure the value to a high precision to save cost.
- Plates should be painted with flat light grey or white paint to improve data capture (by reducing the possibility of reflections and improving profile data of the bar surface).

CONFIGURING GOPXL FOR PLATE ALIGNMENT

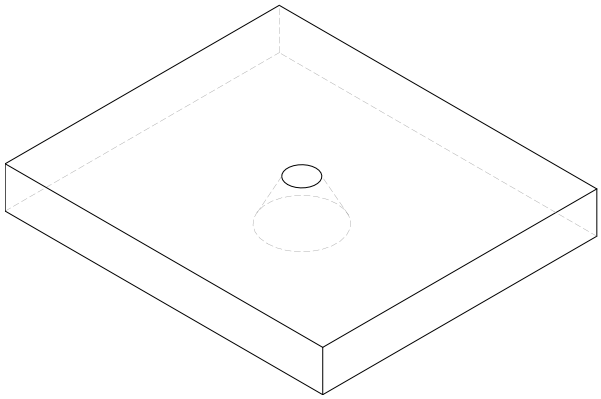
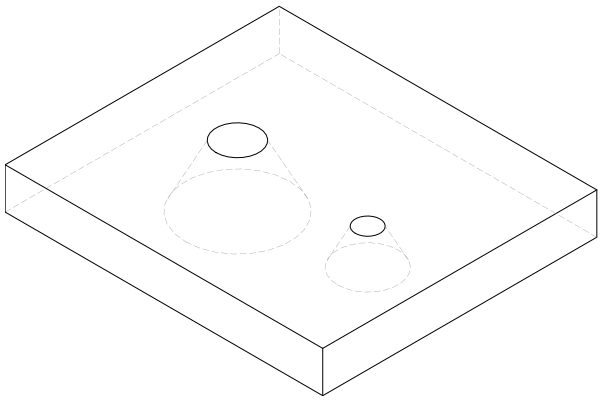
Configure the characteristics of the target; for more information on these settings, see below.



- **Number of Holes** is the number of holes in the alignment plate. For information on the transformations resulting from different types of plates, see *Transformations resulting from plates* below.
- **Diameter**: Sets the diameters of the holes. If you are using a plate with two holes, they must be of different sizes.
- **Plate Height**: The alignment procedure determines the average Z height of the alignment target's top surface and uses the value specified in **Height** to offset the coordinate system from that average Z height; in effect, the bottom of the alignment target becomes the Z origin (the zero reference level). This value can be negative.

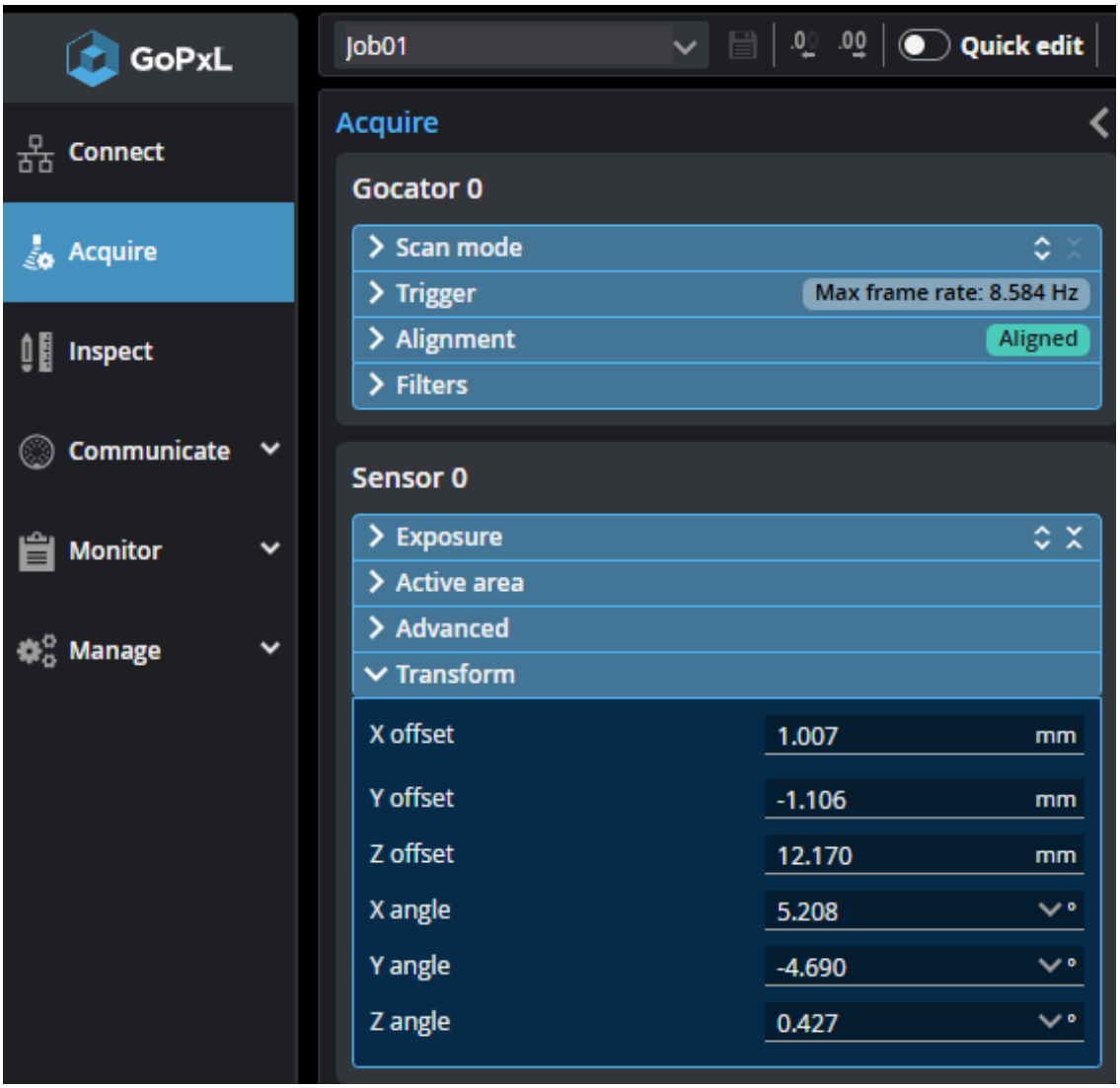
Transformations resulting from plates

Target	Resulting Transformations	Explanation
Plate	X offset Y offset Z offset X angle Y angle	The orientation of the surface surrounding the hole determines the X angle and Y angle transformations. Z angle is not calculated. X and Y offsets are calculated from the X and Y angles.
Plate (single reference	X offset	The center of the hole defines the location of the X, Y and Z origins (the

Target	Resulting Transformations	Explanation
hole)	Y offset Z offset X angle Y angle	offsets). The orientation of the surface surrounding the hole determines the X angle and Y angle transformations. Z angle is not calculated. 
Plate (with two reference holes)	X offset Y offset Z offset X angle Y angle Z angle	The sizes of the primary and the secondary holes must be different. The center of the primary hole defines the location of the X, Y, and Z origins (the offsets). The orientation of the surface determines X Angle and Y Angle. The angular position of the secondary hole on the XY plane, relative to the primary hole, determines Z Angle. 

Transformations

The transformation values determine how data is converted from unaligned coordinates to aligned coordinates (for an overview on coordinate systems, see *Coordinate Systems* on page 51). Transformation values are set automatically when you align a sensor using an alignment routine on the **Acquire** page. The resulting transformations are displayed in the expandable **Transform** section on that page.



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.
X Angle	Specifies the tilt around the X axis.
Y angle	Specifies the tilt around the Y axis.
Z angle	Specifies the tilt around the Z axis.

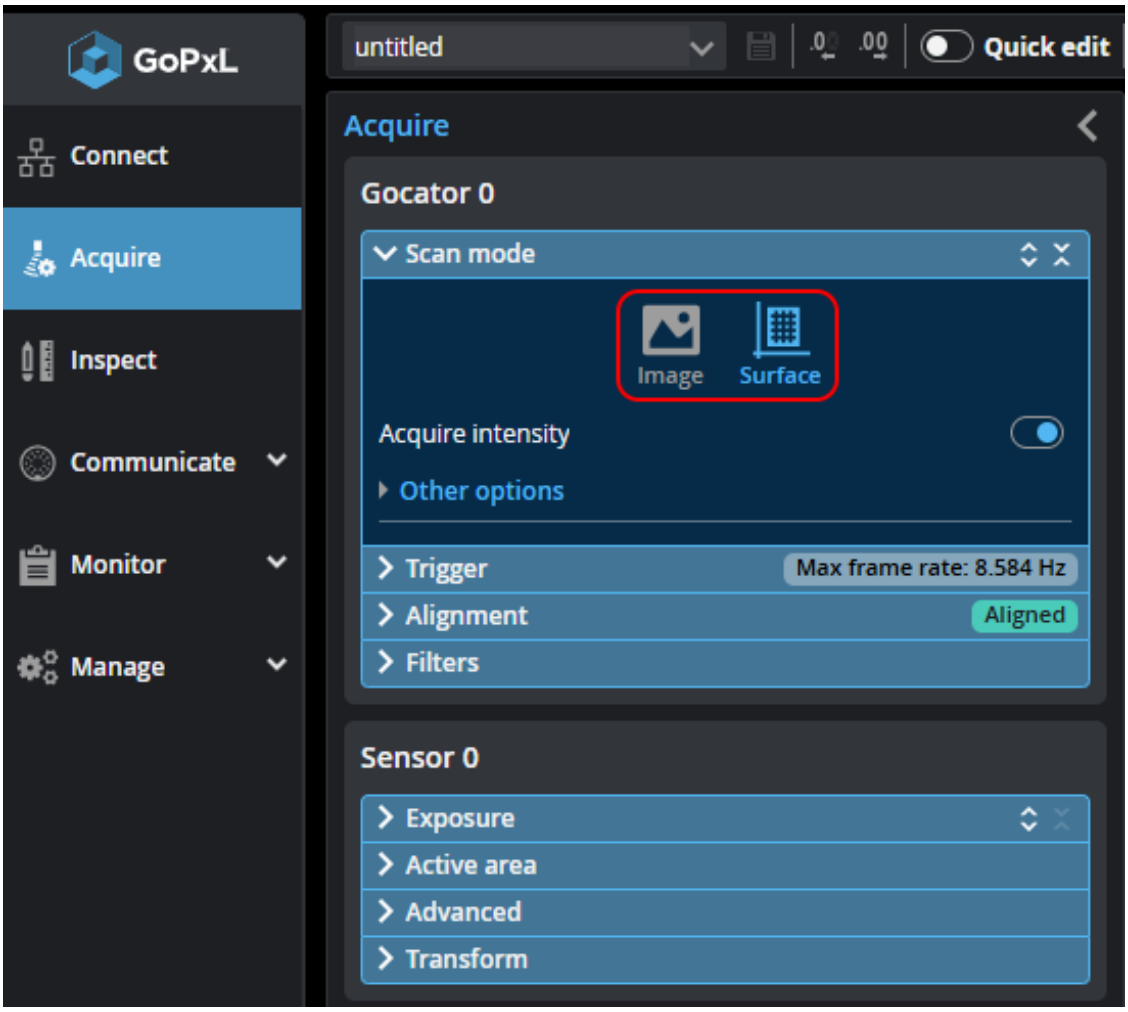
When applying the transformations, the data is first rotated around X (clockwise, with the X axis toward the viewer), then Y (counterclockwise), and then Z (clockwise), and then the offsets are applied.

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). For more information on trigger spacing, see *Trigger Settings* on page 143.

Scan Modes and Intensity

Sensors supports one or more data acquisition modes, in addition to an image mode. You choose the scan mode in the **Acquire** page > **Scan Mode** section.



If you have previously recorded scan data and have configured the input of measurement tools, changing the scan mode will display an error, saying the tool inputs are no longer available. For example, if you have scanned and recorded some Surface data, and added and configured a Surface Position measurement tool, but then switch to the Image scan mode to diagnose exposure issues, GoPxL warns you that the measurement tool's input (its data source) is missing.

Scan Mode	Description
Image	Outputs video images from the sensor. This mode is useful for configuring exposure time and troubleshooting stray light or ambient light problems.
Surface	Outputs Surface data.
Acquire Intensity	When this option is enabled, an intensity value is produced for each data point. For more information on intensity, see *Intensity Output* on page 102.
Other options	The **Other options** section contains spacing related settings. When the **Enable uniform spacing** option is enabled, data points are resampled to a uniform spacing . Set the size of the spacing with the Uniform spacing interval settings; for more information, see *Spacing on X and Y (Uniform Spacing)* below. Disable this option to get data from the sensor at the highest possible rate.

Spacing on X and Y (Uniform Spacing)

Options for spacing on the X and Y axis are available in the **Scan mode** panel, in the expandable **Other options** section.

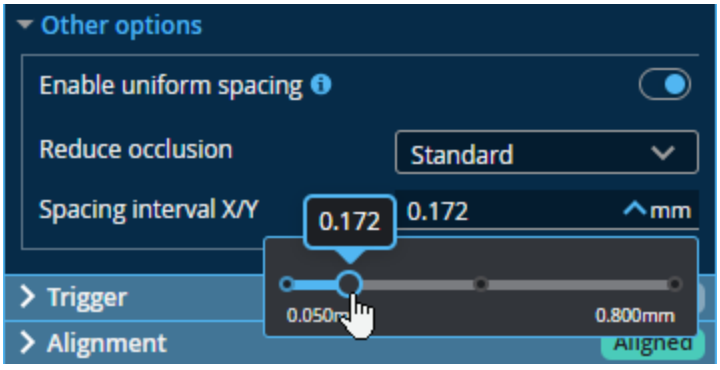
When **Enable uniform spacing** is enabled,data points (the height values) are output using uniform spacing along X and Y. Set the size of the spacing using the spacing interval settings (see below).

When uniform spacing is disabled, the sensor outputs unprocessed data, that is, point cloud data, or non-uniform data: the spacing between data points varies. The sensor reports data points in (x, y, z) coordinate tuples.

Disable uniform spacing to get data from the sensor at the highest possible rate.

The spacing interval is the user-configured spacing between data points in uniform 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.

You can set the spacing interval in the **Spacing interval X/Y** parameter.



Reduce Occlusion

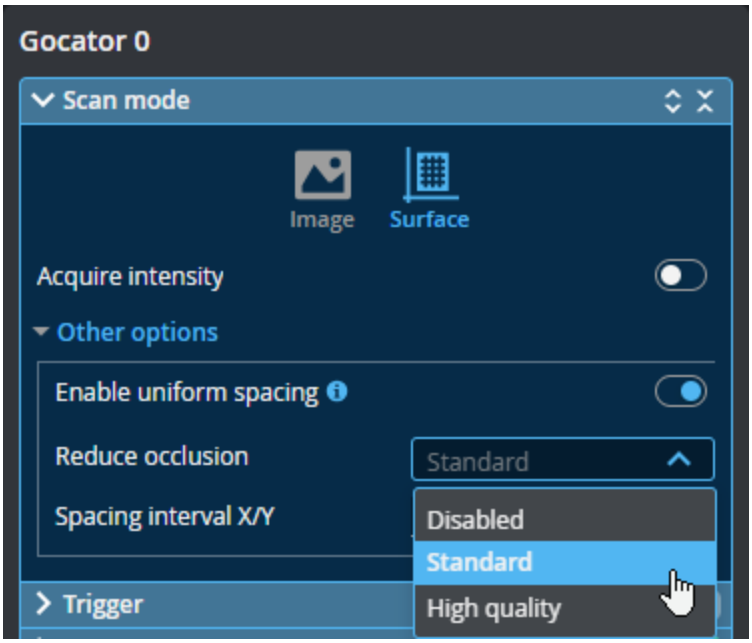
When this option is enabled (by default, set to Standard), in addition to stereo correlation, each camera independently triangulates off the LED light pattern, which may improve performance on targets with complicated shapes that can cause occlusions. The following options are available

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.

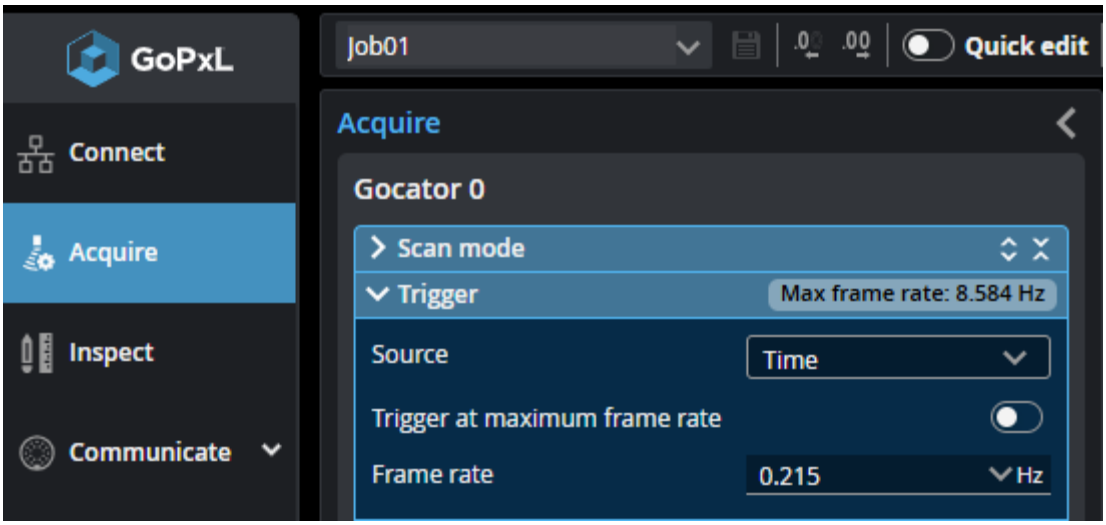
When the **Reduce Occlusion** option is disabled, 3D data is acquired using *only* stereo correlation between the two cameras, meaning that a point on the target must be visible in both cameras to produce a data point.

For more information, see *Stereo Correlation vs. Independent Triangulation* on page 50.



Triggers

A trigger is an event that causes a device to acquire a single frame or image. You configure triggers on the **Acquire** > page, in the **Trigger** panel.

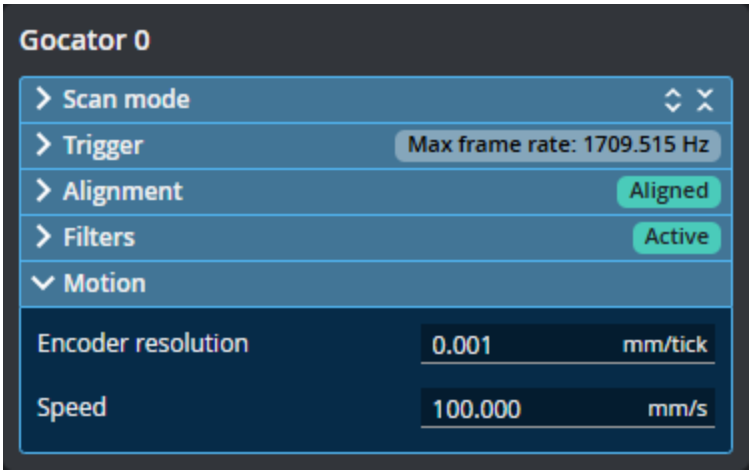


When a trigger is processed, the LED light pattern is strobed and the cameras expose to produce images. The resulting images are processed inside the sensor to yield a 3D point cloud. The data can then be used for measurement.

The sensor can be triggered by one of the sources described 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 1353.
>
> 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 1350 and *Digital Input* on page 1350, respectively.

To set the encoder resolution, expand the **Motion** section and set the value in **Encoder resolution**.



Trigger source descriptions

Trigger Source	Description
Time	Devices trigger frame acquisition at the configured frame rate. If tool processing has not completed at the time of a trigger, that frame will be dropped.
External Input	A digital input can provide triggers in response to external events (for example, a 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. Use the **Trigger Drops** indicator in the **Health** panel on the **Reports** page to check for this condition; for more information, see *Monitor - Measurement Stats, Health & Performance* on page 1167. For information on the maximum input trigger rate, see *Maximum Input Trigger Rate* on the next page.
Software	A network command can be used to send a software trigger. For more information, *Protocols (PLCs and other hardware)* on page 1257.

Depending on the setup and measurement tools used, the CPU utilization may exceed 100%, which reduces the overall acquisition speed.

For examples of typical real-world scenarios, see *Trigger Examples* on the next page.

Trigger Settings

After you choose the trigger source, configure the other parameters in **Acquire** page > **Trigger** panel.

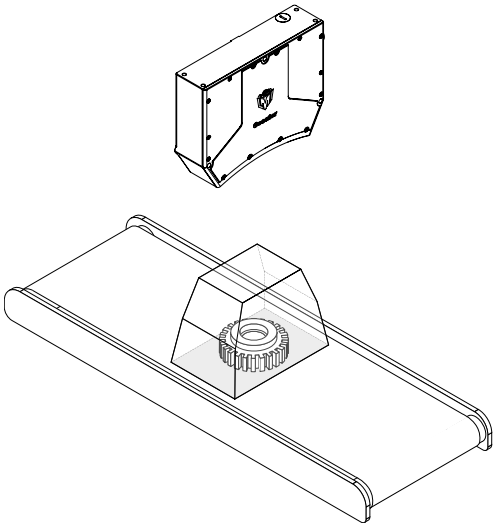
Parameter	Trigger Source	Description
Source	n/a	The source for triggering the acquisition of frames (Sequential, Time, External Input, or Software).
Trigger at maximum frame rate	Time	When this is enabled, frames are acquired at the maximum possible frame rate. The maximum frame rate is limited by the exposure, active area, and light duty cycle settings.
Frame Rate	Time	Controls the frame rate. Fractional values are supported. For example, 0.1 can be entered to run at 1 frame every 10 seconds.
Trigger Delay	External Input	Specifies 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 (for example, a photocell) and the sensor.

Trigger Examples

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

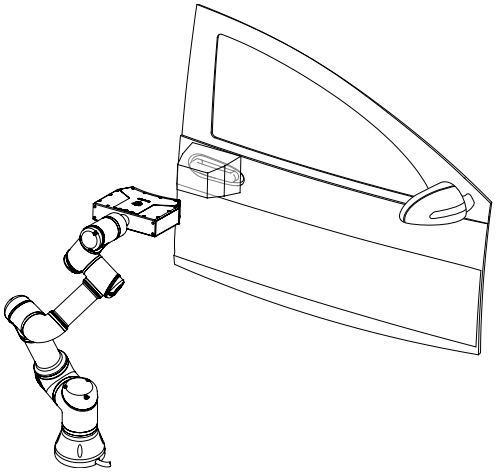
For example, a photocell can be connected as an external input to generate a trigger pulse when a target object has moved into position.

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.



Maximum Input Trigger Rate

The maximum external input trigger rate in a system including Master 810 or 2410 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

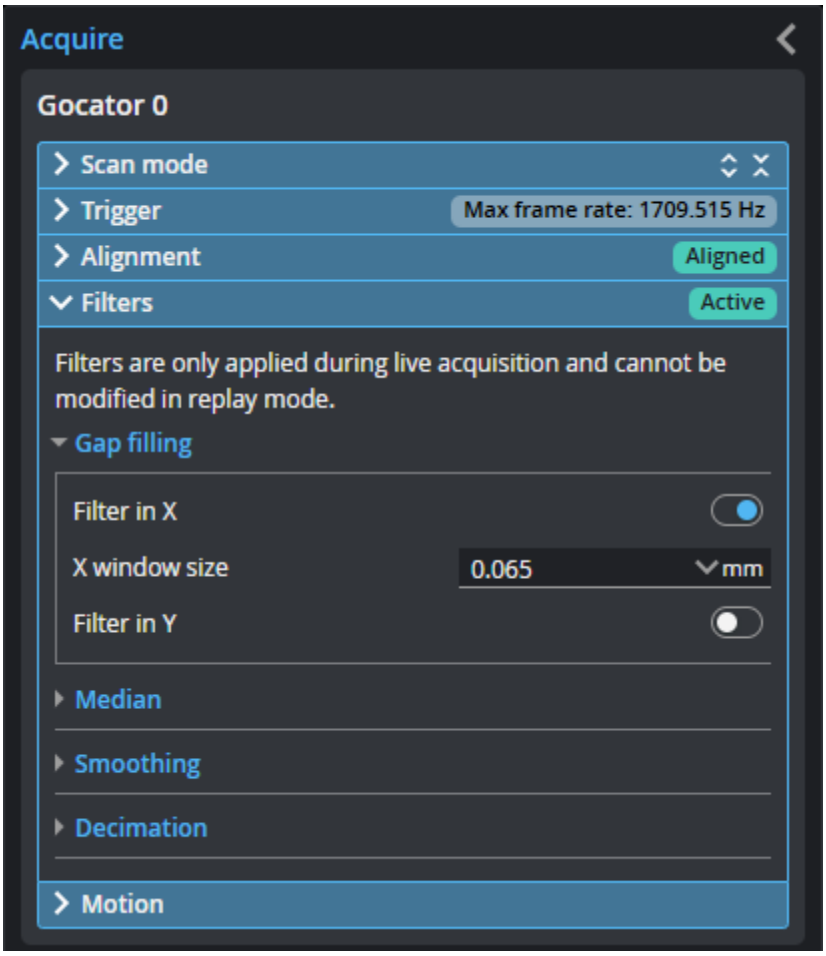
On a standalone sensor, with the encoder directly wired into the I/O port or through a Master 100, the maximum encoder rate is about 1 MHz.

For sensors connected through a Master 810 or higher, with the encoder signal supplied to the Master, the maximum rate is about 300 kHz.

Filters

The filters in the **Filters** panel are used to process scan data along the X or Y axis to remove noise or clean it up *before* it is used by measurement tools or is output. Using the filters can help you get more repeatable measurements. The filters are only available when uniform spacing is enabled. They are only applied during data acquisition; modifying these filters when GoPxL is in replay mode has no effect.

When any filter is enabled, an "Active" indicator is displayed at the top of the **Filters** panel. Otherwise, no indicator is displayed.



The following filters are available in the Filter panel.

Filters

Name	Description
Gap Filling	Fills in missing data using information from the nearest neighbor data points, for example, when data is missing due to occlusions. Gap filling also fills gaps where no data is detected, which can be due to low surface reflectivity, for example dark or specular surface areas, or to actual gaps in the surface. The values in **Filter in X** and **Filter in Y** represent the maximum gaps the sensor will fill. Wider gaps are not filled. 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 the specified window. The sensor can fill gaps along both the X axis and the Y axis. In Profile mode, gap filling is limited to the X axis.
Median	Substitutes the value of a data point with the median calculated within the window or windows set in **X Window Size** or **Y Window Size** around the data point. If the number of valid (non-null) data points in the window is even, the average of the

Name	Description
	two values in the center is used. If the number of valid points is odd, the value in the center of the sorted list of values is used. Missing data points will not be filled with the median value calculated from data points in the neighborhood. With an odd-numbered window size, the output is at the center of the window. With an even window size, the output is 0.5 pixels to the right of the center (that is, using window / 2-1 values from the left, and window / 2 from the right.
Smoothing	Substitutes a data point value with the mean value of that data point and its nearest neighbors within the window or windows set in **X Window Size** or **Y Window Size**. 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 neighborhood.
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 Window Size** to 0.2, only points every 0.2 millimeters will be used. The filter generates points starting from the leftmost edge of the scan data, stepping in equal steps away from that side.

Tool-based filtering is also available on the **Tools** page. The filters described here and in the filter tool topics use the same algorithms.

Using tool-based filtering provides various advantages:

- Additional filters not available in the **Filters** panel. (This mostly applies to Surface filters.)
- Choosing which tools used in a job take filtered data as input. That is, you can decide to have some tools running on unfiltered data and other tools on filtered data.

For more information on tool-based filters, see *Surface Filter* on page 862 (Surface-based) and *Profile Filter* on page 564 (Profile-based).

The filter window sizes are specified in millimeters (and additionally, in the tools, in data points). To calculate the number of data points that a window covers when the units are millimeters, use the following calculation:

- User-specified window size divided by the X spacing interval (that is, the number of millimeters per point) on the **Spacing** tab in the **Sensor** panel. (For more information on spacing intervals, see *Spacing on X and Y (Uniform Spacing)* on page 140.)
- *With the exception of the gap filling filter*, round the result of the division to the nearest integer value. With the gap filling filter, filling is performed within the provided window size.

For example, if you set the size of the filter's window to a value between 1.5 mm and 2.49 mm (inclusively), and the X spacing interval is set to 1 mm, the filter covers 2 data points. A filter window size from 2.5 mm to 3.49 mm results in a filter covering 3 data points.

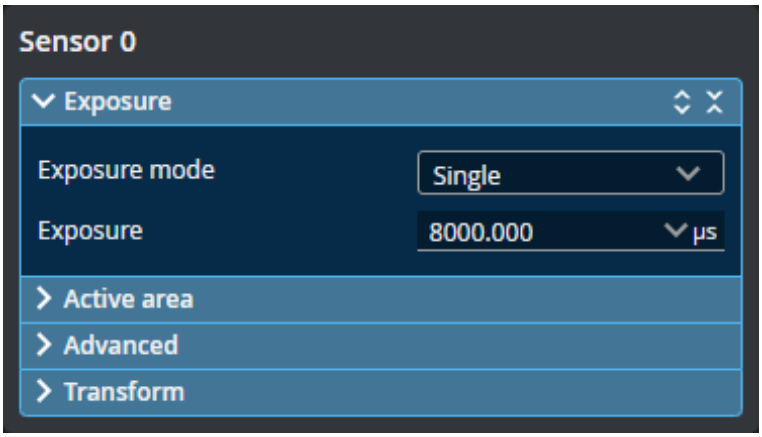
Exposure

Exposure determines the duration of camera and light-source on-time (or pulse width), given in microseconds (µs). Longer exposures can help detect light on dark or distant surfaces, but increasing exposure time decreases the maximum frame rate. Different target surfaces may require different exposures for optimal results. In general, shiny and mirror-like surfaces require a short pulse width and low-gloss surfaces a high pulse width.

Sensors provide two or three exposure modes, depending on the model, for the flexibility needed to scan different types of target surfaces.

To properly set exposure, you can acquire scan data in Image mode, and examine the resulting image to confirm whether the features or flaws you need to measure will be well presented in the scan data. For more information on Image mode and the information it provides for configuring exposure, see *Image Mode* on page 85.

You configure exposure in **Exposure** on the **Acquire** page, in the **Sensor Properties** section.



If you are configuring the exposure of a sensor in a multi-sensor system, select the tab of the correct sensor.

When exposure is set to Dynamic or Multiple, additional parameters are displayed.

Exposure Mode	Description
Single	Uses a single exposure duration for each frame. Use this when the surface is roughly uniform and is the same for all targets. For more information, see *Single Exposure* on page 151.
Multiple	Uses multiple exposures to create a single scan. Used when the target surface has a varying reflectance within a single scan. For more information, see *Multiple Exposure* on page 152.

When the sensor is set to Image mode (see *Scan Modes and Intensity* on page 139), a **Video pattern type** parameter is available. (When this parameter is set to **Scan Sequence**, a **Video pattern Index** parameter is also available.)

Pattern Settings

Parameter	Description
Pattern	One of the following: **Default**: The default projector pattern. **Scan Sequence**: Enables the Pattern Index parameter (see below). **Projector Off**: Turns the projector off during acquisition for customers who want to capture video images using external lighting. **Focus**: A special pattern available on Gocator 3506 sensors used to help focus the sensor. For more information, see *Using the Focus Pattern (Gocator 3506 only)* below.
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 (Gocator 3506 only)

On Gocator 3506 sensors, 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.

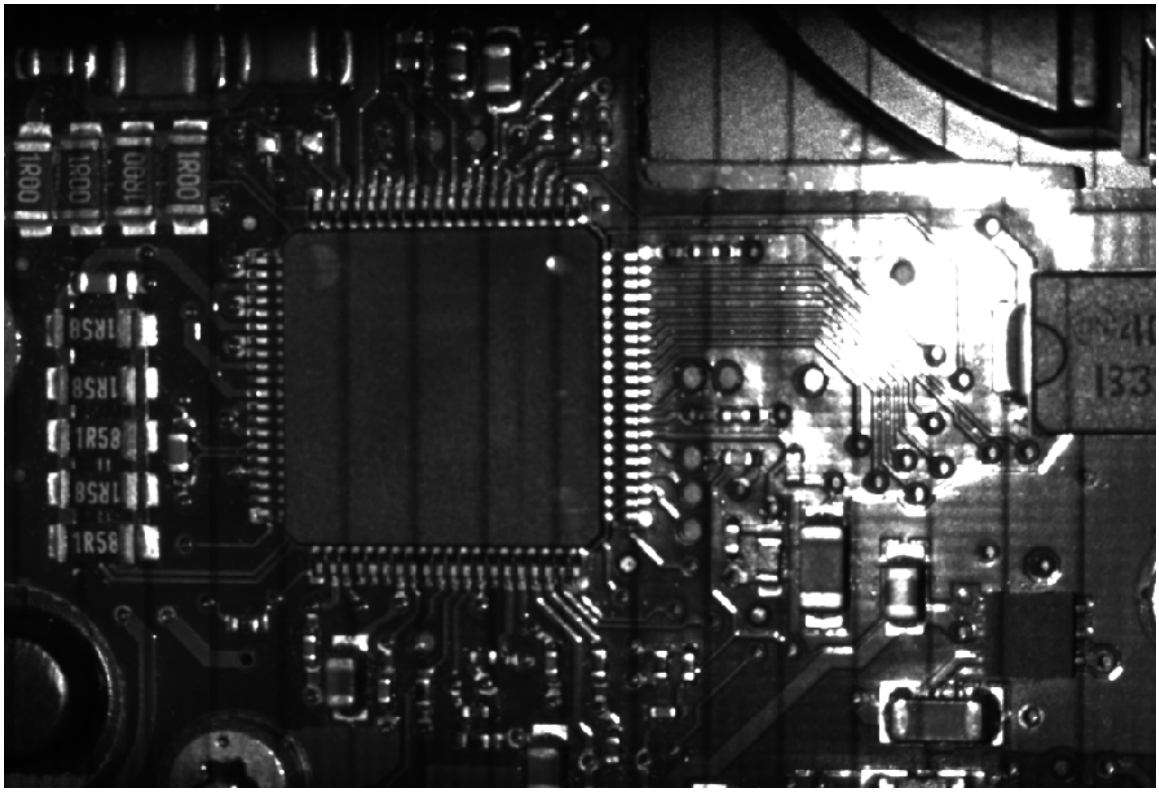
To use the focus pattern:

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

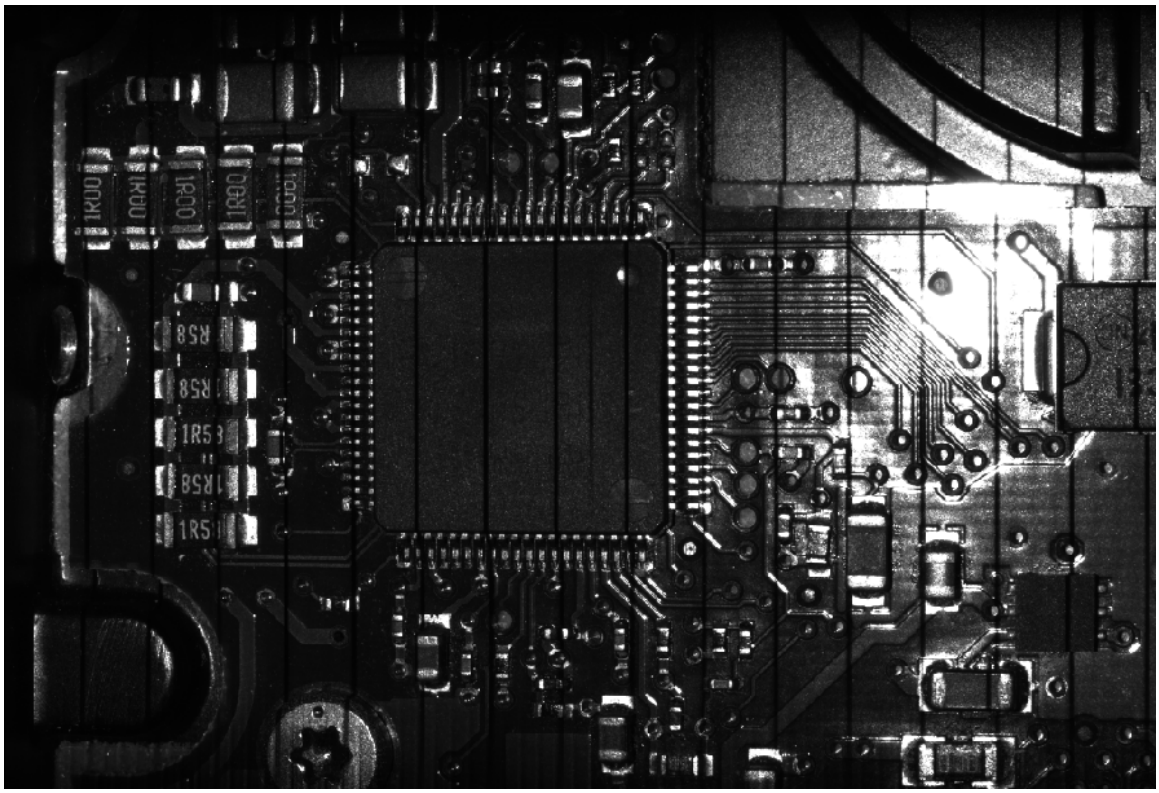
   The target surface should be similar to the material that will normally be measured.
2. Go to the **Scan** page and choose Video mode.
3. Expand the **Sensor** panel by clicking on the panel header or the ⊕ button.
4. Click the **Exposure** tab.
5. In **Pattern**, choose **Focus**.

   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:



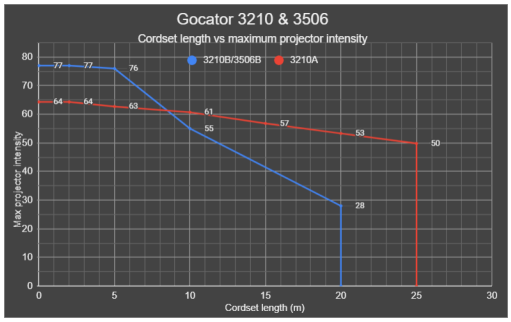
After focusing, the lines are more sharply defined:



Gocator 3210A or B, 3506B on 24 VDC

Gocator 3210 (A and B versions) and Gocator 3506 (A and B versions) can run at 24 VDC instead of the standard 48 VDC when required, for example due to regulations. However, with the exception of Gocator 3506A, lowering the operating voltage of these sensors to 24 VDC causes an increased current flow at constant power draw. The increased current will trip the sensor overcurrent protection at 1.1 amps, which will occur at maximum projector intensity, when the power draw is the highest. Note that longer cordsets introduce additional resistance, leading to increased current, again at constant power draw.

To operate at 24 VDC without tripping overcurrent protection, the sensor lowers the maximum projector intensity to limit the current flow. The following chart plots maximum projector intensity versus cordset length. Note that version B Gocator 3210 and 3506 sensors use a more powerful projector and therefore have different 24 VDC maximum intensity settings.



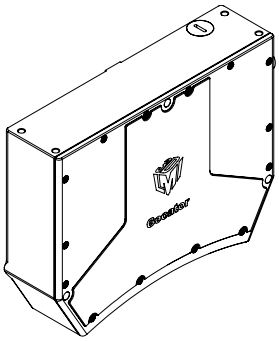
Note also that 3210B and 3506B have a maximum cordset length of 20 meters, whereas 3210A has a maximum of 25 meters. The 3506A is absent from the graph because its maximum power draw does not trip the sensor at 24 VDC.

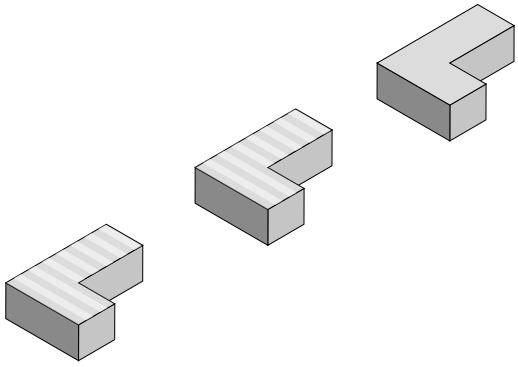
When using 3210B or 3506B in A compatibility mode, the intensity values are similar to those of the A versions (although B versions are generally brighter), except for the maximum cordset length, which for B versions is limited to 20 meters instead of 25 meters.

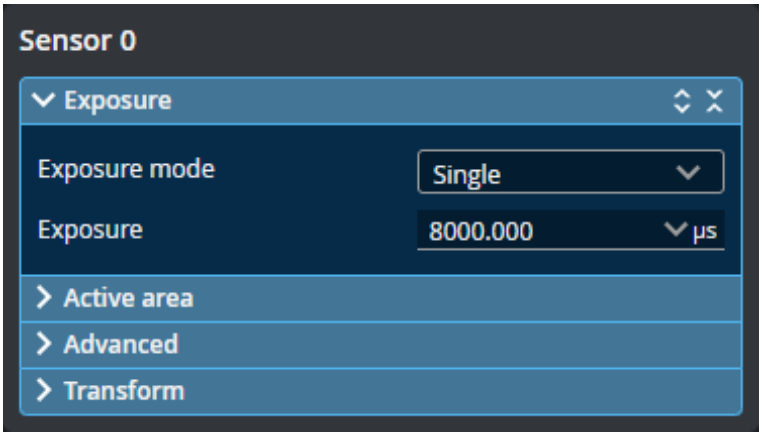
You must consider the impact of the limitations of using 24 VDC on a case-by-case basis, but in general, applications will require longer exposure times. LMI recommends operating at 48 VDC if at all possible, or limiting cordset length. For longer exposure times, the sensor requires new exposure settings 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.







Setting	Description
Exposure Mode	Set to Single.
Exposure	The length of the light emitter's on-time, in microseconds.

Multiple Exposure

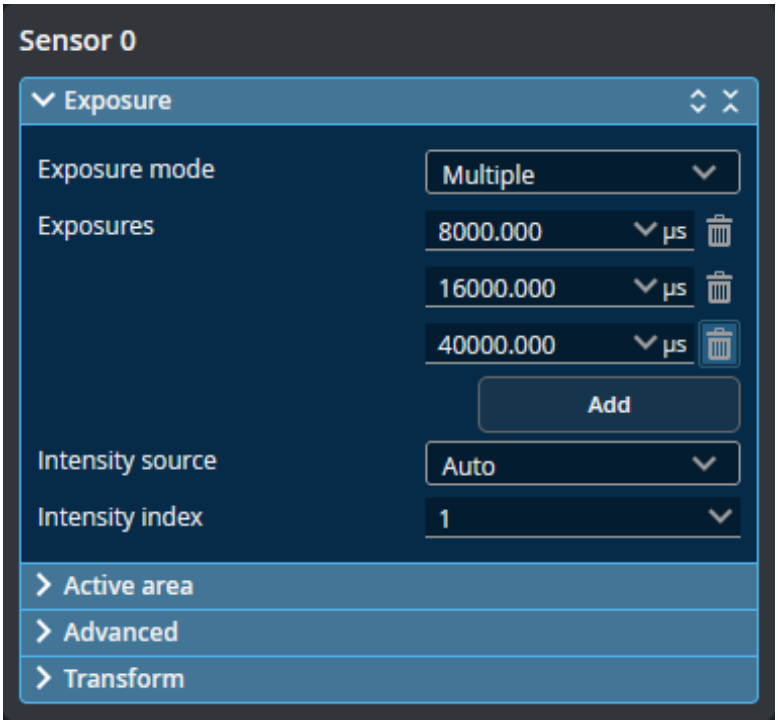
The sensor can combine data from multiple exposures to create a single profile (G2, G4, and G5) or 3D scan (all families). Multiple exposures can be used to increase the ability to detect light and dark materials that are in the field of view simultaneously. Note that you can optionally output the scan

data for the different exposures as separate profiles or surfaces, but only if uniform spacing is disabled.

You can define up to five exposures (three for G3 sensors), with each set to a different exposure level. You add exposures using the **Add** button, and remove them using the trashcan icon. The order of the exposures does not matter.

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 scan data is a composite created by combing data collected with different exposures.



If you have enabled intensity in the **Scan Mode** panel, you can use **Intensity** (**Intensity index** on some models) to choose which exposure the sensor uses for acquiring intensity data. This lets you choose the exposure that produces the best image for intensity data. If intensity is disabled, these settings are hidden.

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

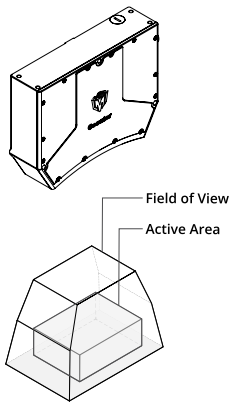
Setting	Description
Exposure Mode	Set to Multiple.
Intensity Source	The camera used for the intensity data. One of the following: Auto (combines intensity data from both cameras), Left, or Right. Choose the option that provides the best data.

Active Area

Active area refers to the region within the sensor's maximum scan volume that is used for data acquisition.

By default, the active area covers the sensor's entire scan area. By reducing the active area, the sensor can operate at higher frame rates. You can also reduce the active area to exclude areas that are affected by ambient light, or to avoid supports or parts of a transport system (such as a conveyor).

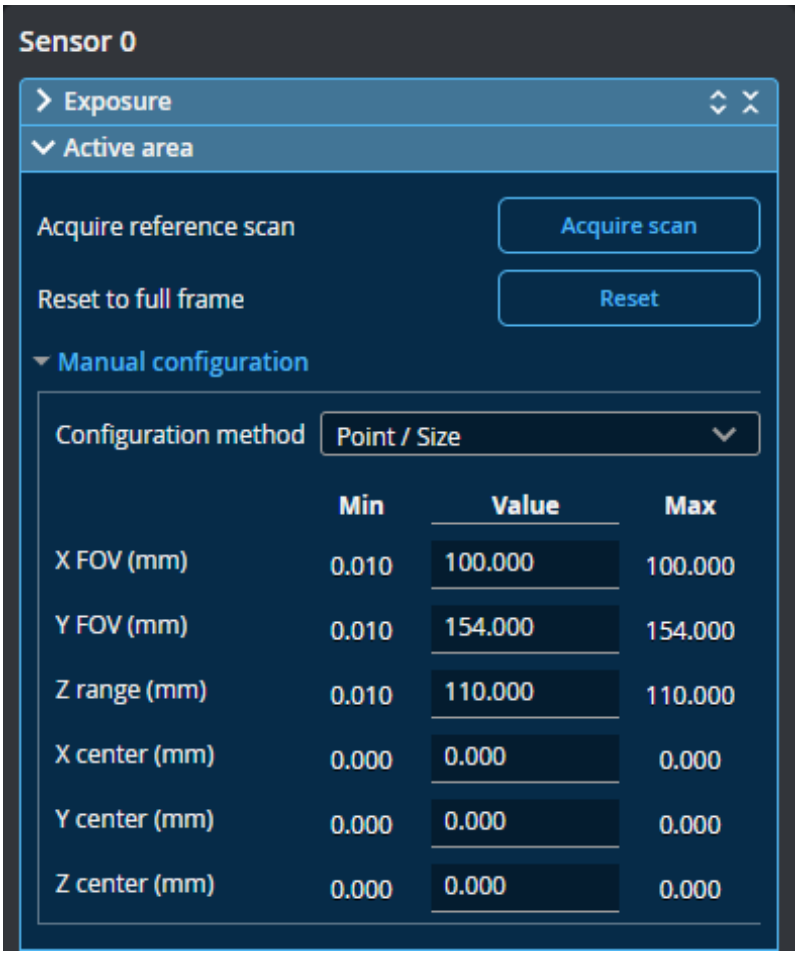
Active area is specified in sensor coordinates, rather than in system coordinates. As a result, if the sensor is already alignment calibrated, press the **Acquire** button to display uncalibrated data before configuring the active area. See *Coordinate Systems* on page 51 for more information on sensor and system coordinates.



G3 sensor capturing data within the active area. In an unaligned sensor, the X, Y, and Z origins are in the centre of the field of view.

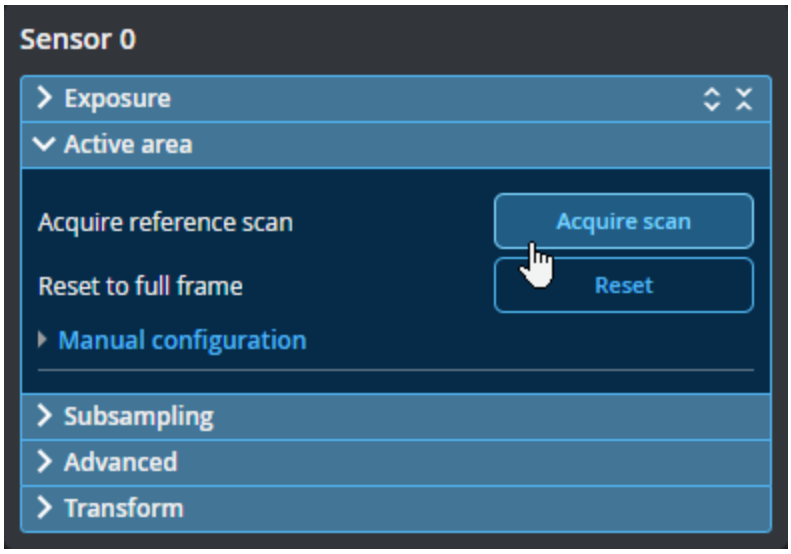
You set the active area in the **Acquire** page, in the **Active area** panel. You can configure the active area with your mouse by adjusting the position and size of the active area in the data viewer. You can also manually configure the active area under **Manual configuration** for precise control.

Two active area configuration schemes are available: Maximum / Minimum and Point / Size (for the parameters related to each scheme, see the tables below). For instructions on setting active area, see *To set a sensor's active area* on the next page.

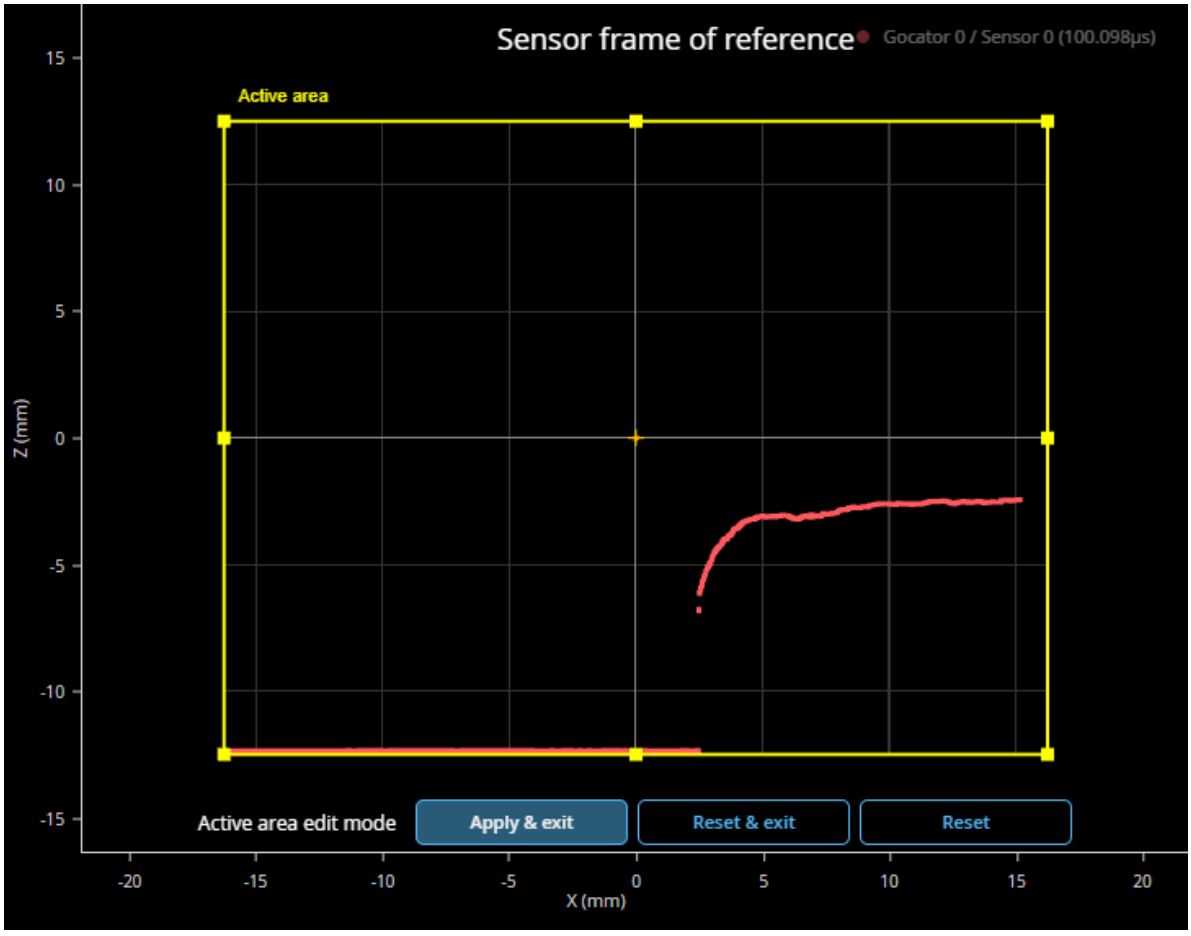


To set a sensor's active area

1. Go to the **Acquire** page.
2. If needed, expand the **Active area** section.
3. (Optional) Place a sample target in the sensor's scan area, and in the **Active Area** section, click **Acquire scan** to get some scan data.



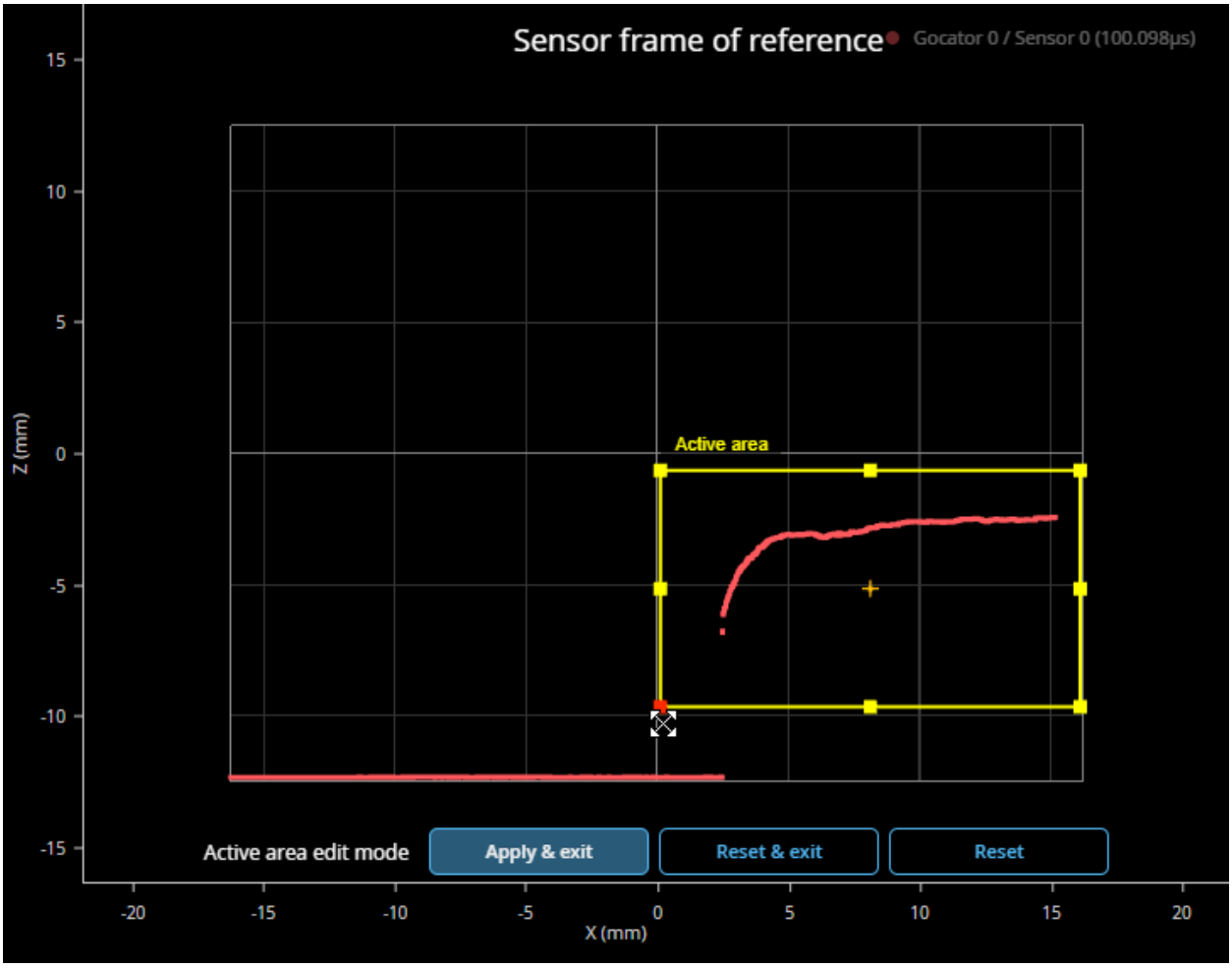
Getting a scan while setting the active area helps you determine the necessary size and position of the active area.



4. Set the active area by doing one of the following:

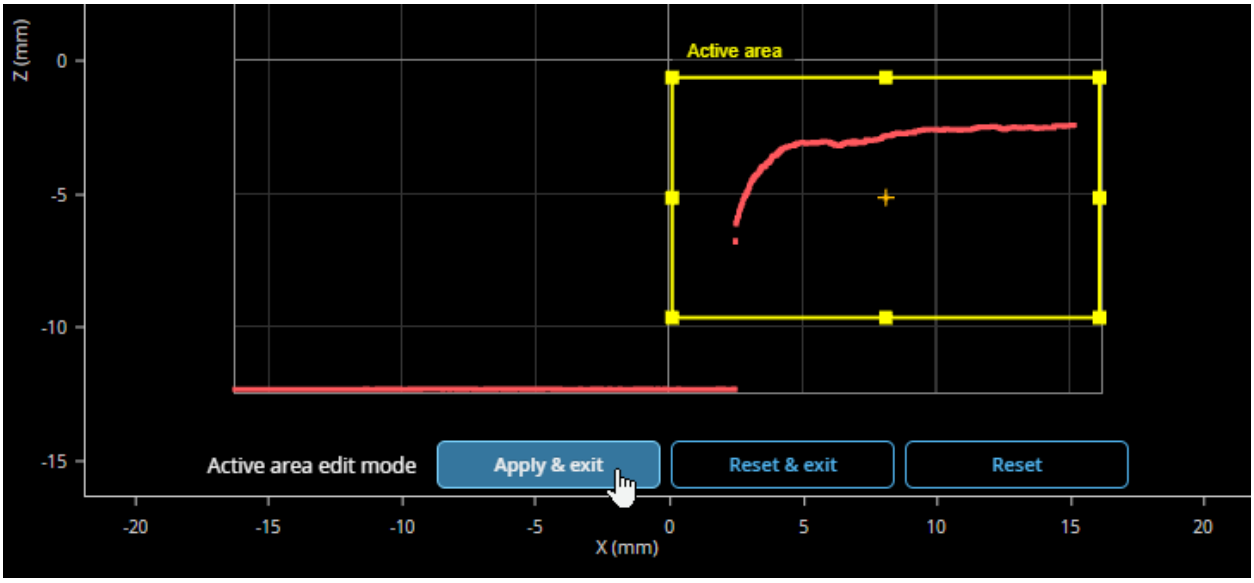
Set active area graphically

a. Adjust the active area size and position using your mouse in the data viewer.



b. Do one of the following:

If you are satisfied with the position and size of the active area, click **Apply & exit**



The data viewer exits active area edit mode. You can examine the position and size of the region by expanding **Manual configuration**.

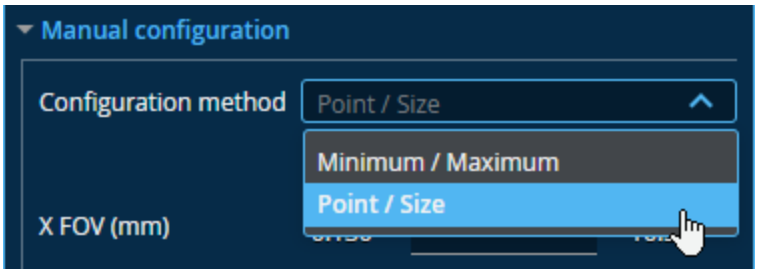
If you need to start over, click **Reset**. This resets the active area to its full size but lets you continue modifying it.

If you no longer need to configure the active area, click **Reset & exit**.

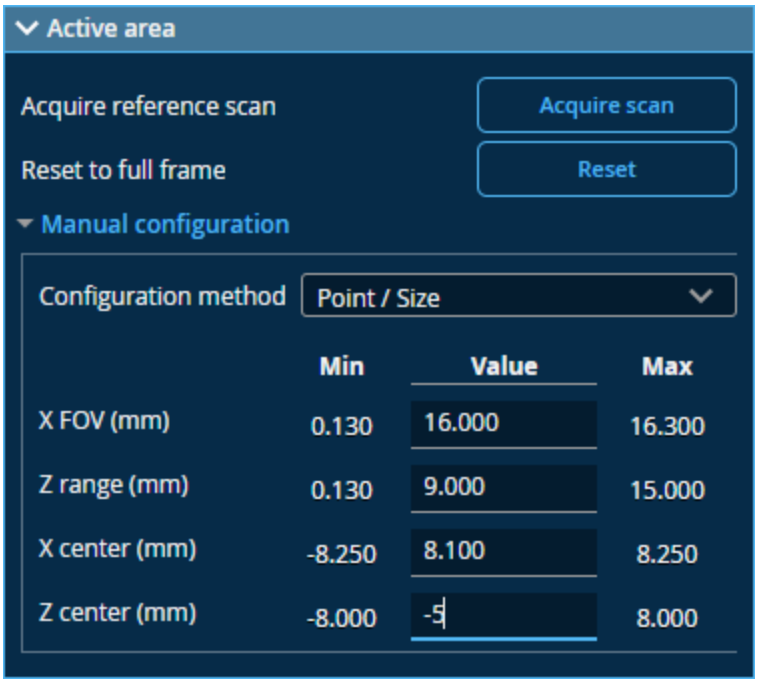
Set active area manually

a. Expand the **Manual configuration** section.

b. Choose the active area configuration method in **Configuration Method**.



c. Enter the values manually in the fields.



Point / Size: For parameter descriptions, see *Active area parameters - Point / Size method* on the next page.

Minimum / Maximum: For parameter descriptions, see *Active area parameters - Maximum / Minimum method* on the next page.

d. To set the values, click outside the field you are editing.

5. Save the changes in the job file.



To reset the active area

- Click the **Reset** button.

Active area parameters - Point / Size method

Name	Description
X FOV (mm) Y FOV (mm)	The width of the active area along the X and Y axis. This sets the effective field of view of the sensor.
Z range (mm)	The height of the active area along the Z axis. This sets the effective measurement range of the sensor.
X center (mm) Y center (mm) Z center (mm)	The X, Y, and Z position in the entire sensor field of view of the *center* of the active area.

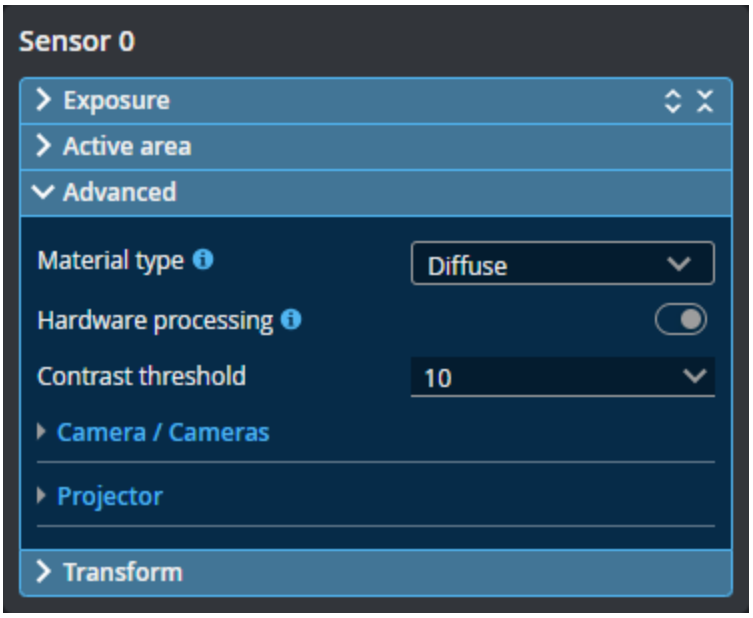
Active area parameters - Maximum / Minimum method

Name	Description
Min X Max X	The minimum and maximum X values.
Min Y Max Y	The minimum and maximum Y values.
Min Z Max Z	The minimum and maximum Z values.

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.

Advanced Settings

The settings in **Acquire** page > **Sensor** > **Advanced** section let you configure camera gain, material characteristics, and projector.



Material Type

You can configure data acquisition 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 **Material type** setting in the **Advanced** section on the **Acquire** page. 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.

	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://lmi3d.com/product-downloads/). 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 *Software Upgrade* on page 1184.

Camera Gain

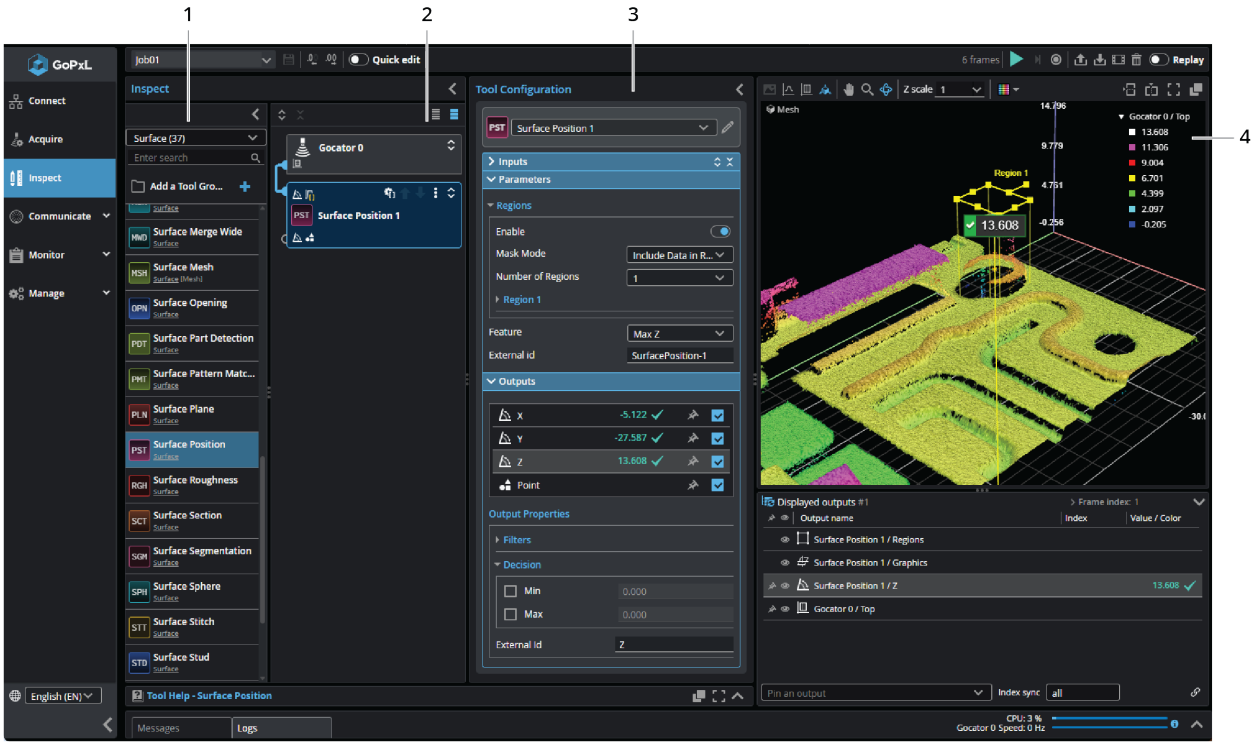
Setting	Description
Camera Gain	
	Digital camera gain can be used when the application is severely exposure limited, yet dynamic range is not a critical factor.

Other Advanced Settings

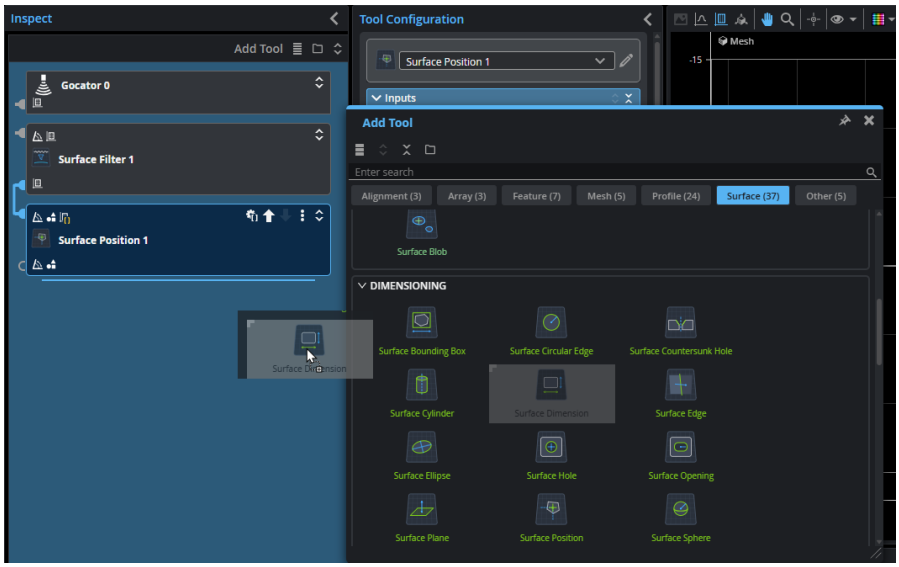
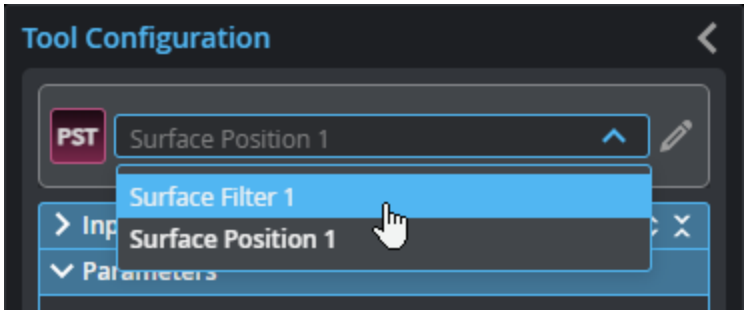
Setting	Description
Hardware decoding	Not settable.
Contrast Threshold	Controls the contrast threshold. Allows tuning the detection of points based on the intensities observed in Image mode.
Projector	Contains projector settings. **Intensity** Current projector intensity as a percentage. **Pattern uniformity** Pattern uniformity coefficient. Balances the center and sides of the projected image by lowering center intensity. Default = 1.0 no effect Min = 0.01 minimal center intensity Max = 1.0 full center intensity

Inspect - Measurement and Processing Tools

You add and configure measurement and processing tools on the **Inspect** page. Add tools using the **Add Tool** window that appears when you click the **Add Tool** button. You organize tools in your tool chain in the Tool Diagram (1). You configure tools in the Tool Configuration pane (2).



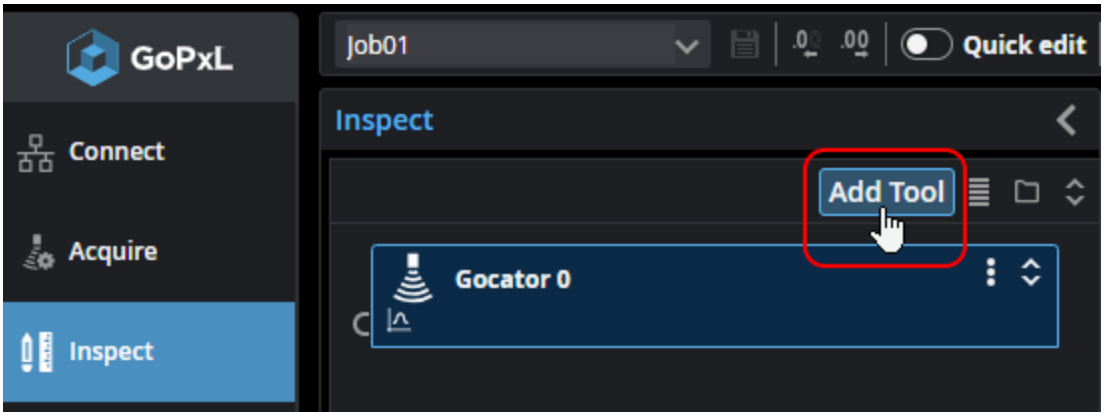
	Element	Description
1	Tool Diagram	Lists the tools that you have added. To add a tool, click **Add Tool**, and then double-click it in the **Add Tool** window or drag and drop to the Tool Diagram. For more information, see *Adding a Tool* on page 164.

	Element	Description
		 The Tool Diagram lets you better visualize the workflow in complex tools chains. For more information, see *Working with Tool Chains* on page 172.
2	Tool configuration	You configure a tool using the **Inputs**, **Parameters**, and **Outputs** expandable sections in the **Tool Configuration** panel. Typically, you configure tools from top to bottom, that is, by setting and configuring the inputs first, then the parameters, and finally the outputs, all in the order they are presented in these sections. You can select which tool you want to configure from the tool selector drop-down, or by selecting the tool in the Tool Diagram to the left.  For information on parameters common to most tools, see *Tool Configuration* on page 196. For information on the different types of tools, see their topics in the following sections. Surface tools: See *Surface Tools* on page 712. Feature tools: See *Feature Tools* on page 252. Mesh tools: See *Mesh Tools* on page 493.

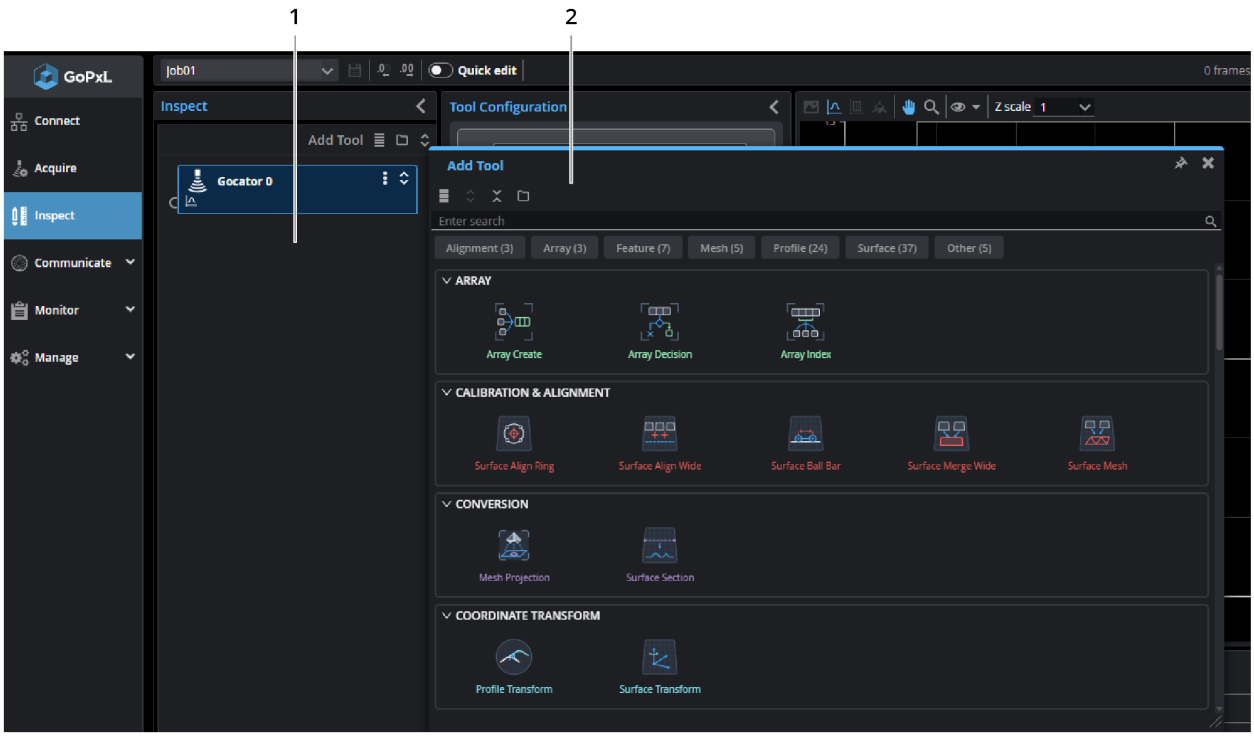
	Element	Description
		Array tools: See *Array Tools* on page 233. Measurement Formula tool: See *Measurement Formula Tool* on page 489. Script tool: See *Script Tool* on page 670. Data Export tool: *Data Export Tool* on page 247.
3	Data viewer	When configuring tools, you can graphically configure regions of interest. The data viewer also displays currently selected and pinned measurement values. The data viewer's toolbar lets you change the display mode (for more information, see *Data Viewer* on page 79) and split a data viewer or pop out a data viewer in a new browser window.

Adding a Tool

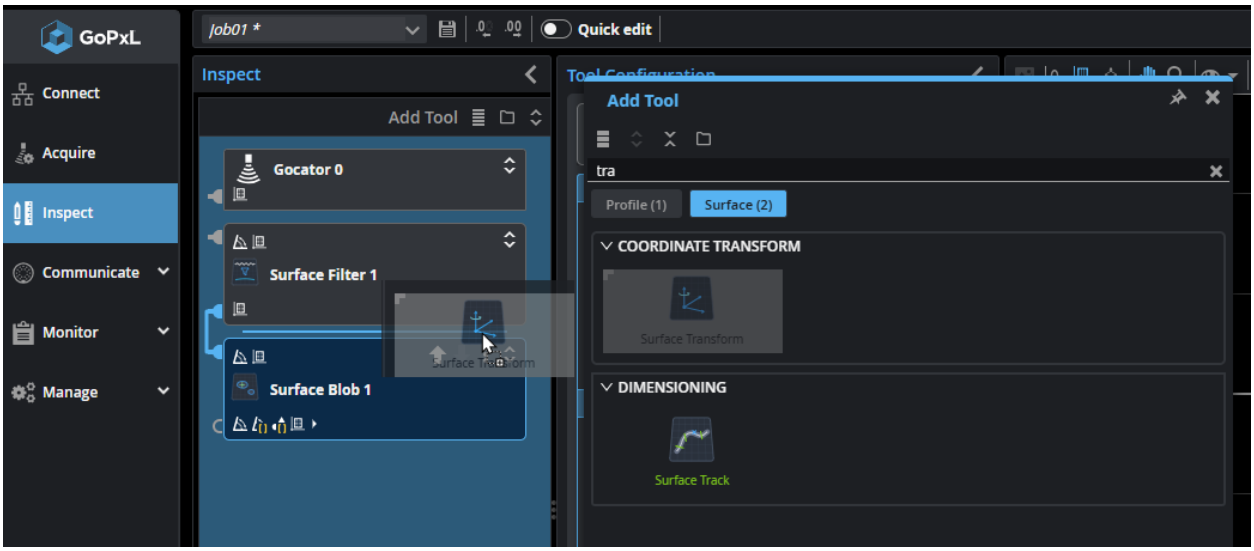
You add a tool to a job using the **Add Tool** window. To open the tool picker window, click **Add Tool** in the Tool Diagram.



The tool picker opens and displays the available tools.



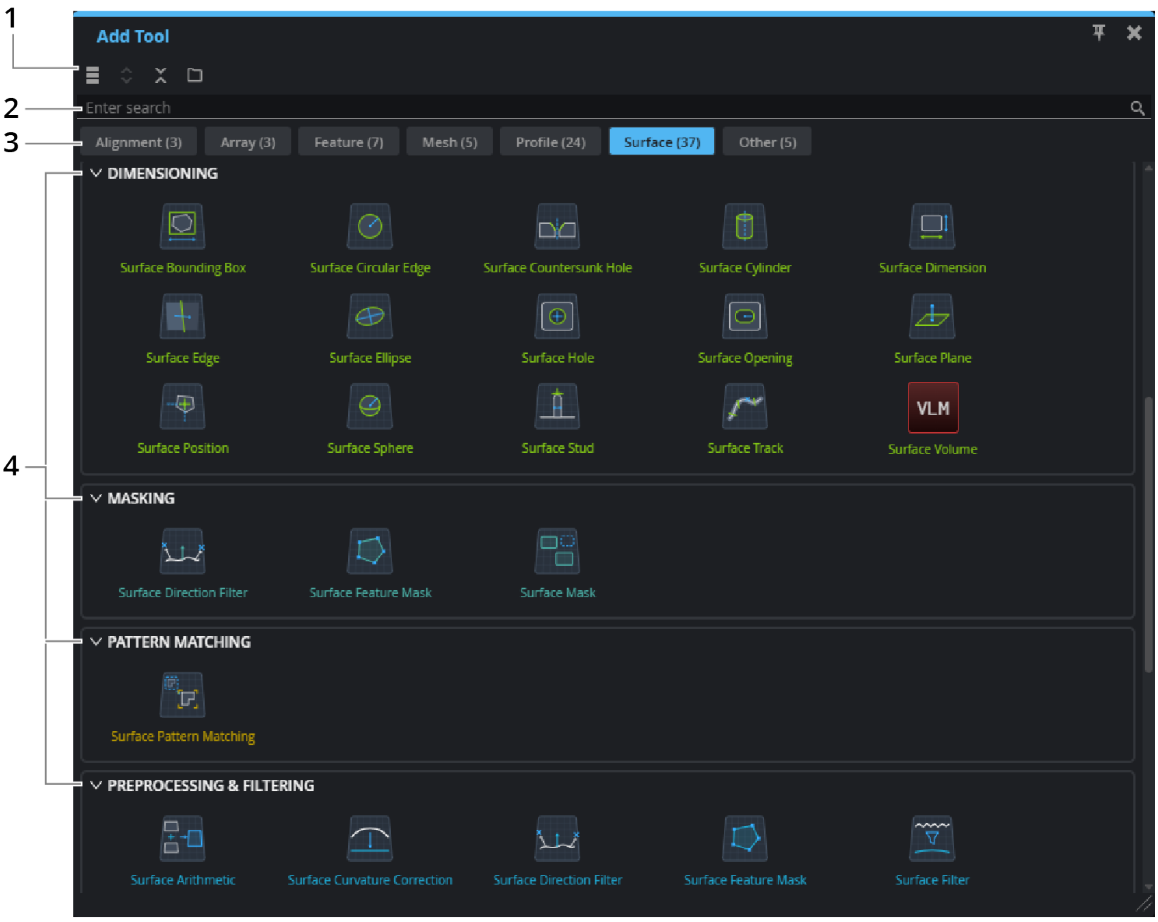
You can add a tool from the tool list in two ways: by dragging and dropping the tool onto the Tool Diagram panel (1) or by double-clicking the tool in the Tool List (2). For drag-and-drop operations, you can drop the tool where you want in a tool chain; a line in the Tool Diagram shows where the tool will be added when you drop it. Double-clicking a tool in the tool list adds the tool after the currently selected tool in the Tool Diagram.



GoPxL automatically sets the added tool's input to an appropriate output when possible. The output GoPxL selects is either directly from a sensor or sensor group, or from the output of another tool, depending on various factors (see *Tool Connection Logic* on page 167).

Understanding the Tool List

The Add Tool window helps you find tools quickly and easily.



By default, tools are organized into collapsible category groups, such as Dimensioning, Preprocessing & Filtering, and so on (4 above). Click the first icon in the icon row (or , 1 above) to switch between a simple list (no categories) and a grouped list. For a list of the categories, see *Tool Categories* below.

You can search for tools by name by typing in the text field (2). You can also filter tools using the buttons above the list of tools (3), for example by input type (Surface, Array, Profile, and so on).

By default, the Tool List closes if you click outside it. You can pin it to keep it visible when you are interacting with anything in the Inspect page, for example, to consult a tool's help topic in the Tool Help panel.

To create a tool group, click the group icon ( ), or add it from the list of tools.

Tool Categories

Categories can contain tools with any input type, unless you are filtering the tools by type. For example, by default, the Dimensioning category contains Profile, Mesh, and Surface tools. Some

tools appear in more than one category.

Tool Categories

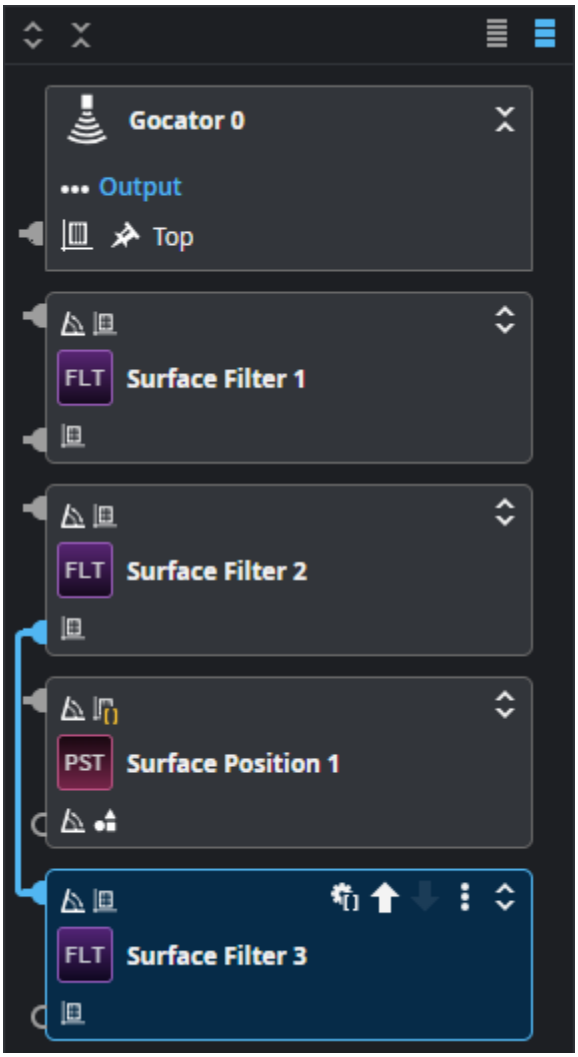
Category	Description
Array	Array tools. For more information on Array tools, see *Array Tools* on page 233.
Calibration & Alignment	Tools related to performing 6 degree of freedom system alignment. For more information, see *Aligning Sensors* on page 122.
Conversion	Tools you use to convert or extract scan data from the input data. For example, you can extract a section (profile) from Surface data, or create Surface data from Mesh data.
Coordinate Transform	Tools that let you perform translation and rotation transformations on input data.
Counting	Tools that count instances (blobs, edges, and so on) in the input data.
Data I/O	Tools related to data export or for sending data from a device (PLC) to a sensor.
Deep Learning	AI-based tools you use for applications such as anomaly or object detection, classification, and so on.
Dimensioning	Tools related to measuring dimensions, such as heights, widths, and so on.
Geometry & Features	Tools related to GoPxL's geometric features (Circle, Line, Plane and Point features) and their dimensioning, as well as the robot pose tool.
Intensity	
Logic	Tools that let you use conditional logic, formulas, or scripts in your jobs.
Masking	Masking tools.
OCR & ID	Tools that let you perform optical character recognition and read optical labels such as barcodes and QR codes.
Pattern Matching	Tools that let you define a pattern or template, identify it in the input data, and return positions, angles, counts, and so on.
Preprocessing & Filtering	Tools that let you perform filtering (such as gap filling, averaging, and so on) or masking, or binary operations on input data.
Robot & Guidance	Tools related to robot integration.
Segmentation	Tools that isolate instances in input data (blobs, segments, and so on).
Stitching	Tools that combine scan data, for example, combining the scan data from more than one sensor into single output of Surface data.
Surface Quality	Tools related to roughness and flatness.
Other	Miscellaneous tools.

Tool Connection Logic

When you add a tool, GoPxL looks for compatible outputs among the other tools to set the added tool's inputs, starting with the tool or sensor directly "above" the added tool in the list of added tools

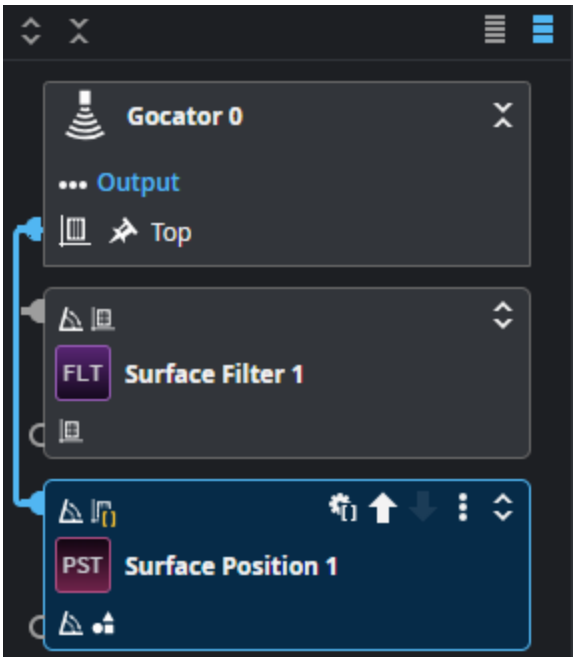
(in the Tool Diagram panel). By "compatible," we mean, for example, data type (Profile vs. Surface, or uniform vs. non-uniform or point cloud data). GoPxL ignores tools "below" the added tool for this initial search. GoPxL checks the compatibility of all the outputs in a tool before moving up to the next tool.

In the following, a Surface Position tool has been added before the third Surface Filter tool. GoPxL will start searching for compatible outputs in Surface Filter 2 and then move to Surface Filter 1. If no compatible outputs are found in the preceding tools, GoPxL will search for compatible outputs in Gocator 0. Surface Filter 3 is ignored for auto-connection.



When a preceding tool has multiple enabled outputs, GoPxL goes through them in the order they are defined in *the tool* (that is, from the top down).

If no outputs from the preceding tools are compatible with the tool's inputs, GoPxL searches through the sensor or sensor group outputs. In the following, the added Surface Position tool "skips" the Surface Filter tool, whose output has been disabled.

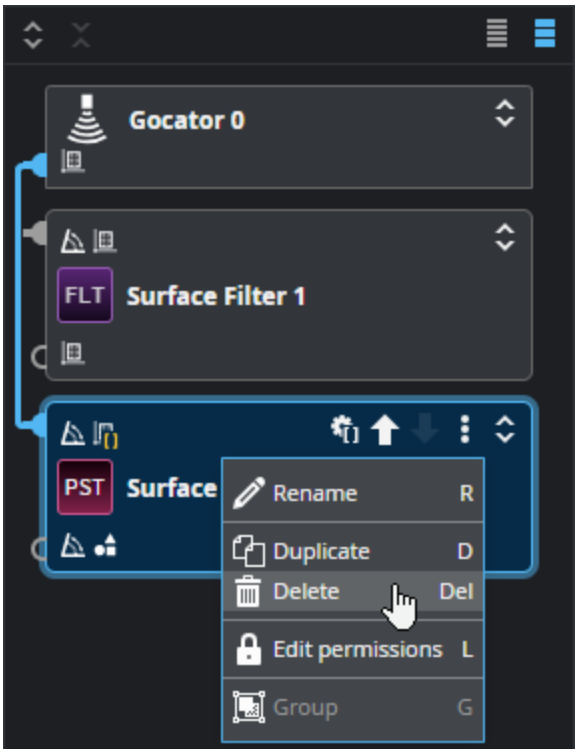


For information on other operations you can perform in the Tool Diagram, see *Working with Tool Chains* on page 172.

Removing a Tool

If you no longer need a tool, you can remove it in the Tool Diagram. Removing an unused tool can increase performance, as it no longer takes up processing time.

To remove a tool, select it in the Tool Diagram and press Delete on your keyboard. You can also open the tool's Action menu ( ) and choose **Delete**. You can also press the Delete key on your keyboard. If multiple tools are selected, a delete operation will affect all of them.

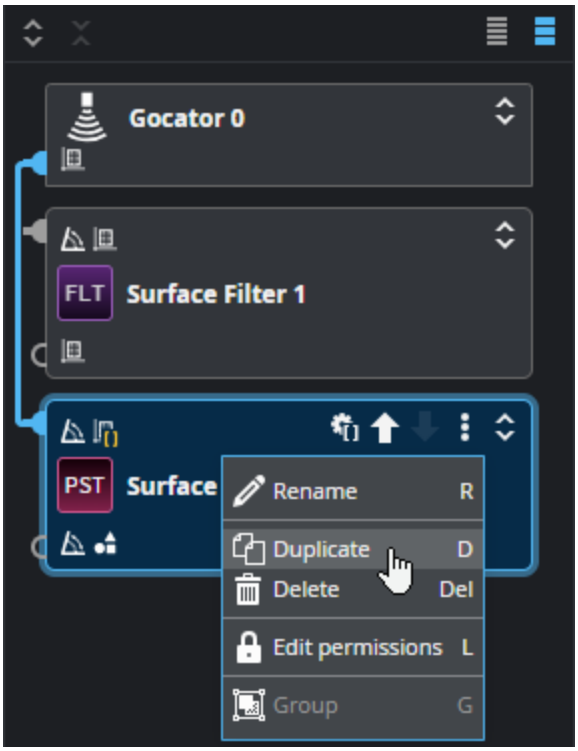


For information on other operations you can perform in the Tool Diagram, see *Working with Tool Chains* on page 172.

Duplicating a Tool

You can quickly create a copy of a tool you have previously added in GoPxL in the Tool Diagram panel. The tool inputs, parameter settings, and enabled outputs of the original are copied. This is useful, for example, when you need almost identical tools with only minor variations, such as different maximum and minimum decision values, or regions of interest shifted from one component to another.

In the Tool Diagram panel, you duplicate a tool by opening the tool's Action menu ( ) and choosing **Duplicate**. You can also press the D key on your keyboard. If multiple tools are selected, a duplication operation will affect all of them.



For information on other operations you can perform in the Tool Diagram, see *Working with Tool Chains* on the next page.

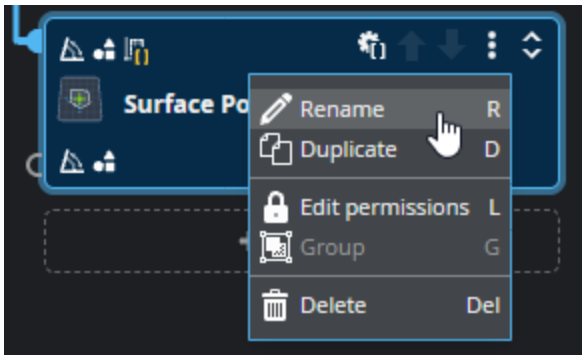
Renaming a Tool

In the Tool Diagram panel, you rename a tool using the action menu of an individual tool.

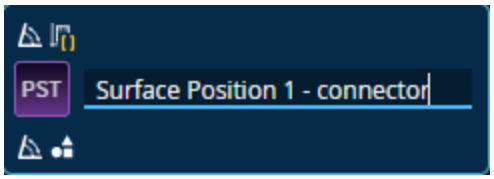
To rename a tool in the Tool Diagram panel

1. Click the action menu icon ( ).

   A context menu containing various actions 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 outside the text field.

For information on other operations you can perform in the Tool Diagram, see *Working with Tool Chains* below.

Working with Tool Chains

The Tool Diagram panel between the tool list and Tool Configuration panel provides a visual representation of the data flow in a system (the output from a sensor group, and the input and output of tools). It lets you create and view complex tool chains with drag-and-drop, mouse, and keyboard operations, letting you quickly and easily implement and maintain applications requiring multiple, interconnected tools. Tools are represented by "blocks" in the tool chain.

All data types (if available) and their relationships between tools are displayed:

- Profile data (either directly from a sensor group's output or from tool output)
- Surface data (either directly from a sensor group's output or from tool output)
- Measurements (for use as anchors)
- Geometric features
- Tool data (some data outputs are intended to be consumed only by SDK applications and can't be used as part of a tool chain)

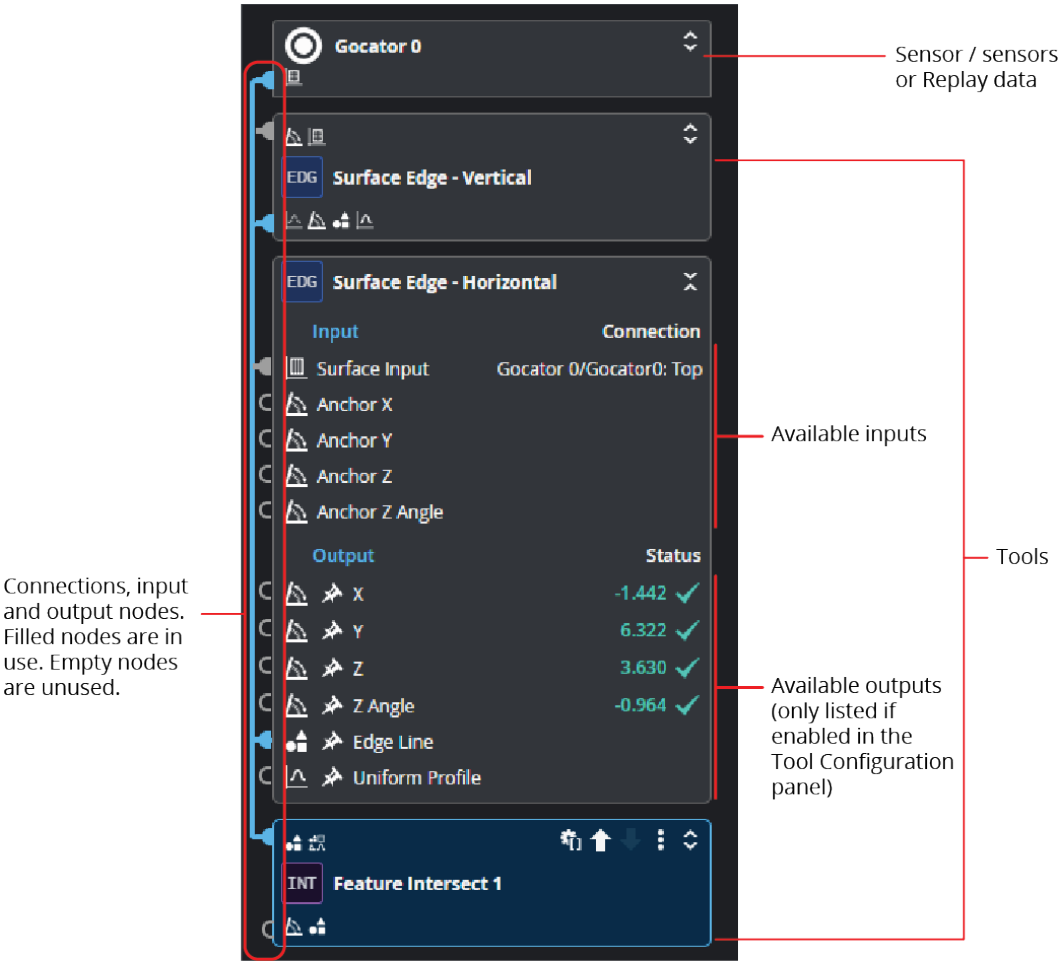
You can perform many operations on multiple tools at once. For more information, see *Keyboard and Mouse Shortcuts* on page 70.

For details on how the Tool Diagram panel displays information, see *Understanding the Data Flow in Tool Chains* on page 177.

For details on how to connect and disconnect, see *Connecting Tools* on page 182 and *Disconnecting Tools* on page 184.

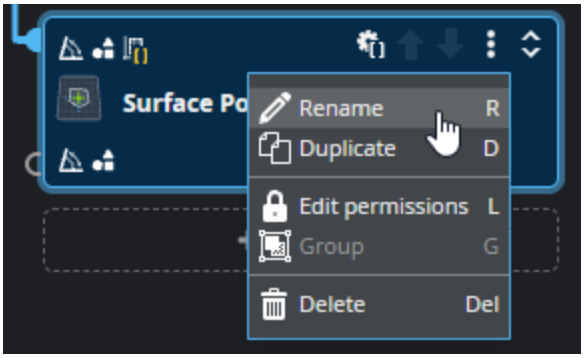
The Tool Diagram panel is open by default. The inputs, parameters, and outputs of the tool selected in the Tool Diagram panel are displayed to the right of the Tool Diagram panel.

The following shows the main features of the Tool Diagram panel.



Tool Diagram panel showing sensor group, tools, outputs/inputs, and data flow connections.

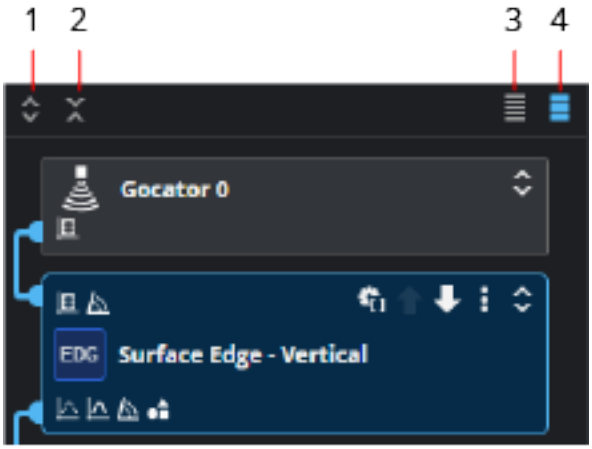
At the top of a tool, an action menu provides functions to rename, duplicate, and delete the current tool.



Action menu (collapsed tool)

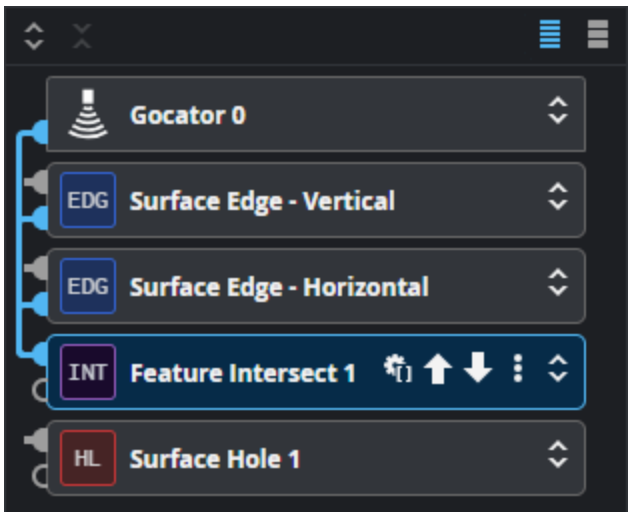
Changing Tool Display Options

The buttons at the top of the Tool Diagram panel let you control how the panel displays sensor groups, 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.

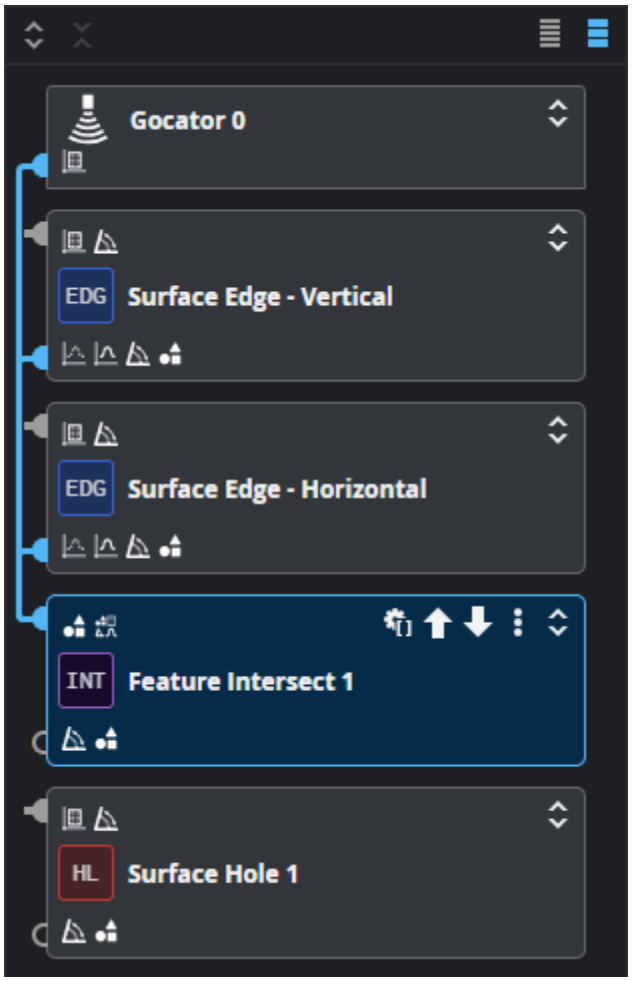


The following describes the toolbar's functions:

1. Open all: Expands the sensor or sensor group block and all tools in the Tool Diagram panel, displaying a list of available inputs and enabled outputs for each one.
2. Close all: Collapses all items in the Tool Diagram panel.
3. 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.



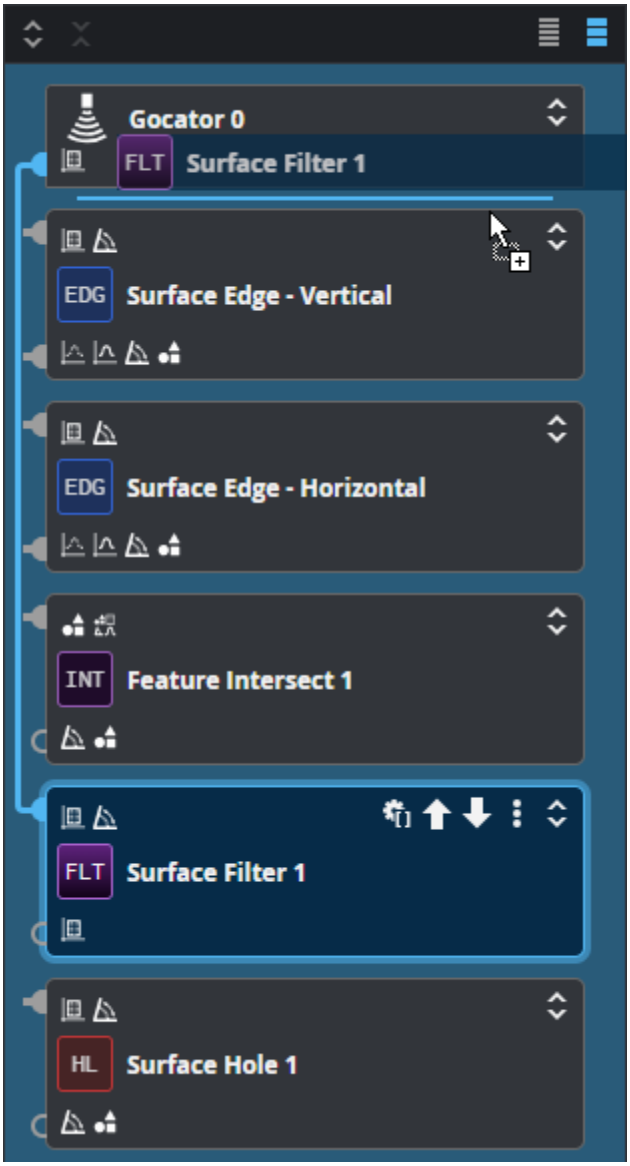
4. Standard view: Shows small icons that indicate the types of the inputs and outputs the sensor group or a tool has. For a list of inputs and outputs, see *Data Types* on the next page.



Reordering Tools

You can reorder tools in the Tool Diagram to organize them more logically or to better represent the data flow. For example, you could group tools that output geometric features with the tools that use them, or group tools you use as anchors with the tools that use those anchors. Note that the order of tools in the Tool Diagram panel does not affect the flow of data.

You can reorder tools you have added by dragging and dropping them to the position you want in the list of tools. You can also click the up and down arrows in a selected tool block, or a block the mouse is hovering over. You can use your keyboard's up and down arrow keys to move the selected tool up or down in the tool chain. If multiple tools are selected, a move operation will affect all of them.



Moving the Surface Filter tool to the top of the tool chain.

Data Types

GoPxL represents the data types of inputs and outputs with icons. When an input or an output

Icon	Description

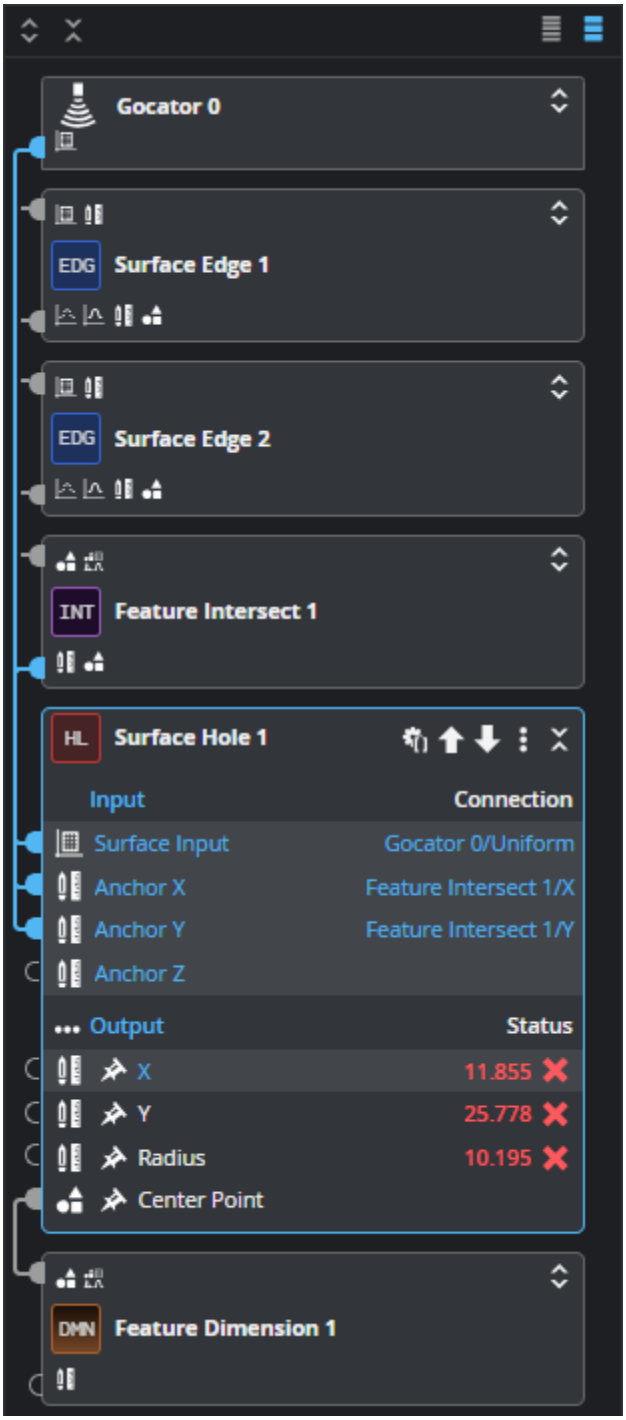
	Uniform Surface data
	Point Cloud Surface data (non-uniform)
	Uniform Profile data
	Point Cloud Profile data (non-uniform)
	Measurement
	Geometric feature
	Multiple input types

Data type icons, as well as tool icons, can be modified with a "[ ]" badge to communicate information related to arrays and batching. For more information, see *Arrays, Batching, and Aggregation* on page 199.

Understanding the Data Flow in Tool Chains

The rectangular blocks displayed in the Tool Diagram represent a sensor or group of sensors at the top, and any tools you have added below that. Sensor groups display output connection nodes, whereas tools display both input and output connection nodes.

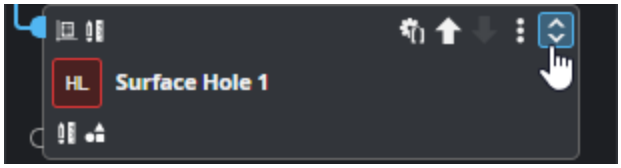
The appearance of nodes changes depending on whether they are connected and whether they are selected. Connections that are used are solid. Connections that are not used are empty. When a sensor or tool is expanded, you can see which specific inputs or outputs are used as part of the tool chain. For example, in the expanded Surface Hole tool below, we can see that the first three inputs (*Surface Input* and the X and Y anchors, receiving their input from the sensor group at the top and from Feature Intersect, respectively) and the Center output are involved in the chain of sensor group and tools.



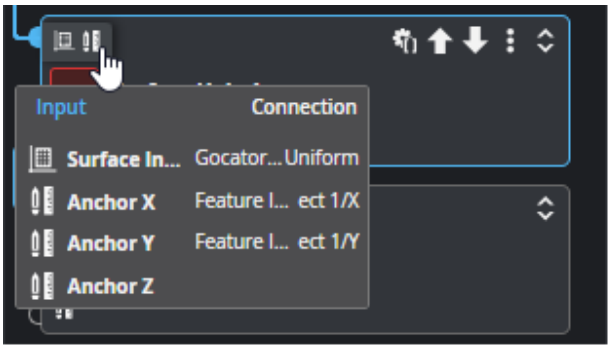
When a tool is collapsed, however, you only know that at least one input or output is used (or none at all). For example, looking at the collapsed Feature Dimension tool at the bottom, we know that at least one input (the connection node at the top) is used, and that none of the tool's outputs are used. Also, we know that inputs and outputs of the three collapsed tools at the top are used, but not exactly which ones.

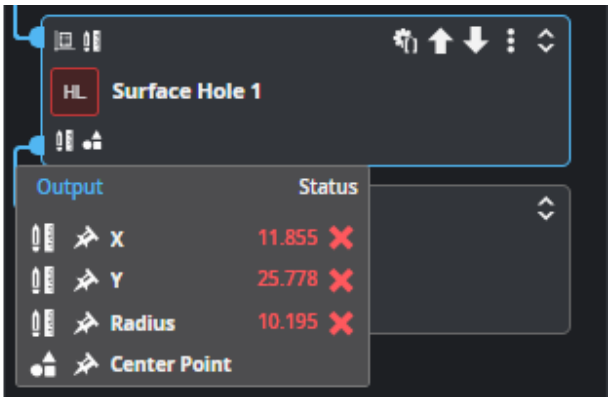
In both cases (collapsed or expanded), the data flow of the selected tool or sensor block is indicated by blue connection lines. For more information, see below.

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.

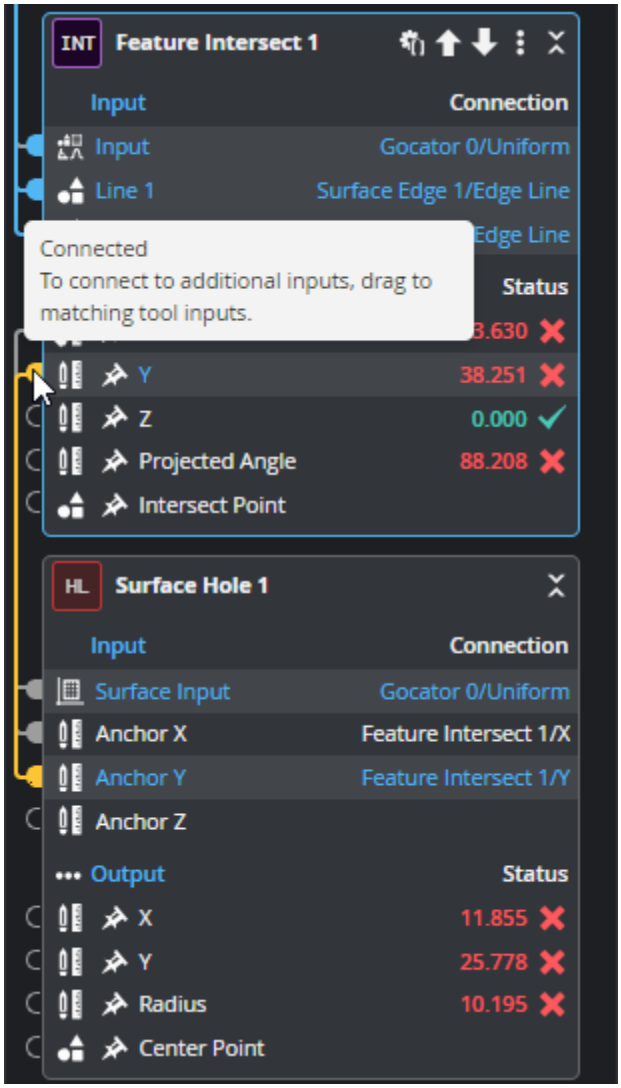


When a tool is collapsed, you can see which inputs are connected and to what source, and also see which outputs are enabled and their values, by clicking one of the rows of small icons at the top or bottom of the block. You can also pin outputs from here; for more information on pinning, see *Displaying and Pinning Outputs* on page 94.

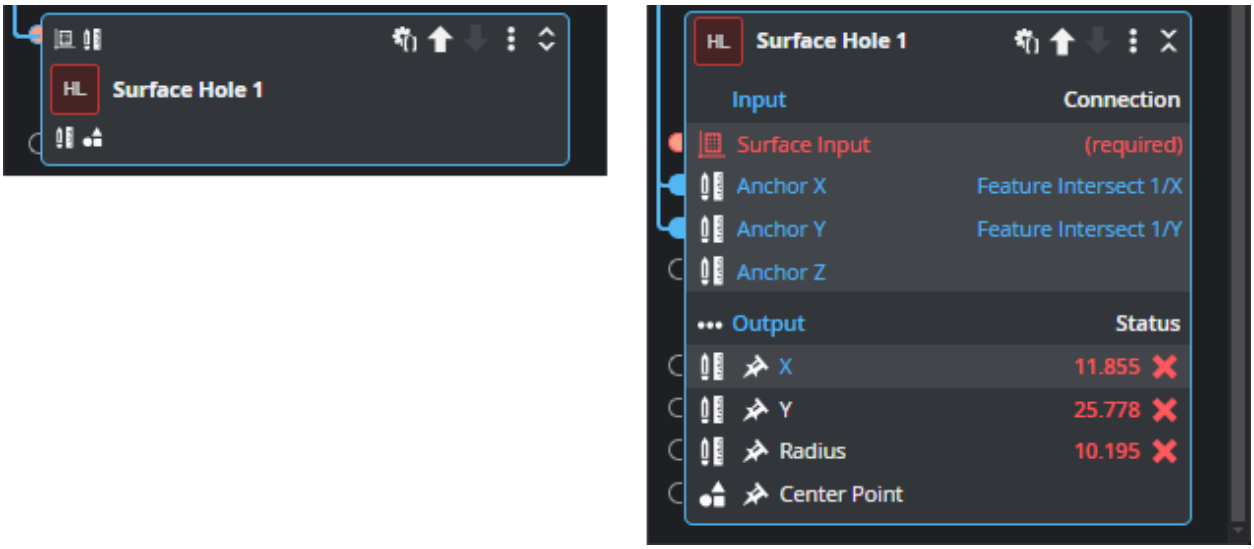




If you hover the mouse pointer over a blue connected node, a part of the blue connection line is highlighted in yellow to indicate what it is connected to. In the image below, you can see that by hovering over an output (the Y measurement of the Feature Intersect tool) it is used as an input (the Y anchor) of the Surface Hole tool at the bottom.

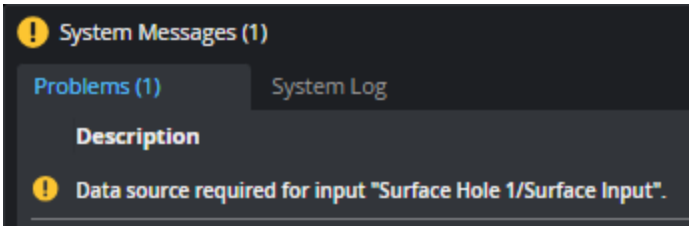


If a tool's input is missing, that input is displayed in red in the Tool Diagram panel to show that you must connect it to something.

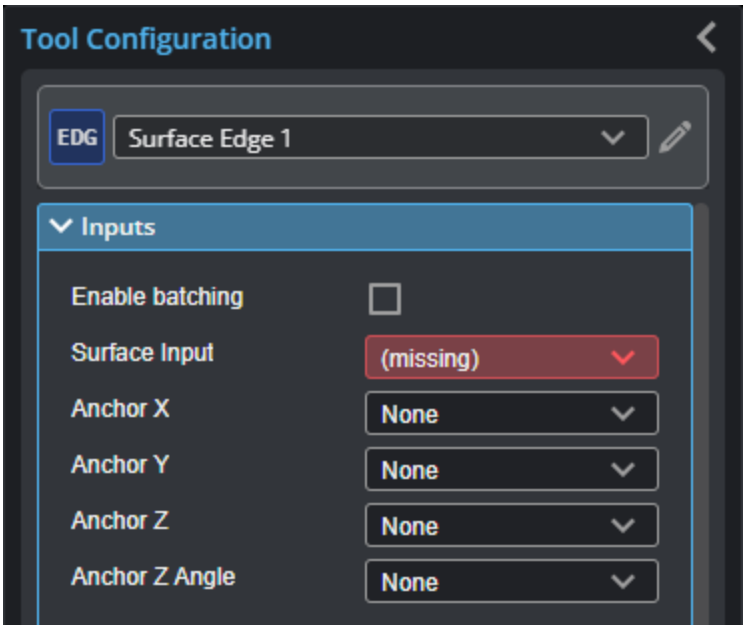


Collapsed tool and expanded tool with a missing input

You can also consult the **Problems** tab in the **System Messages** panel at the bottom of the interface to see which tools are missing inputs.



In the Tool Configuration panel, a tool whose input is no longer available (for example, you have removed the preceding tool in the tool chain) displays "(missing)" in the input parameter.



Connecting Tools

The Tool Diagram panel lets you quickly connect tools using drag-and-drop operations.

In the following, we connect a measurement output from one tool to an anchor input of another tool. However, the same procedure applies when connecting other kinds of outputs to inputs, such as when connecting a geometric feature output from one tool to the input of another tool, or when connecting Surface output (such as the output from the Surface Filter tool) to the input of another tool.

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. You can enable a tool's output in the tool's **Outputs** section in the **Tool Configuration** panel, or by using the **Output** menu in the tool's block in the Tool Diagram panel.

   For information on adding tools, see *Understanding the Data Flow in Tool Chains* on page 177.

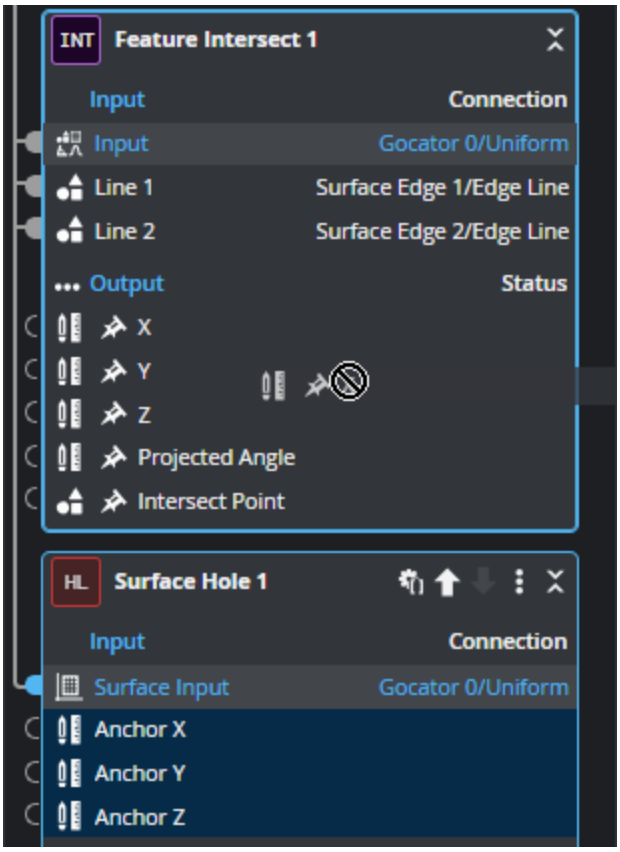
2. Locate the tool whose output you want to use (the "source" tool).

3. Do one of the following:

   From an expanded tool

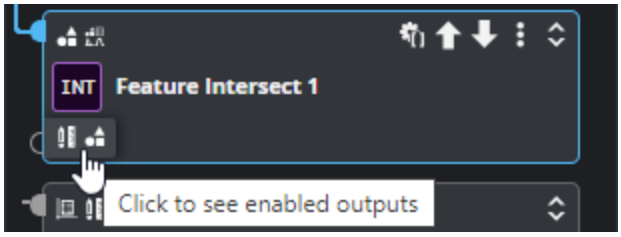
   a. Click and hold the output you want to connect to the other tool's input and drag it to the input.

      Here, we are dragging the X measurement from the Feature Intersect tool. Tools with compatible inputs (here, the anchor inputs of the Surface Hole tool), are highlighted with a dark color.

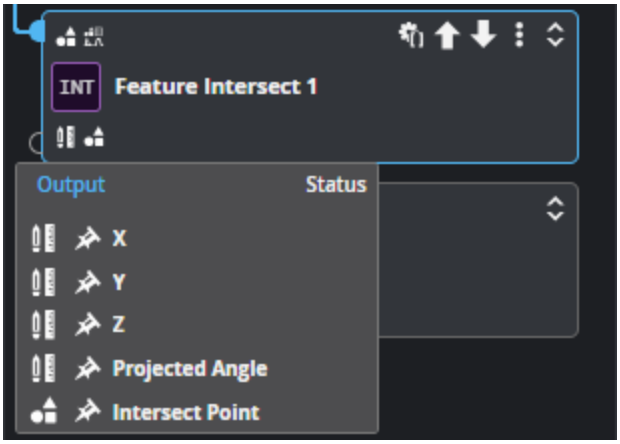


From a collapsed tool

a. Click the small row of icons at the bottom of the tool to expand the list of

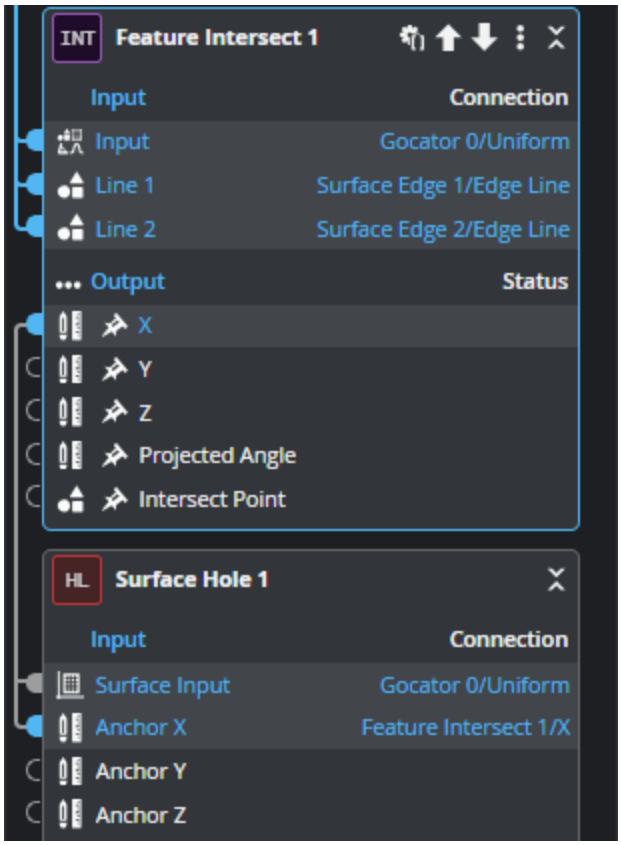


A list of enabled outputs is displayed in a pop-up list.



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).

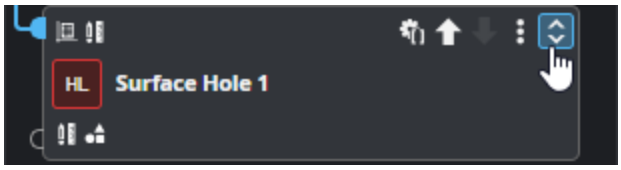


Disconnecting Tools

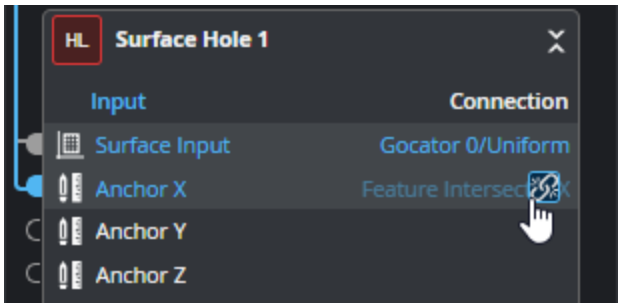
You can quickly disconnect an input in the Tool Diagram panel, but only if the tool containing the input is expanded.

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.

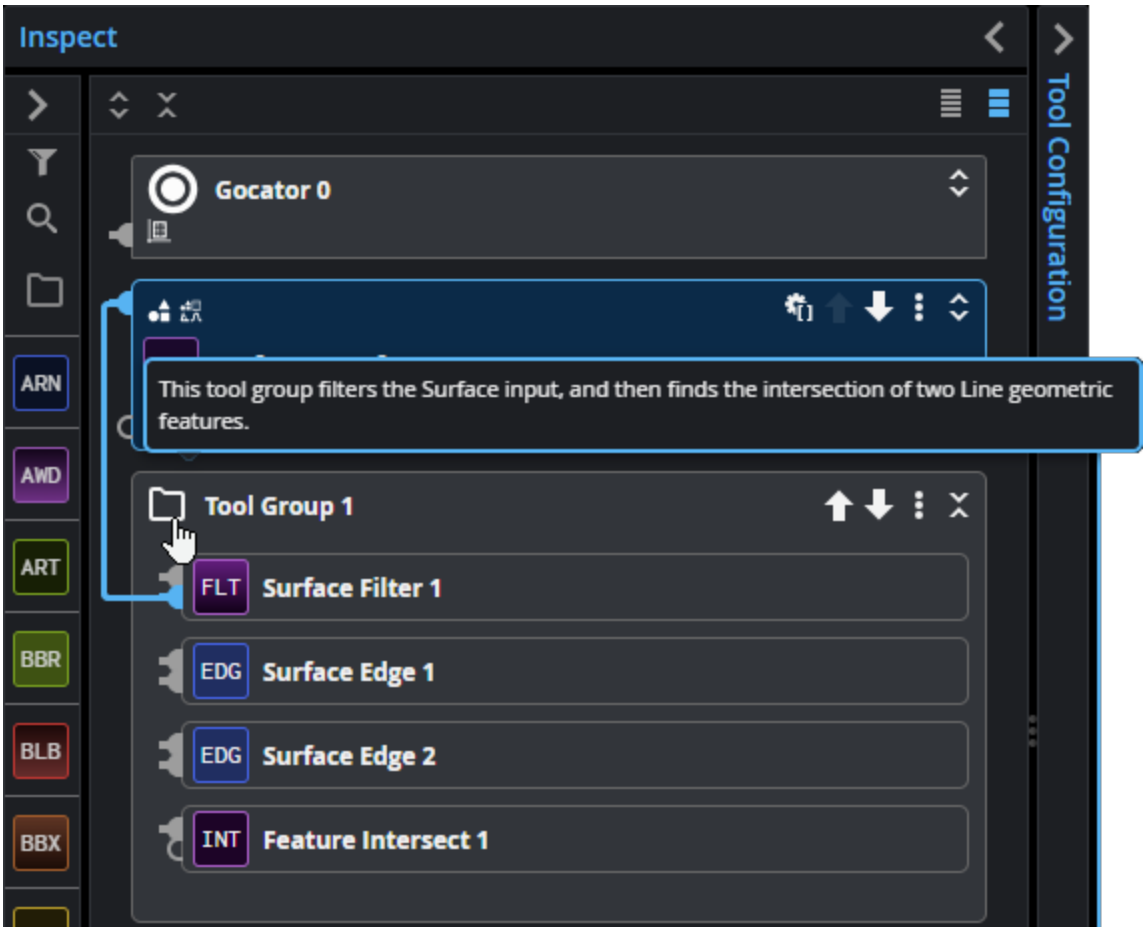
   The input is disconnected from the other tool's output.

Tool Grouping

GoPxL supports "tool grouping," letting you better organize large jobs and improve the application design workflow. Tool groups are collapsible and can be named to easily identify them. You can also duplicate tool groups: duplicated groups contain duplicates of the original group's tools. This saves you time when you need set up the same set of tools multiple times, for example, for multiple features on a target. Just adjust the parameters in the group's tools as needed for each feature.



You can also provide a description of a tool group, which is displayed when you hover over the tool group's folder icon. Double-click the icon to add or edit a description, or choose this action from the tool group's action menu.



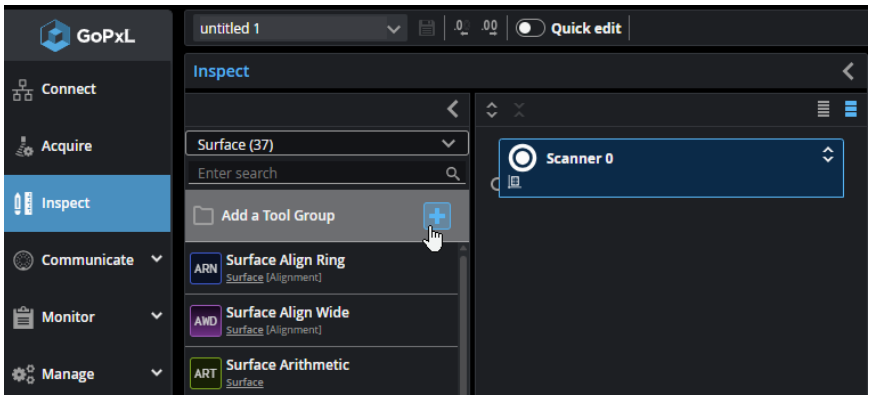
Tool grouping actions are available in the Action menu of a tool block (adding tools to a new or existing group) and in a tool group's Action menu.

Note that tool groups can't be nested. Also, tool blocks in a group can't be expanded.

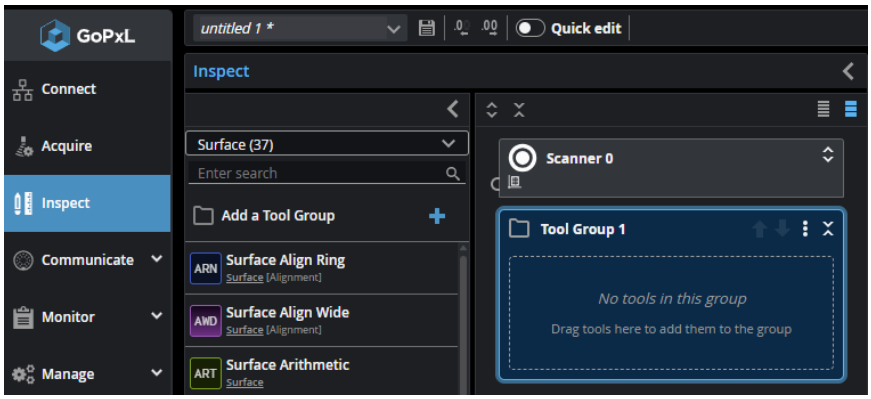
For a list of shortcuts, see *Keyboard and Mouse Shortcuts* on page 70.

To create a tool group

1. In the tool list, click the plus symbol next to **Add a Tool Group**.

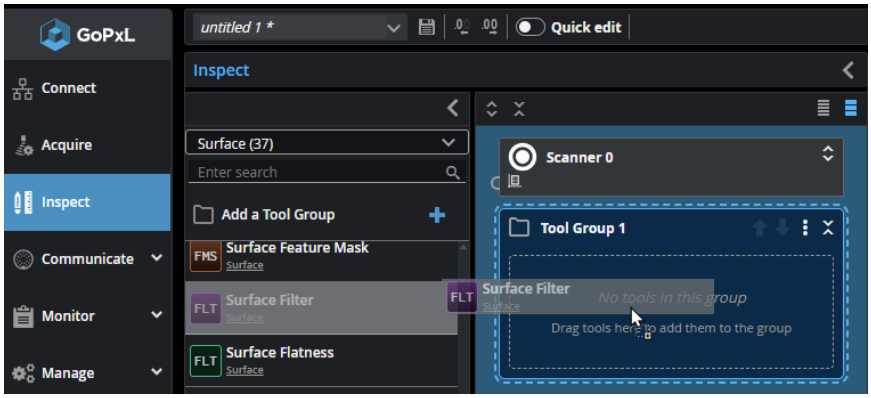


   GoPxL adds a tool group.

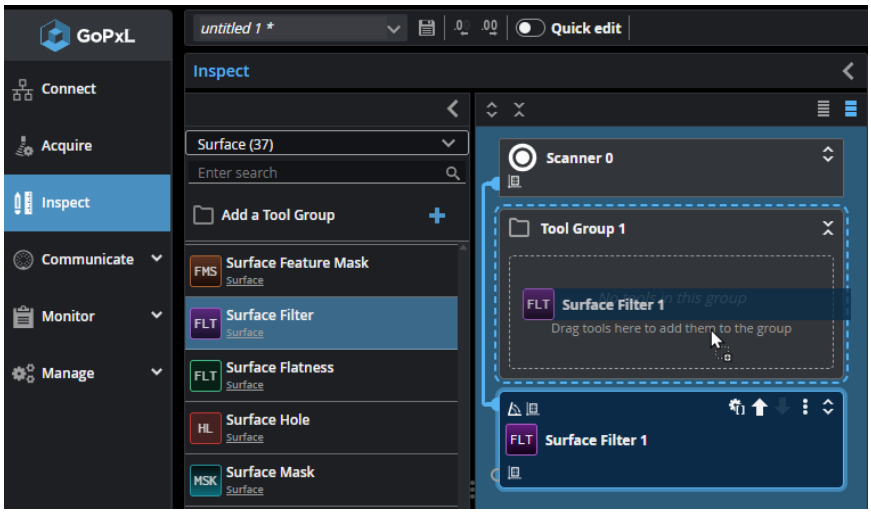


2. Add tools to the group.

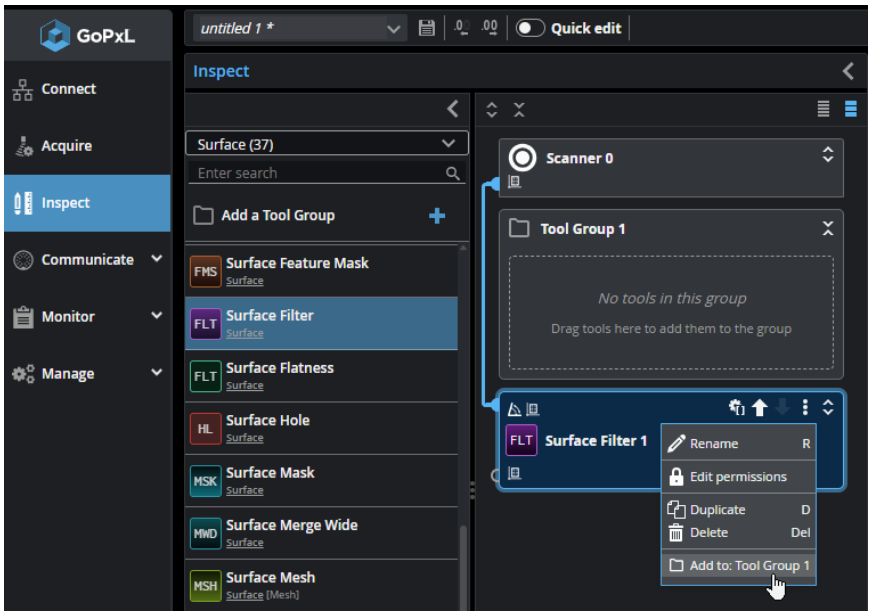
   You can add tools to the group by dragging and dropping them into the group from the tool list:



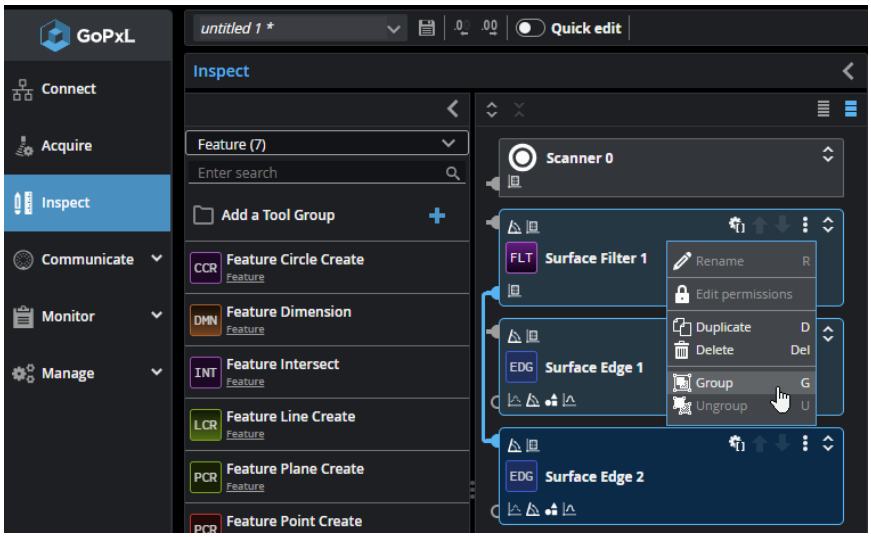
   Or by dragging and dropping one or more tools already in the Tool Diagram:



   You can add tools from the tool block's "actions" menu.

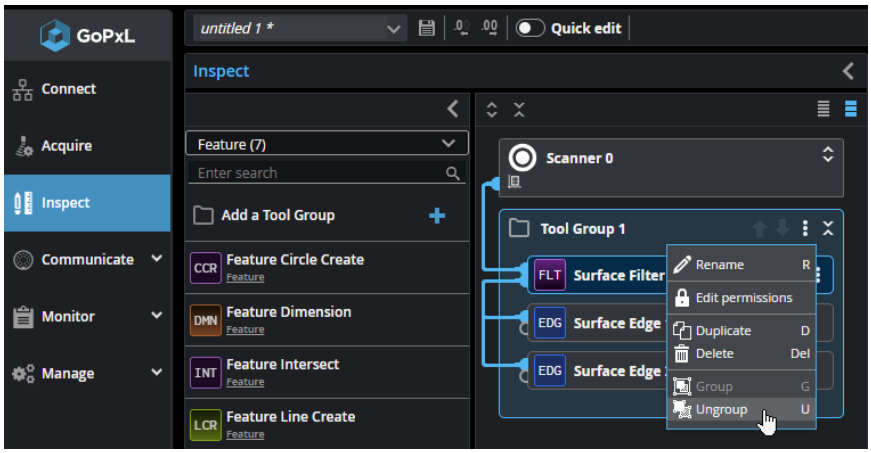


Finally, you can add tools to a new group by selecting two or more and selecting "Group" from the action menu of one of the tools:

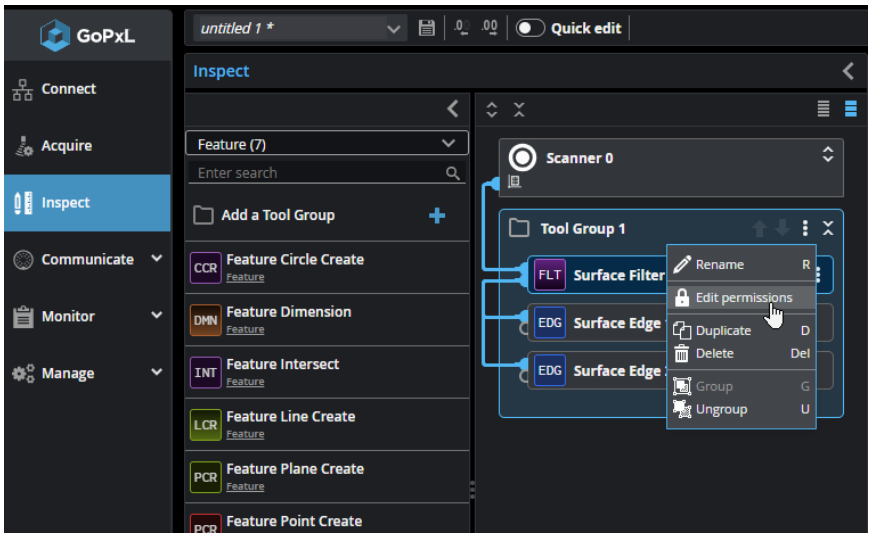


Similarly, you can remove a tool by dragging it from a group and dropping it outside the group, in the Tool Diagram.

You can remove all tools from a group by selecting "Ungroup" from the tool group's action menu:

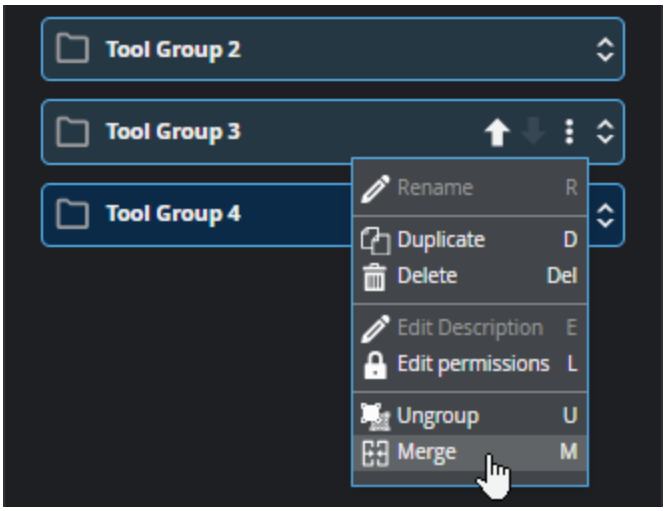


3. If required, you can lock a tool group to prevent unauthorized changes to it and to the tools it contains.



   After a tool group has been locked, tools can't be removed or added to it, and the tools themselves are locked. For more information on tool and tool group locking, see *Tool and Group Locking* on the next page.

You can merge tool groups by selecting them and choosing "Merge" from the group's Action menu. Note that the group whose Action menu you use for this is the group into which the other group's tools will be merged. For example, in the following, the tools in "Tool Group 2" and "Tool Group 4" will be moved to "Tool Group 3."



Tool and Group Locking

GoPxL lets you lock tools to prevent unauthorized changes to tool inputs, parameters, and outputs. You can also lock an entire tool group, which locks all tools it contains, and also prevents adding tools to the group, or removing them. When a tool or group is locked, all actions except "Edit permissions" on the Action menu are blocked. You can also choose between simply locking to prevent changes, or hiding tool parameters entirely.

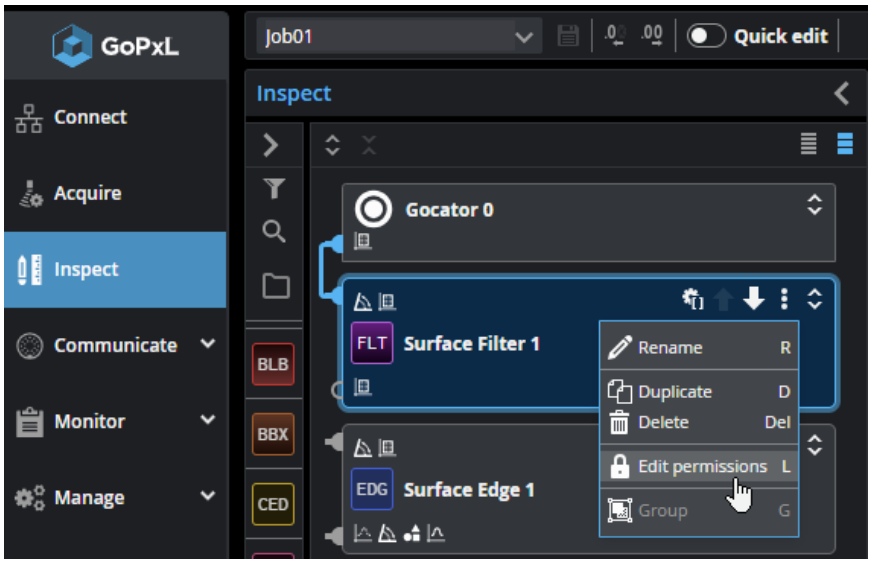
Note that after locking a tool or group using a password, you must use the same password to lock subsequent tools or groups.

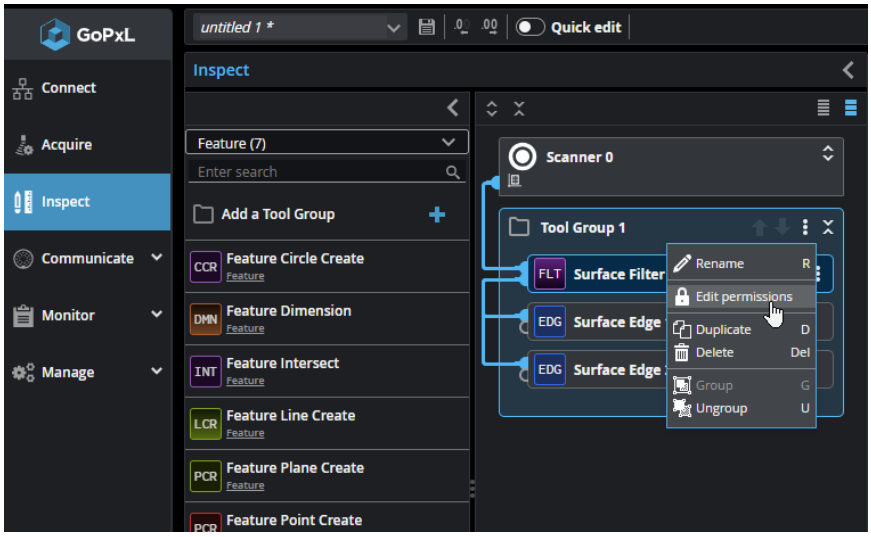
After the last password-protected locked item has been unlocked, the password is cleared: if you lock any items after this, you must set the password again.

For a list of shortcuts, see *Keyboard and Mouse Shortcuts* on page 70.

To lock a tool or tool group

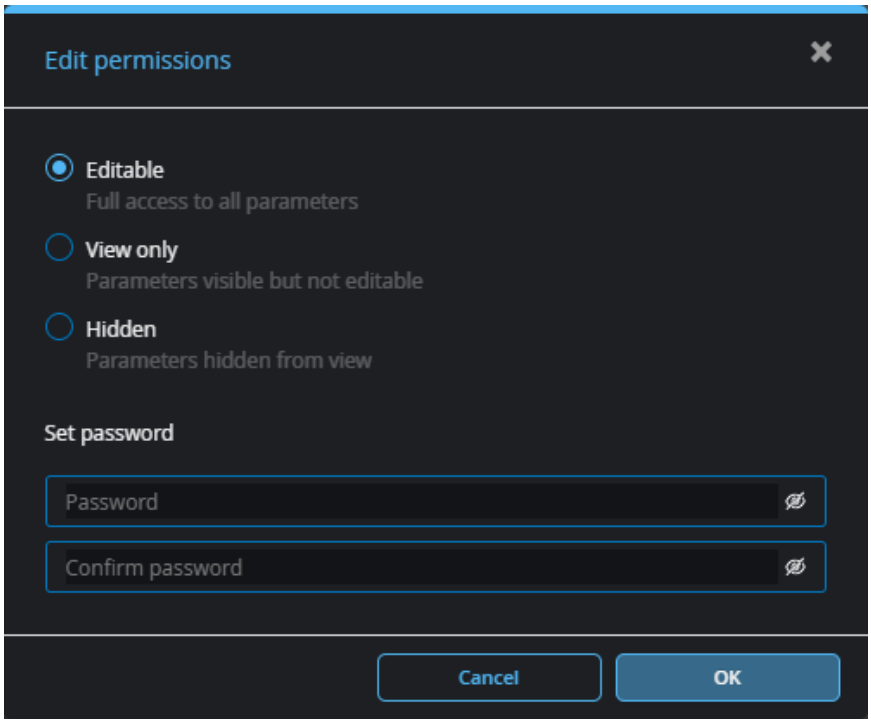
1. Click the action menu of the tool block or the tool group block and select "Edit Permissions."





An **Edit permissions** dialog is displayed.

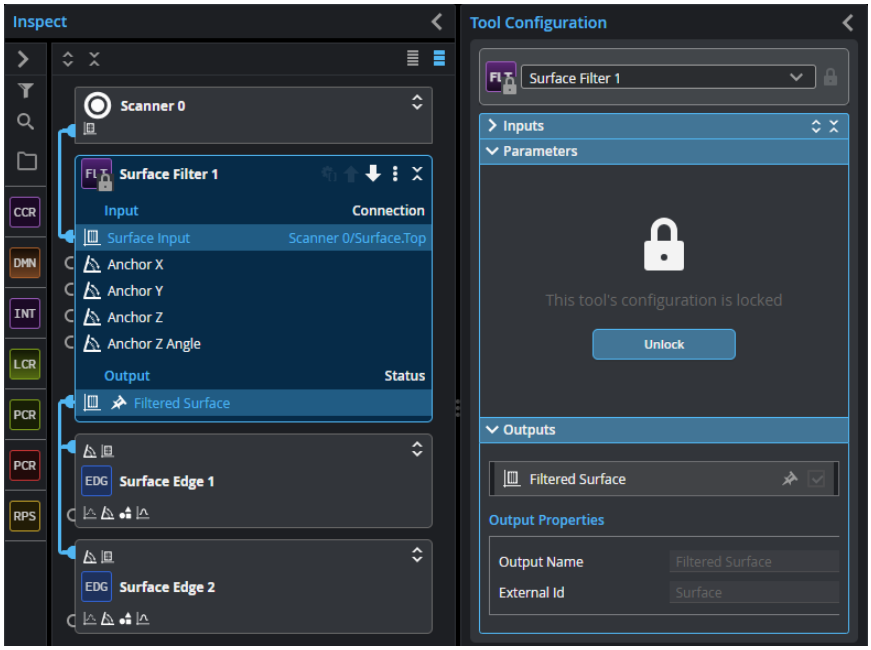
2. In the permissions dialog, choose how the tool or tool group is locked and type a password in the fields.



View only - Inputs, parameters, and outputs are visible but can't be changed; users can interact with drop-downs. Outputs can still be pinned or unpinned.

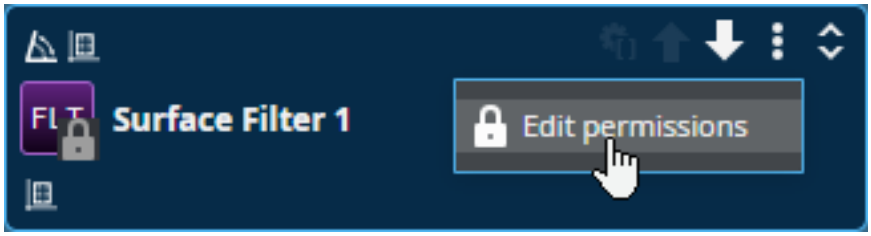
Hidden - Parameters are completely hidden. Inputs and outputs are visible but can't be changed.

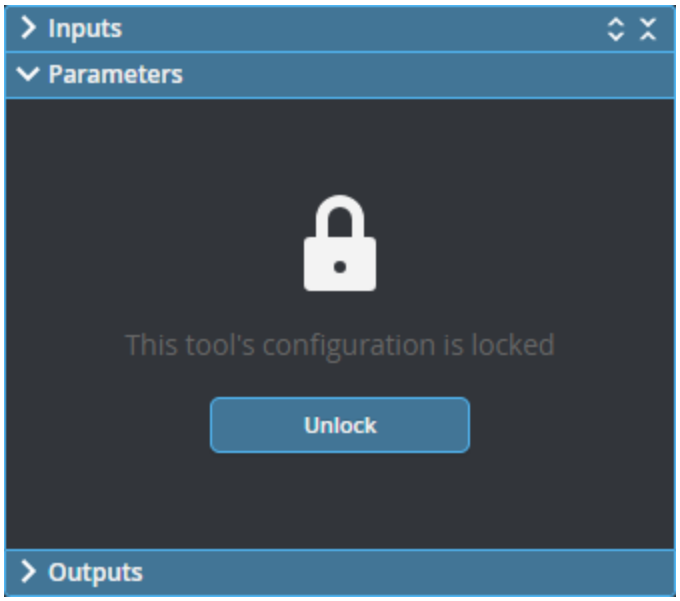
After a tool or group has been locked, a lock icon is shown over its icon, and with the Hidden permission, in the **Parameters** section.



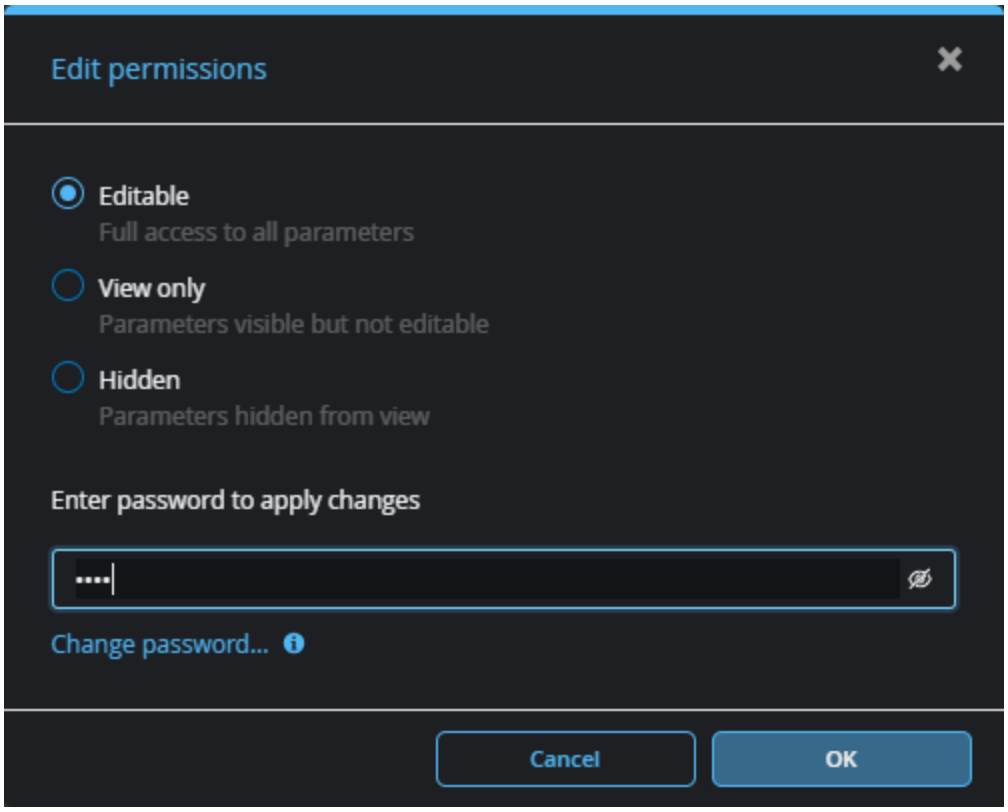
To unlock a tool or tool group, or change permission level

1. Select **Edit permissions** in the action menu, or **Unlock** in a tool's **Parameters** section if you chose to hide the parameters.





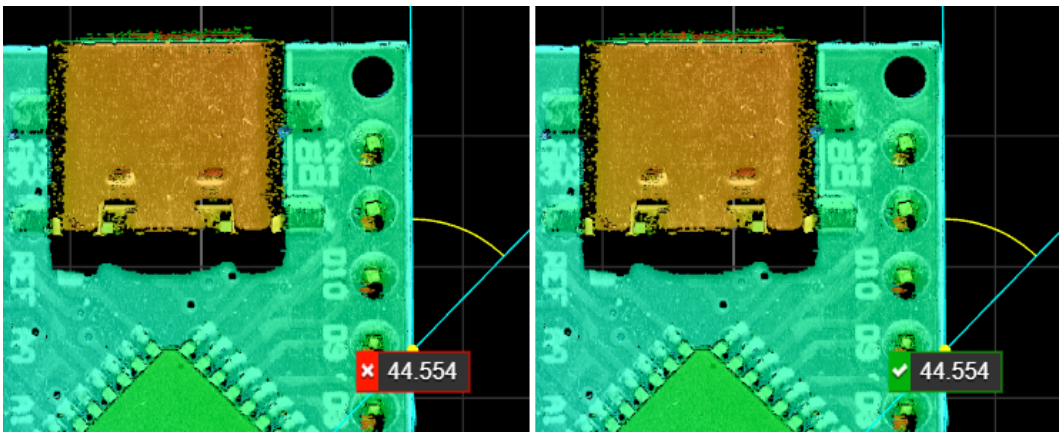
2. In the **Edit permissions** dialog, set the permission level to Editable, type the password in the password field, and click OK.



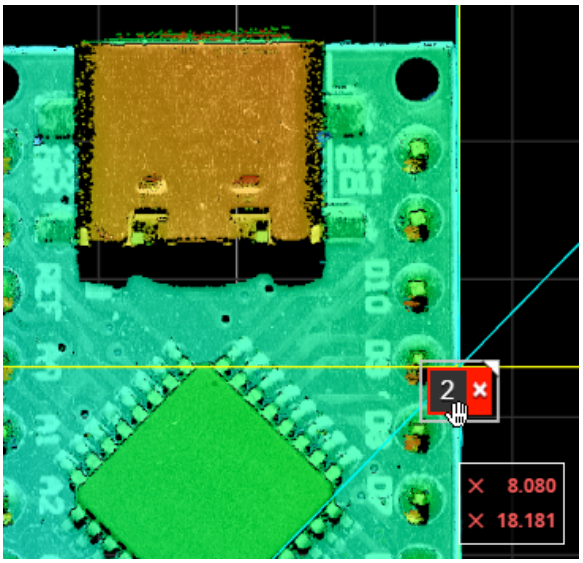
You can also change the permission level from Hidden to View only.

Measurement Values and Labels

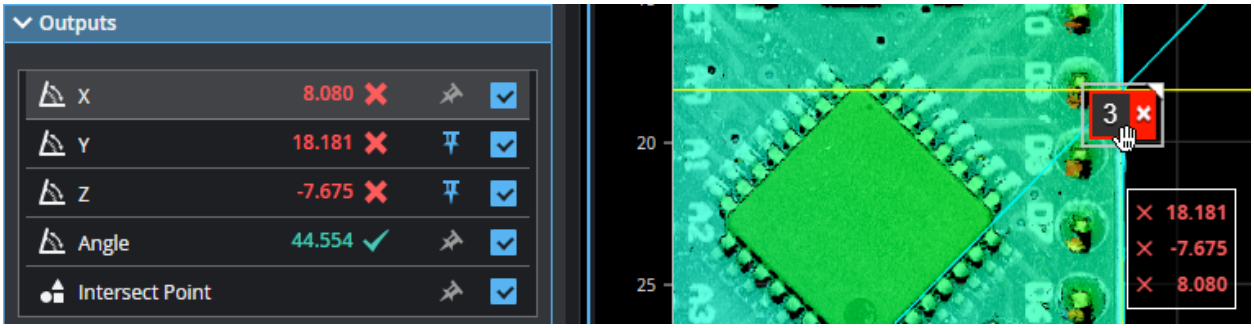
GoPxL displays currently selected and pinned measurement values in the data viewer, in addition to labels (depending on the tool). When a single measurement is selected or pinned, the measurement value is displayed in a label to the right of a fail (✖) or pass (✔) icon.



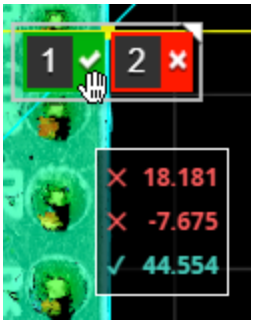
When multiple measurements appear close together in the data viewer, the label style changes and indicates the number of measurements that are displayed in the same location before the pass/fail icon. To display the values, hover the mouse pointer over the label.



When both pinned and selected measurements are displayed, pinned measurements are listed first, in their order in the tool's Outputs panel, followed by the selected measurement. In the following, the pinned Y and Z measurements are listed first, followed by the selected X measurement.

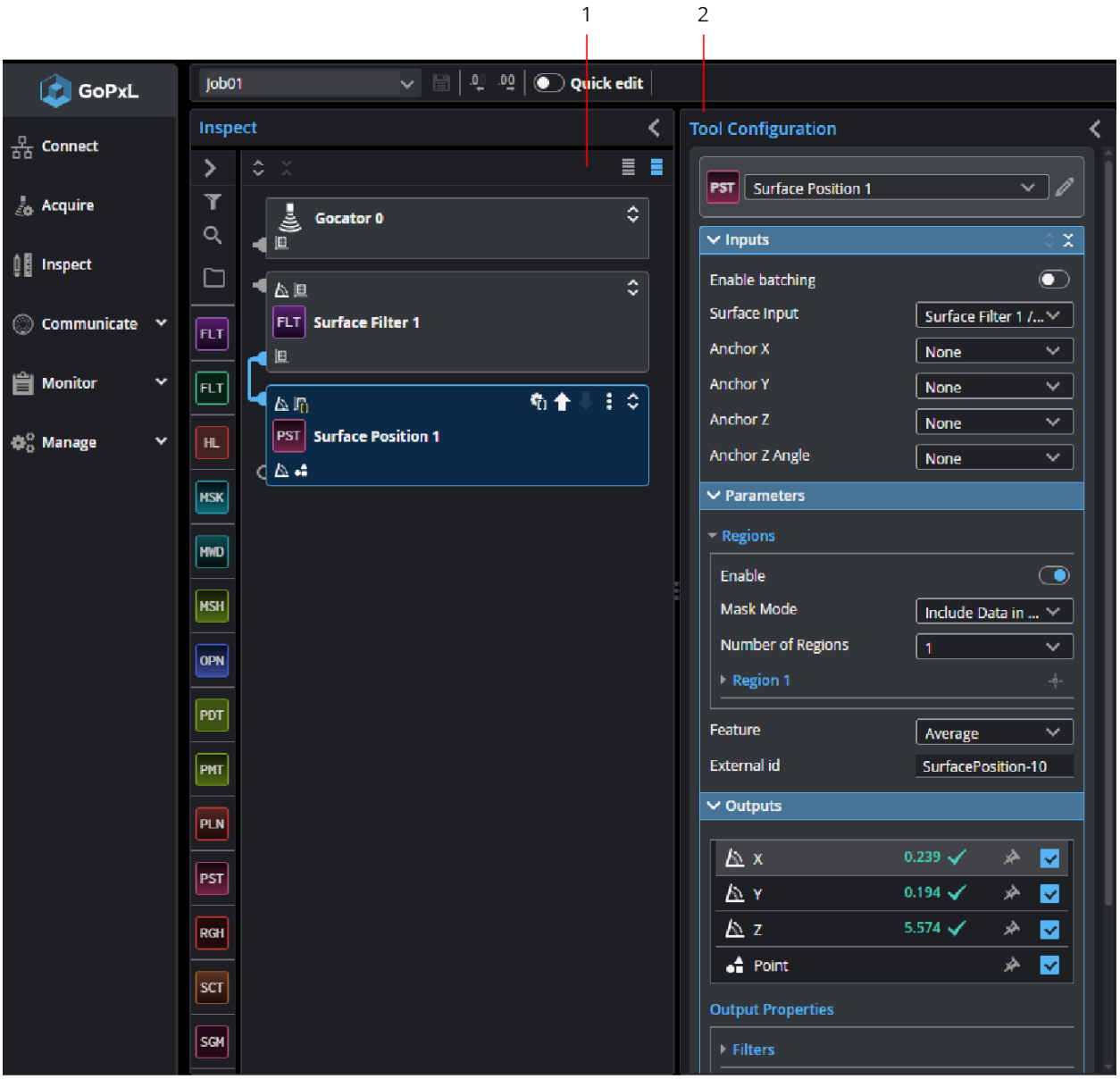


The counts for pass and fail measurements are indicated separately, but the values are listed together, as described above.



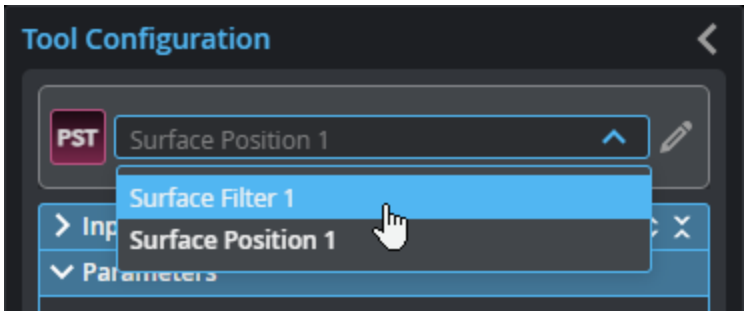
Tool Configuration

You configure a tool's inputs, parameters, and outputs in the tool configuration area to the right of the Tool Diagram.



	Element	Description
1	Tool Diagram	Lists the tools that you have added. The Tool Diagram lets you better visualize the workflow in complex tools chains. For more information, see *Working with Tool Chains* on page 172.
2	Tool configuration	Lets you configure the selected tool using the **Inputs**, **Parameters**, and **Outputs** sections in this area.

Element	Description
	You can select which tool you want to configure from the tool selector drop-down, in addition to selecting the tool in the Tool Diagram.



The following sections describe the parameters common to most tools.

Inputs and Outputs **198**

Arrays, Batching, and Aggregation **199**

Arrays from Multi-instance Tools 200

Arrays from Multi-layer Output 202

G2 Multi-sensor Output 203

Batching 204

Aggregating 206

Batch and Array Indicators in GoPxL 208

Measurement Anchoring **209**

Regions **214**

Standard Regions 216

Flexible Regions 217

Working with Circular and Elliptical Regions 221

Region Rotation 223

Feature Points **224**

Geometric Features **227**

Fit Lines **227**

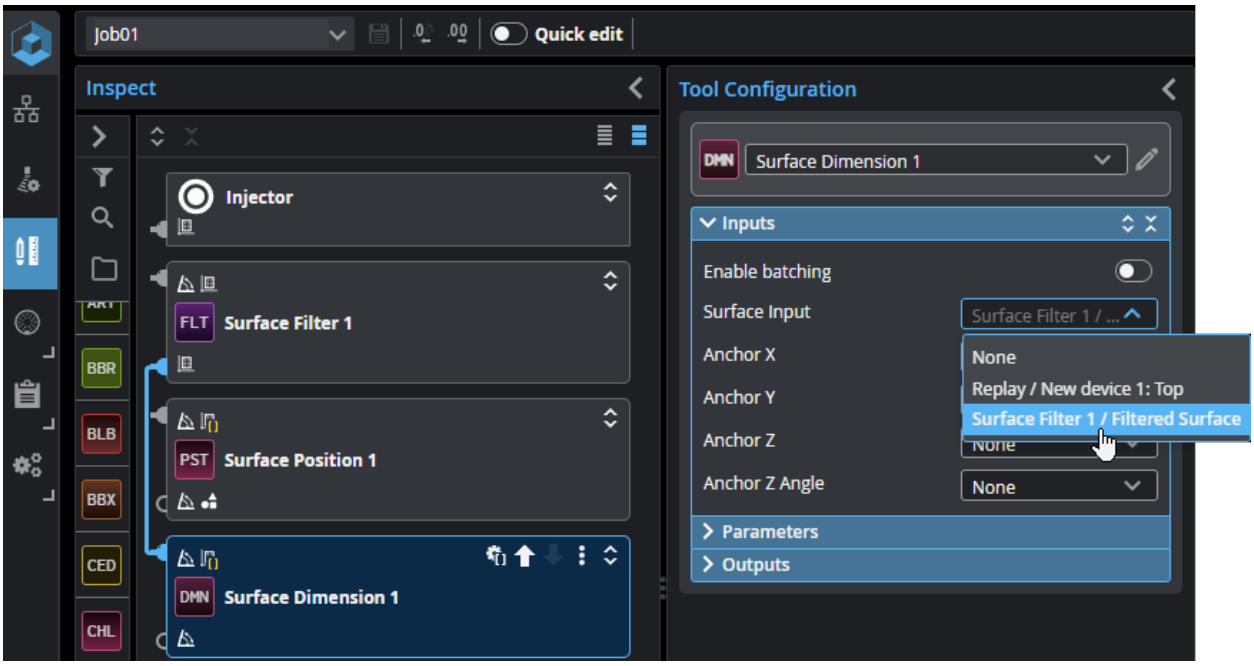
Output Filters **228**

Decisions **229**

Inputs and Outputs

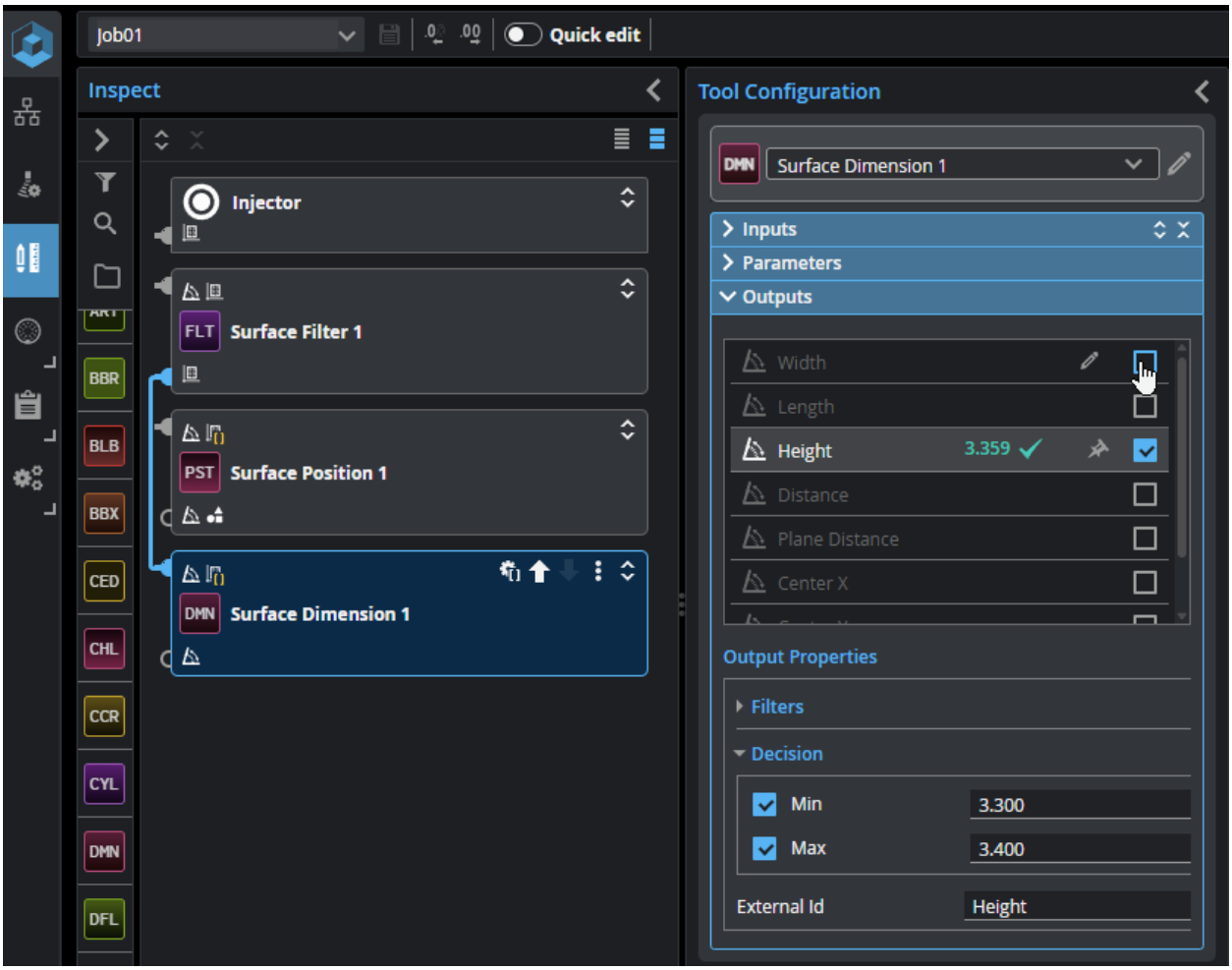
All tools take one or more inputs, either directly from a sensor or sensor group, or from another tool. All tools produce output, such as measurements or modified scan data.

The simplest case is a tool taking a single input, such as a profile (line profilers only) or surface data. The tool performs measurements on the input or processes it, and produces outputs.



Most tools can also take "arrays" as inputs. For more information on arrays and how tools work with them, see *Arrays, Batching, and Aggregation* on the next page.

All of the outputs (measurements, features, and scan data such as Image, Profile, or Surface data) available in a tool are listed in the **Outputs** section of the tool. To enable an output, check its checkbox in the tool's **Outputs** panel.



Enabling the Width measurement. Only the Height measurement is currently enabled, which is displayed in the tool's output section in the Tool Diagram.

For information on output filters, see *Output Filters* on page 228.

For information on decisions, see *Decisions* on page 229.

Arrays, Batching, and Aggregation

GoPxL supports data structures called arrays to help simplify otherwise complicated applications.

When working with arrays, you can select an array element (or all of the elements) across all arrayed tool outputs to make it easier to focus on a single array element. For more information, see *Displaying and Pinning Outputs* on page 94.

An array is a group of data items bundled into a single structure, such as multiple profiles or surfaces, multiple measurements, or multiple geometric features.

The items, or elements, in the array can be processed by other tools in one of two ways:

- Batching: Tools in batching mode process each array element *separately*. In other words, the tools "loop" through the elements of the array. Typically, tools in batching mode will output an array of outputs, where each element corresponds to the elements in the input. You enable batching by checking the **Enable batching** parameter in a tool's **Inputs** section. For more information, see *Batching* on page 204.
- Aggregating: Aggregating tools combine each element in the array and process it as a *single* piece of data. For more information, see *Aggregating* on page 206.

Most tools support the batching of array elements. Some tools support both batching and aggregating.

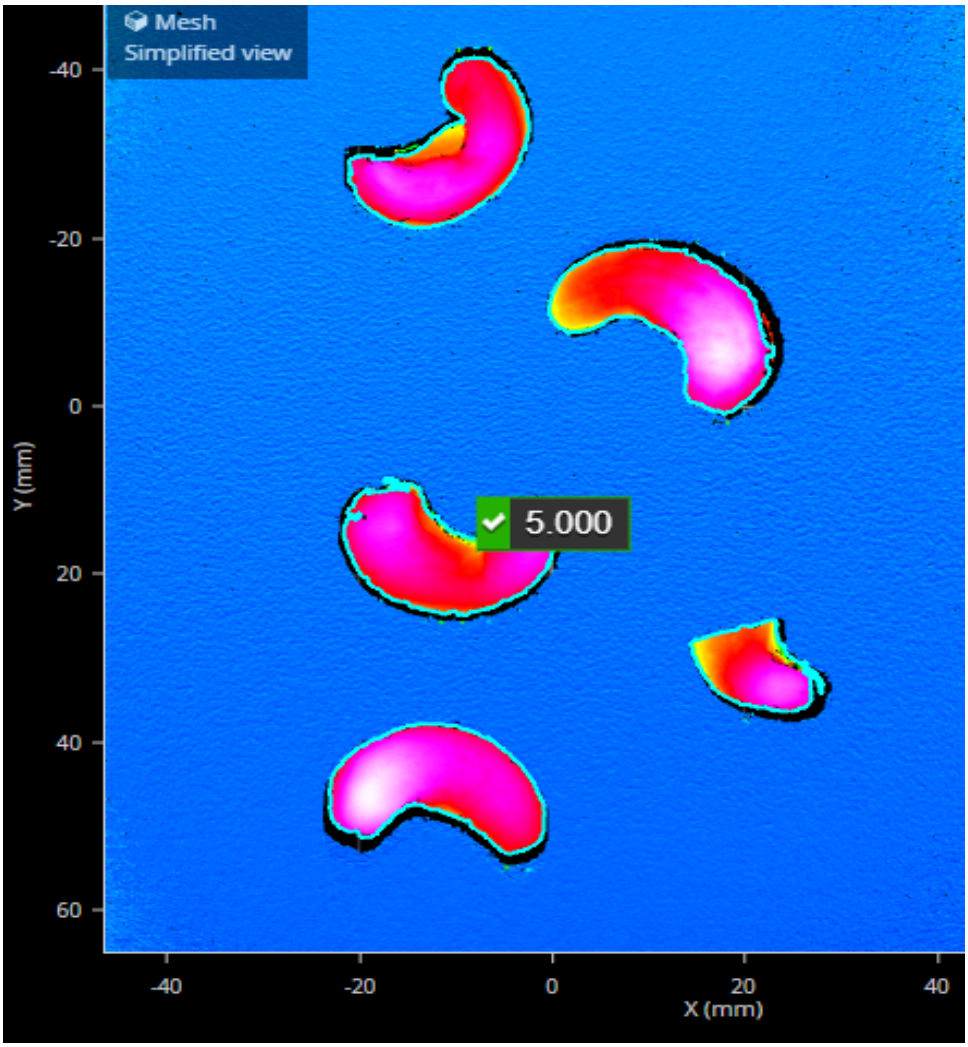
Array data can come from various sources:

- Outputs from tools like Surface Blob, Surface Segmentation, and Surface Pattern Matching. For more information, see *Arrays from Multi-instance Tools* below.
- G4 and G5 (single- and multi-sensor) and multi-layer output. For more information, see *Arrays from Multi-layer Output* on page 202.
- G2 multi-sensor output. For more information, see *G2 Multi-sensor Output* on page 203.
- The Script tool. For more information, see *Script Tool* on page 670.
- The Array Create tool. For more information, see *Array Create* on page 234.

For information on working with arrays in the Displayed Outputs panel, see *Displaying and Pinning Outputs* on page 94.

Arrays from Multi-instance Tools

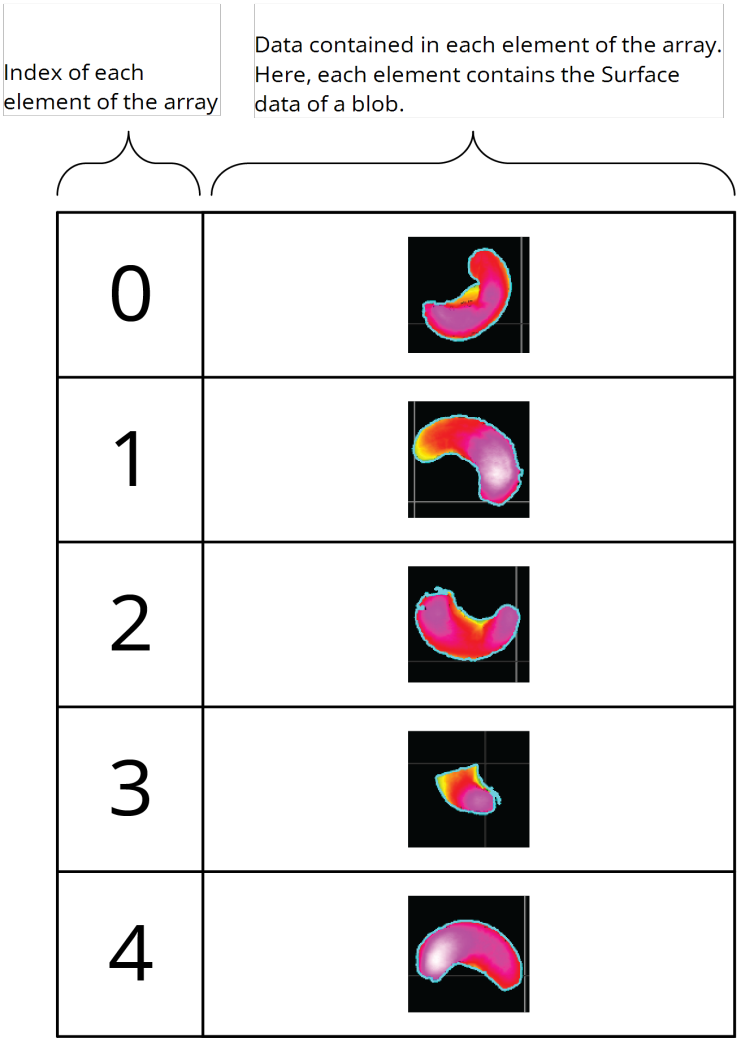
The Surface Blob, Surface Segmentation, and Surface Pattern Matching tools can identify multiple occurrences of features or parts in scan data. When you check **Enable Array Output** in the tool's **Parameters** section, the tool outputs an array containing the occurrences. Otherwise, the tools produce a separate output for each identified feature or part. In the following, a Surface Blob tool has identified five blobs in scan data.



Surface Blob tool identifying blobs in scan data.

When the tool's **Enable Array Output** setting is checked, the Surface Blob tool outputs an array containing all of the blobs it identifies.

An array containing objects identified in some scan data could be represented in the following way. (These are "blobs" from the Surface Blob tool.)



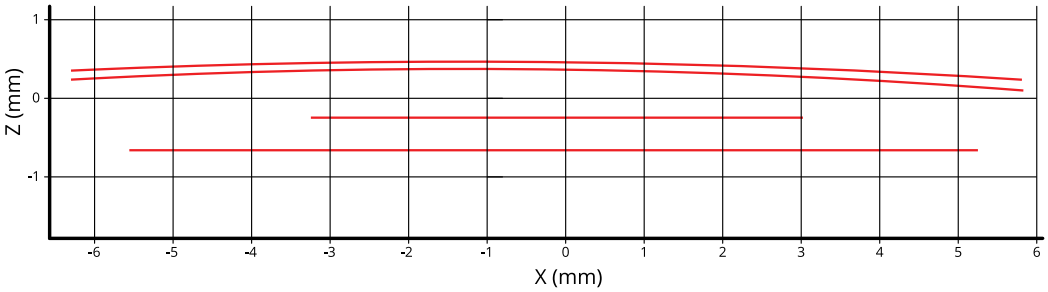
Note that array elements start at 0; that is, arrays are 0-based.

Other tools can then take this array as input. For more information, see *Batching* on page 204 or *Aggregating* on page 206.

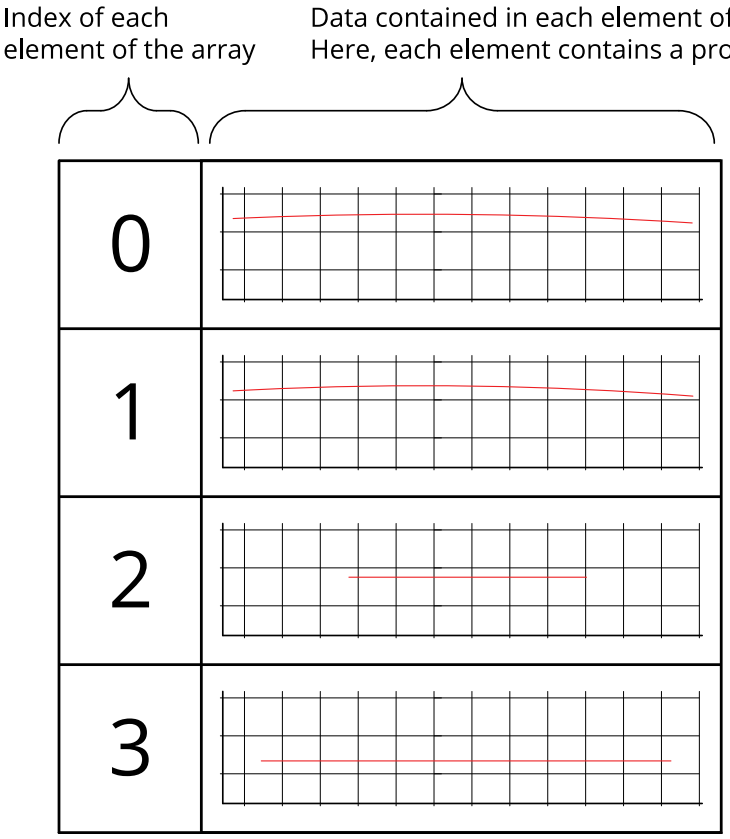
Arrays from Multi-layer Output

When configured for multi-layer scanning, confocal sensors can scan translucent or transparent targets and return a profile or a surface representing each layer. By default, these profiles are output as an array. (Note that you can optionally have GoPxL output the profiles individually by *enabling* **Separate layer outputs** on the **Scan** page; for more information, see *Scan Modes and Intensity* on page 139.)

The following illustration is a representation of multilayer Profile scan data from a confocal sensor, showing four profiles.



The structure of an array containing these profiles could be represented as follows:



Other tools can then take this array as input. For more information, see *Batching* on the next page or *Aggregating* on page 206.

G2 Multi-sensor Output

In G2 multi-sensor systems where sensors are in a top-bottom layout, Profile and Surface data can optionally be output as an array, where top data (from one or more sensors) is one array element, and bottom data (from one or more sensors) is another array element.

Other tools can then take this array as input. For more information, see *Batching* on the next page or *Aggregating* on page 206.

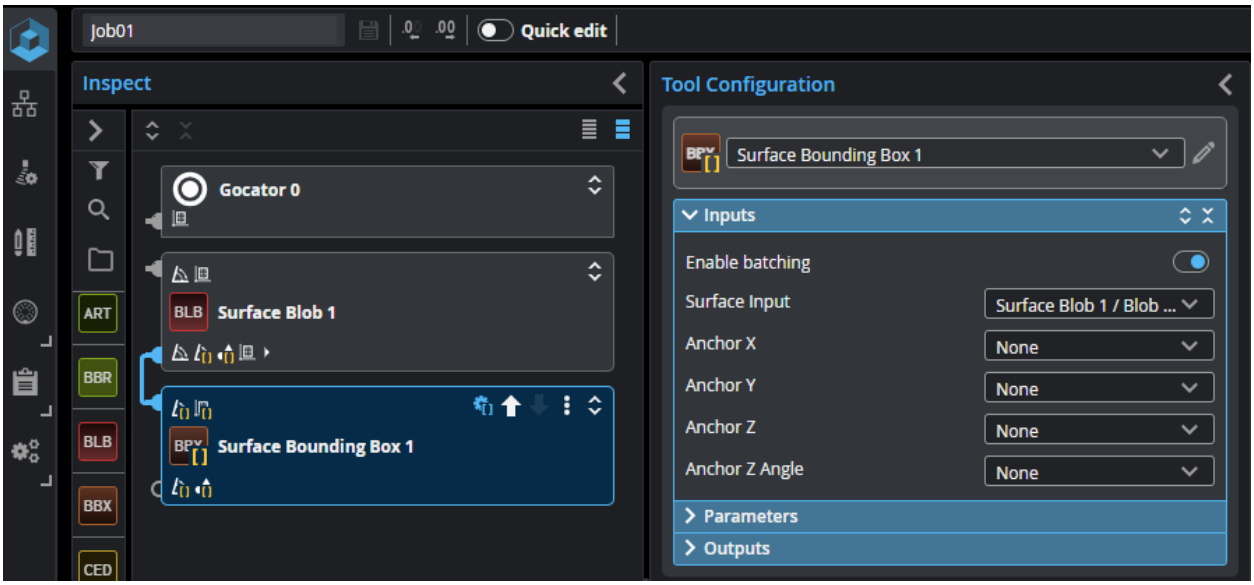
To understand how GoPxL communicates information related to batching and arrays, see *Batch and Array Indicators in GoPxL* on page 208.

Batching

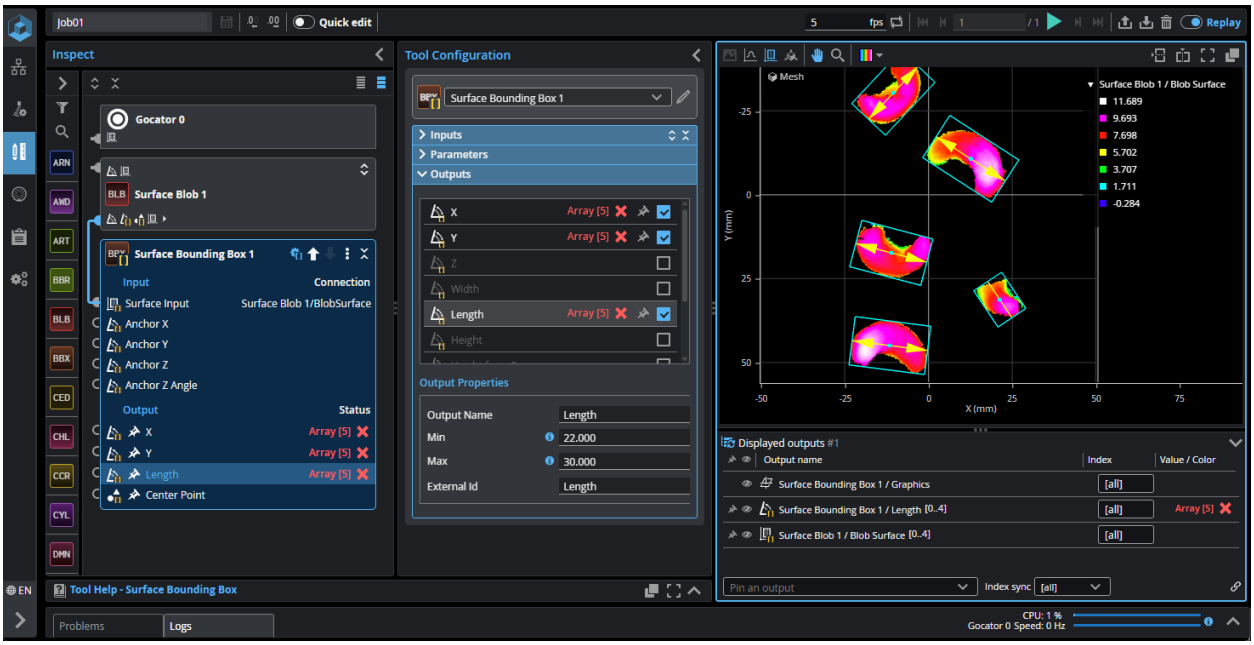
When a tool is in *batching* mode, it processes each item in the input array *individually*. A tool in batching mode also outputs one or more arrays (such as measurements, Profile or Surface data, geometric features, and so on), where each element of the output arrays corresponds to an element in the input array.

When a tool is in batching mode, there is no limit to the number of elements in the input array (other than any CPU limitations).

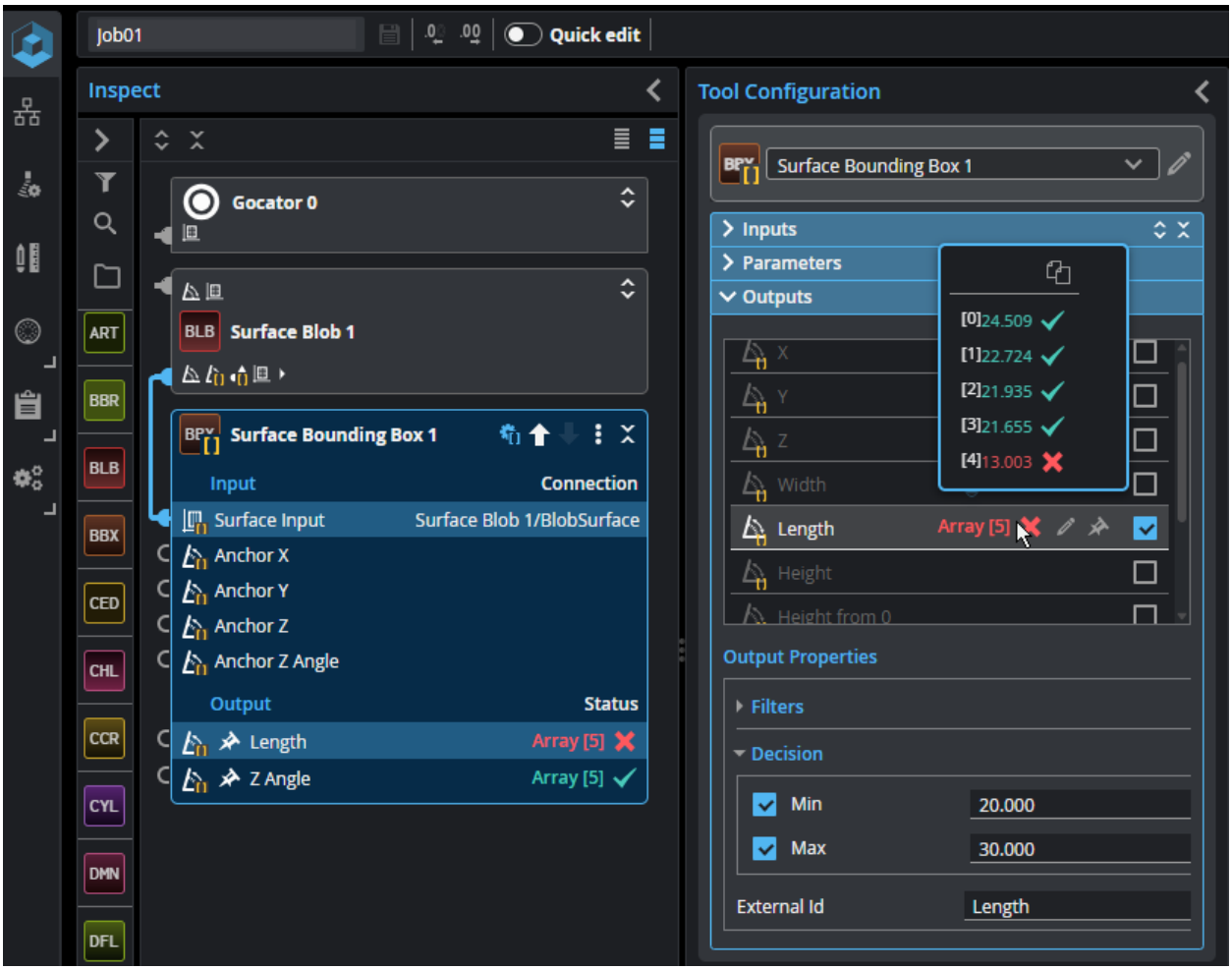
Batching is enabled in tools by checking **Enable batching** in the tool's **Inputs** section, or by clicking the Batching icon ( ) in a tool's block in the Tool Diagram panel (see below).



The following shows the Surface Bounding Box tool in batching mode (taking the Surface Blob tool's array of outputs as input). The tool calculates the bounding box for each element in the input array, and returns the enabled measurements as arrays. That is, it returns an X, Y, and Z measurement for each object.



Hovering over batched outputs in the Tool Diagram or in the **Tool Configuration** panel displays a list of the elements in the array.



When at least one result in the array is a fail, the output is considered a fail and the "Array [n]" label is red. Otherwise, it is green. Outputs within the pop-up list are displayed in green or red, depending on whether they represent pass or fail. You can copy the values of the measurement into the clipboard by clicking the copy icon ( ). The format of the copied values is as follows:

```
[24.509,22.724,21.935,21.655,13.003]
```

Aggregating

Some tools can take arrays as an input and combine (or "aggregate") the data in the array elements and treat it as a single input (see below for an example). These tools are called "aggregating tools"; see *Aggregating Tools* on the next page for a list of the aggregating tools. In order for a tool to aggregate an array input, its **Enable Batching** parameter must be disabled. Otherwise, it will process the array elements individually. Note that aggregating tools also take non-array input (such as a single scan of Profile or Surface data), and simply process the data as usual.

Some aggregating tools are limited to a maximum of a 2-element array in their input. Other tools can accept an arbitrary number. See *Aggregating Tools* on the next page to determine the limits of a tool. Note that when a tool is in *batching* mode, there is no limit to the number of elements in the input

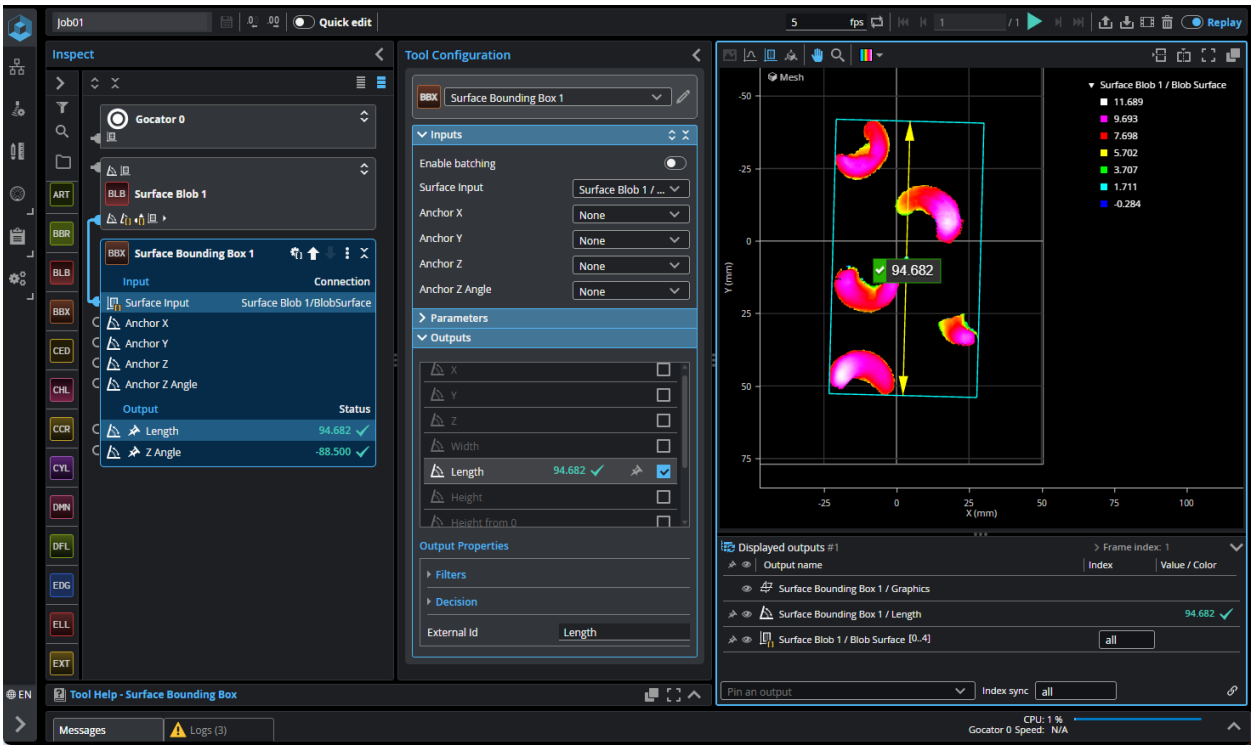
array (other than any CPU limitations), regardless of the array size limitations when the tools is aggregating data.

> Remember to uncheck **Enable Batching** if you want a tool to aggregate array inputs.

Aggregating Tools

	Number of array elements supported for aggregation	
Tool	**2 elements maximum**	**arbitrary (2 or more elements)**
Profile Circle Radii		✓
Profile Closed Area		✓
Profile Bounding Box		✓
Profile Bridge Value		✓
Profile Circle		✓
Profile Dimension		✓
Profile Ellipse		✓
Profile Groove	✓	
Profile Intersect		✓
Profile Line		✓
Profile Line Intersect		✓
Profile Panel	✓	
Profile Position		✓
Profile Round Corner	✓	
Profile Strip		✓
Surface Bounding Box		✓
Surface Dimension		✓
Surface Ellipse		✓
Surface Flatness		✓
Surface Plane		✓
Surface Position		✓
Surface Section		✓
Surface Sphere		✓
Surface Stick		✓

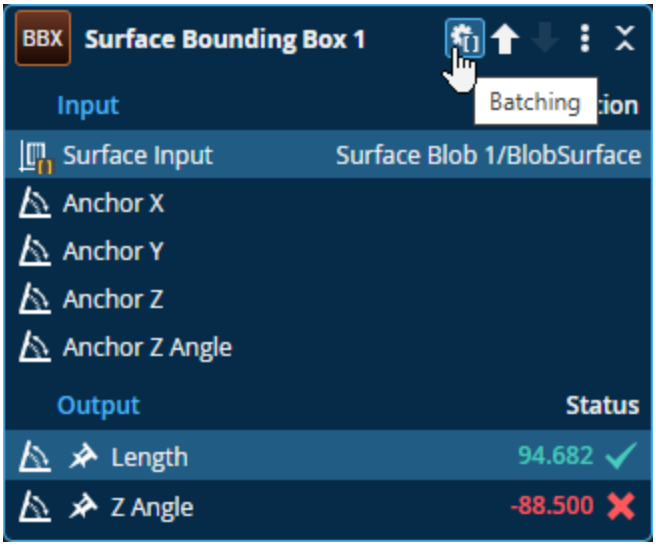
Taking the example from above, if batching in the Surface Bounding Box tool is disabled, it will aggregate the scan data in the input array, and fit the bounding around the scan data of all the elements together.



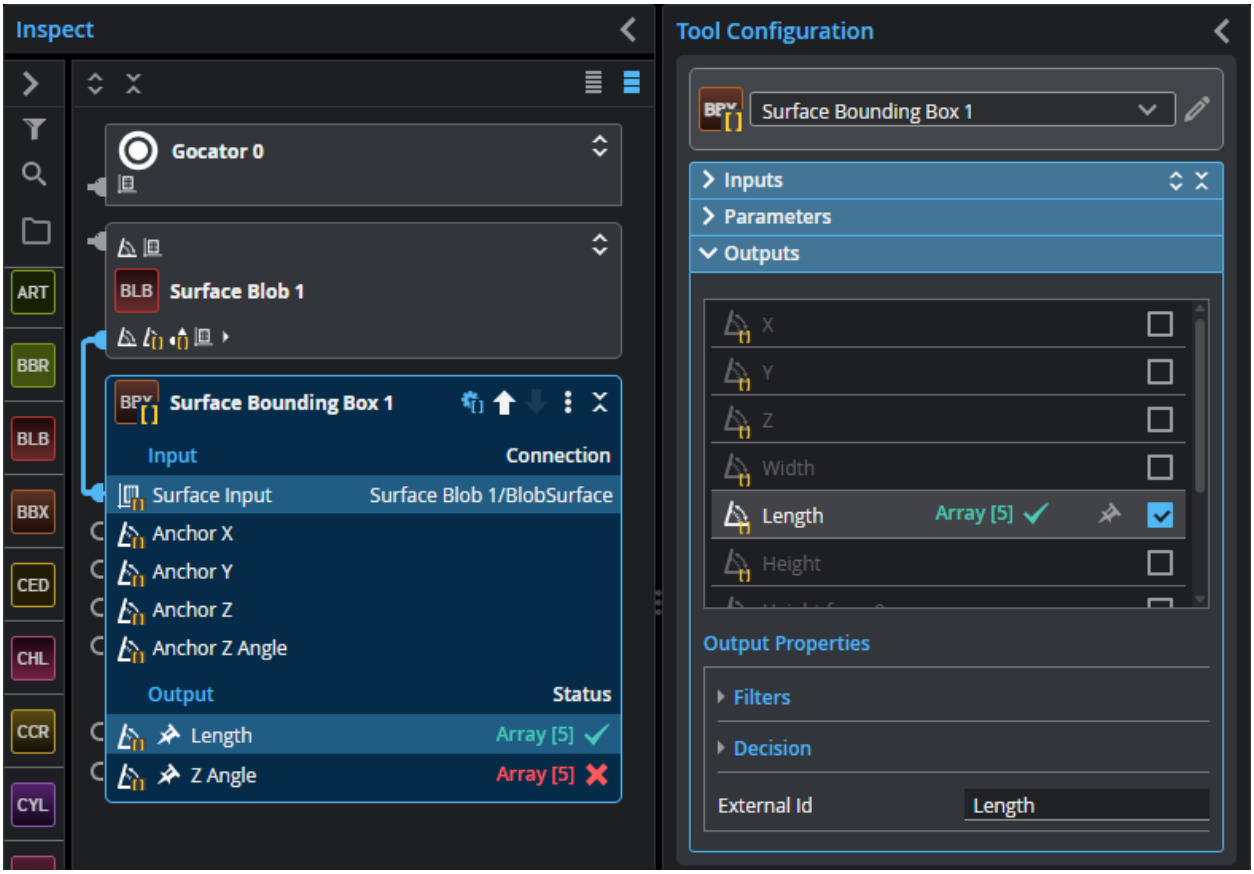
Batch and Array Indicators in GoPxL

GoPxL communicates information related to batching and arrays by adding a "square bracket" badge ("[ ]") to input, output, and tool icons in the web interface.

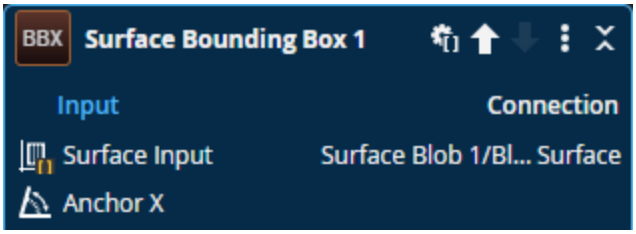
In the Tool Diagram panel, for tools that support batching, a Batching icon ( ) is displayed at the top of the tool block. Use this to toggle batching for that tool, or toggle **Enable batching** in the **Tool Configuration** panel. If the tool is not currently selected (that is, you have clicked on it), indicated by a blue border, you must hover the mouse pointer over the top area of the tool block to see whether it supports batching.



When you enable batching on a tool, the Batching toggle icon turns blue, and GoPxL places yellow batching badges on various elements in the web interface. Below, the tool is in batching mode, and its inputs and outputs are therefore all arrays.



For tools that can take array inputs when batching is not enabled, the scan data input (Surface Input) shows a batching badge. Tools like this aggregate array inputs in this state; for more information, see *Aggregating* on page 206.



Measurement Anchoring

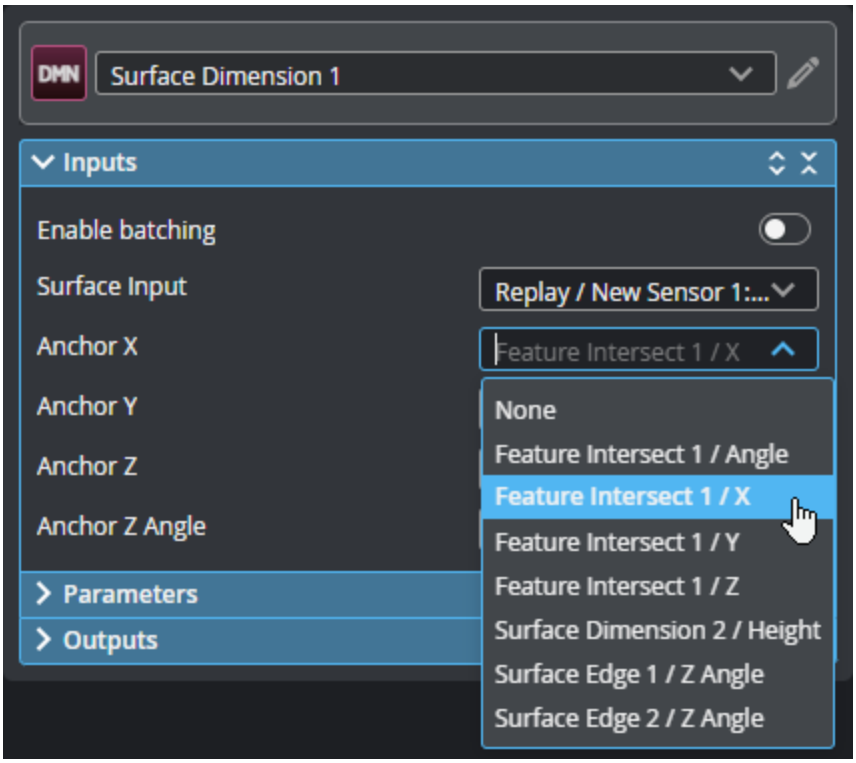
The position of parts moving on a transport mechanism such as a conveyor typically changes from part to part in one or both of the following ways:

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

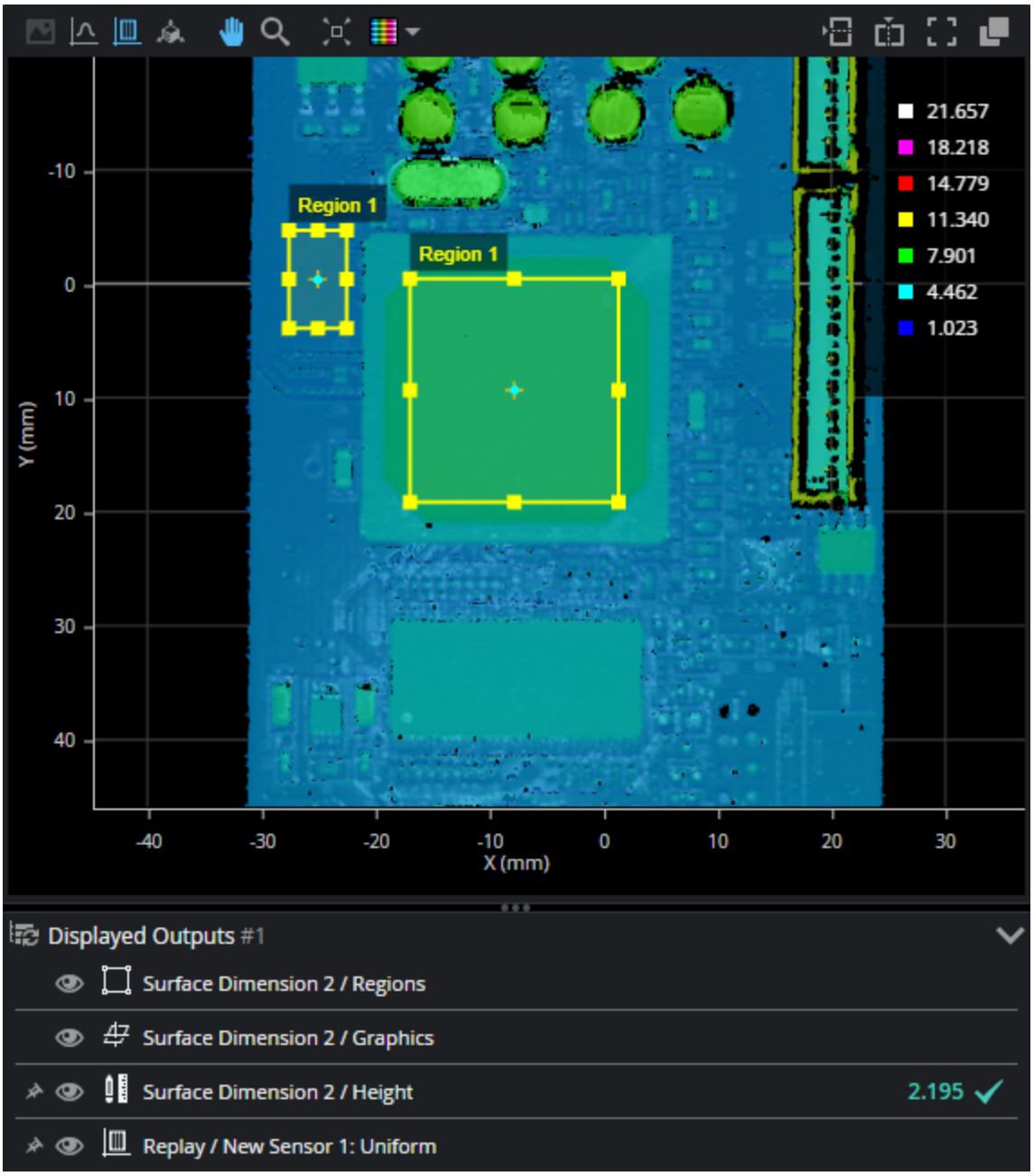
When the position and angle variation between parts is minor—for example, when scanning electronic parts in trays—you can anchor one tool to one or more measurements from one or more other tools to compensate for these minor shifts. As a result, GoPxL can correctly place the anchored tool's measurement 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 often use the Surface Pattern Matching tool to compensate. For more information, see *Surface Pattern Matching* on page 913.

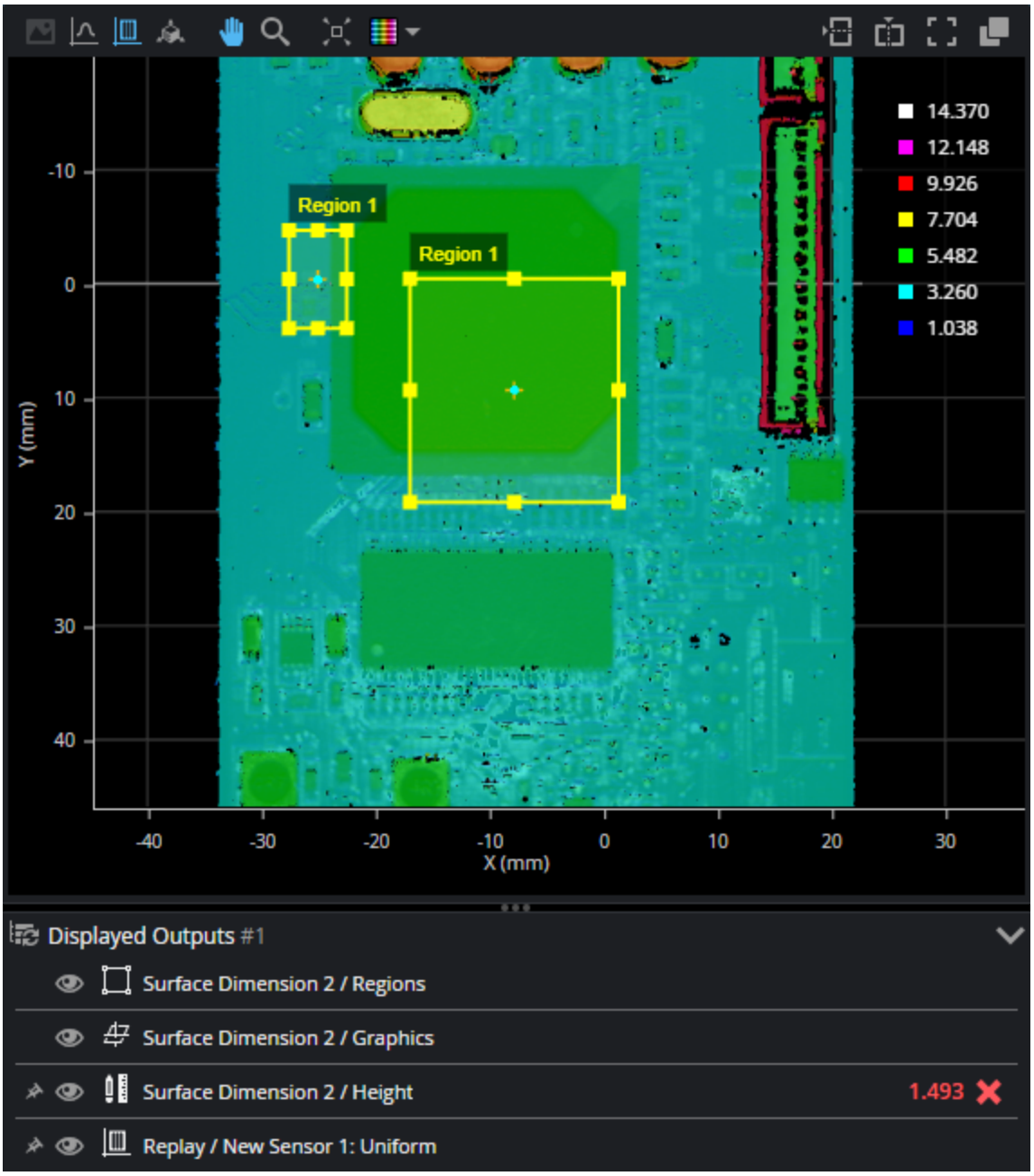
You set a tool's anchors in its **Inputs** section, by setting the desired anchor (such as **Anchor X** or **Anchor Y**) to the appropriate measurement from another tool:



For example, the following shows a surface scan of a PCB. A Surface Dimension height measurement returns the height of a component in the measurement region relative to a nearby reference region; the height is between the measurement's thresholds and is a pass (the green value in the **Displayed Outputs** pane).

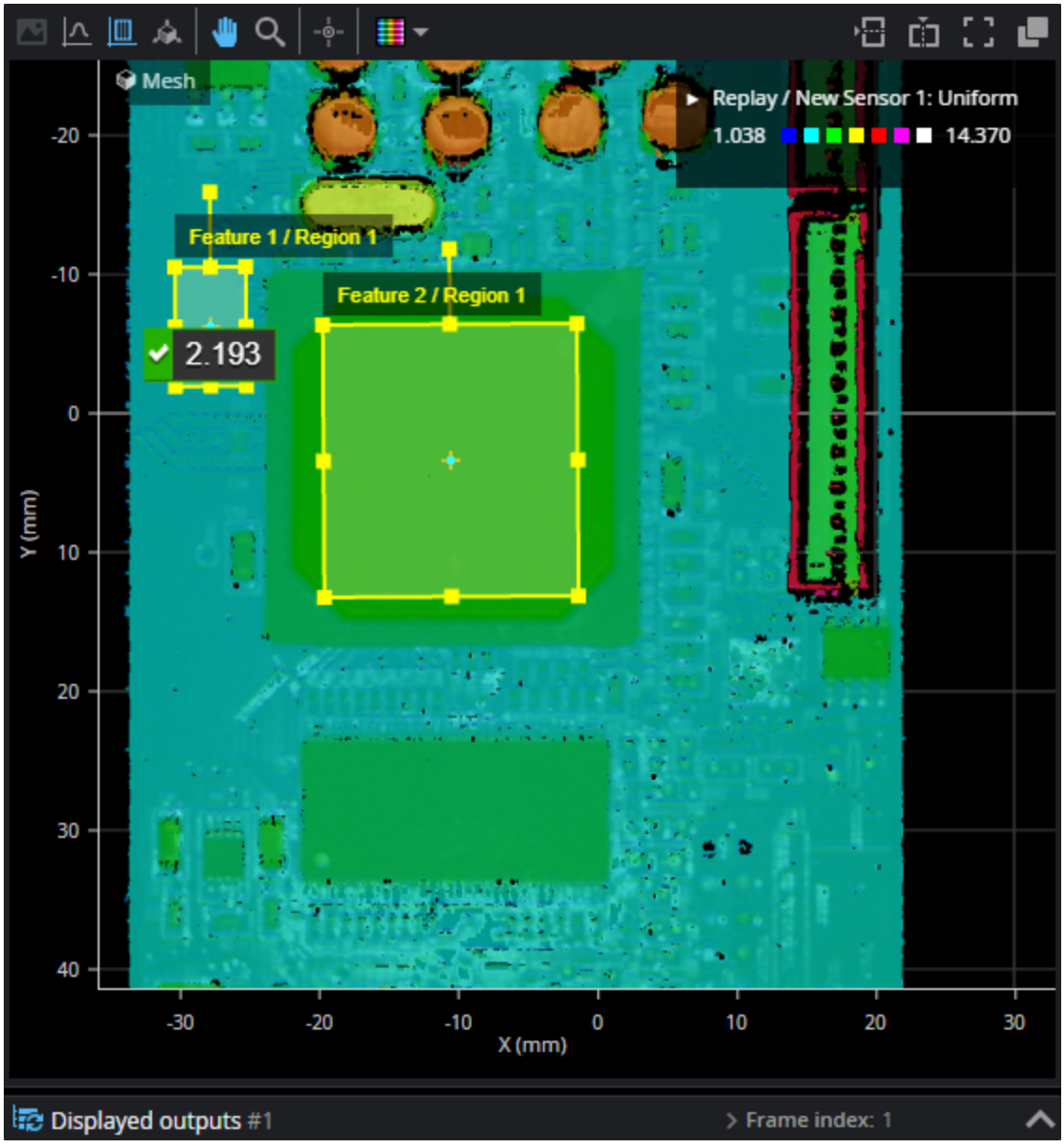


In the following scans, the part has shifted, but the measurement regions remain where they were originally configured, in relation to the sensor or system coordinate system, so the measurement returned is incorrect, and the measurement fails. A measurement could also return a pass decision when it should not.

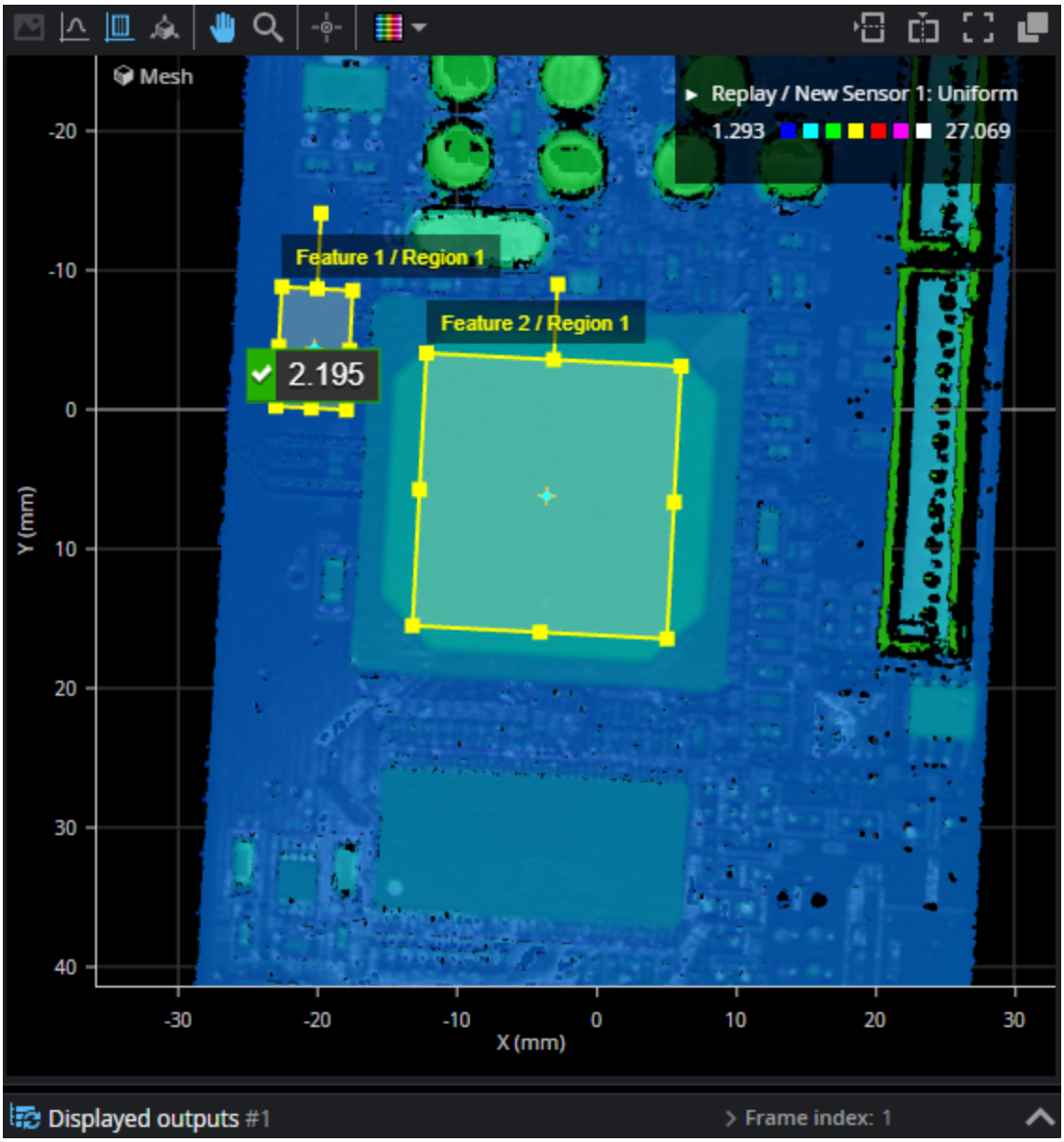


When you set any of a tool's anchor sources, an offset is calculated between the anchored tool and the anchor source. This offset is used for each frame of scanned data: the anchored tool's measurement region is placed in relation to the anchor source, at the calculated offset.

In the following image, the Surface Dimension tool is anchored to the X and Y measurements from two Surface Edge tools (placed over the left and bottom edges of the PCB), and GoPxL compensates for the shift—mostly along the X and the Y axis in this case—and returns a correct measurement, despite the shift.



You can combine the positional anchors with an angle anchor (such as 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 compensates for the rotation, and the anchored tool returns a correct measurement.



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

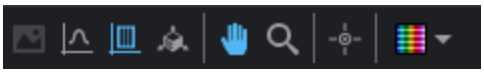
If angle anchoring is used without X or Y anchoring, the tool's measurement region rotates around its *center*. If only one of X or Y is used, the region is rotated around its center and then shifted by the X or Y offset.

Regions

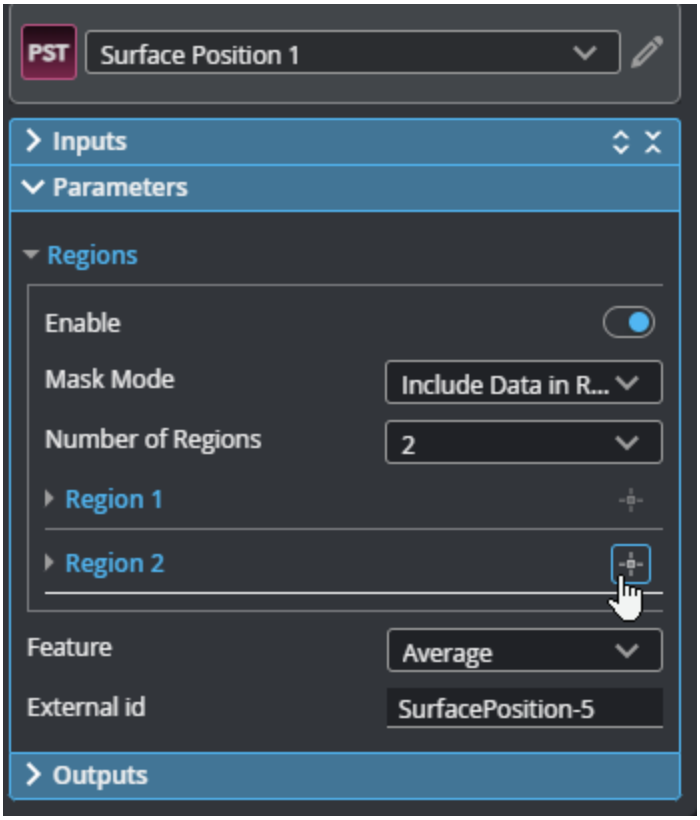
Most measurement tools use user-defined regions, or regions of interest, to limit the area in which measurements occur. Unlike reducing the active area, reducing the region of interest does not increase the maximum frame rate of the sensor.

You can use Ctrl-Z and Ctrl-Shift-Z to undo and redo changes to the last changed region's size and position.

When the data viewer is in Surface / Top or Profile / Front mode (not in Perspective mode), you can quickly center all of a tool's regions in the data viewer and reset them to their default size by clicking the "Center tool regions" button ( ) above the data viewer; this is useful if you have zoomed in or out a lot.



You can also center and resize a specific region by clicking its "Center tool region" button in the tool's **Parameters** section, next to the region expander.



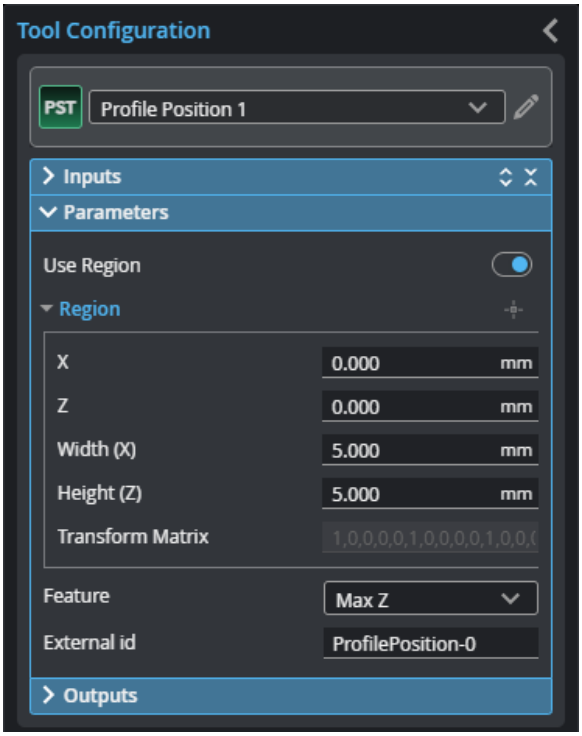
When a tool has multiple regions, when you center the regions, they are resized and arranged in a row in the center of the data viewer.

Some tools provide "flexible" regions, which in addition to rectangular regions let you create circular and elliptical regions (which can optionally be annular). Some tools also let you use Surface and Surface Intensity data as masks. For information on setting flexible regions, see *Flexible Regions* on page 217.

Other tools are currently limited to standard, rectangular regions.

Standard Regions

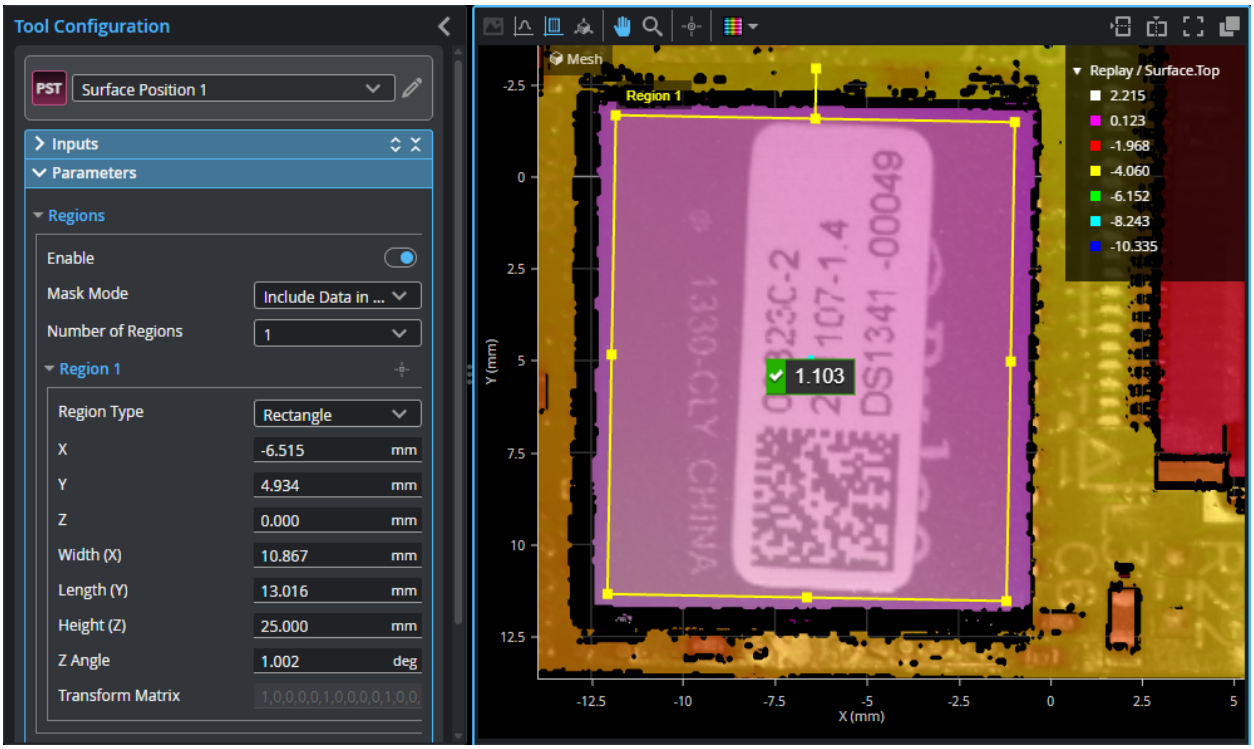
Some tools provide regions limited to boxes/rectangles.



The **Transform Matrix** setting is not currently used.

Flexible Regions

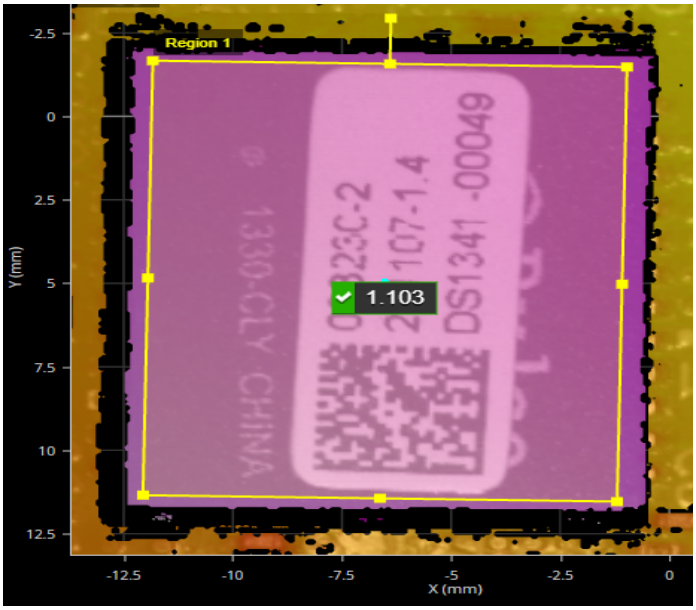
Many tools provide "flexible" regions, which let you choose region types such as circle and ellipse, in addition to rectangle regions. For a complete list and information on the related settings, see *Flexible Region Parameters* below.

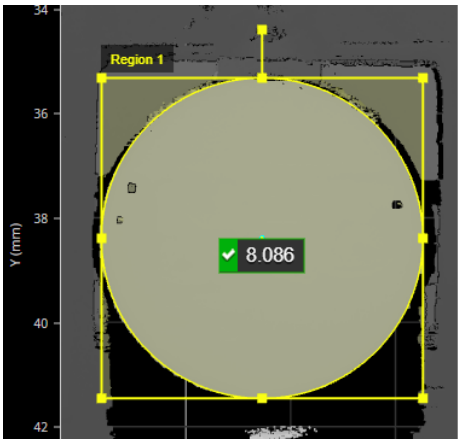
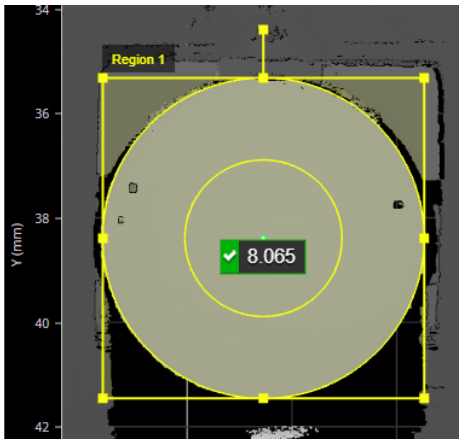
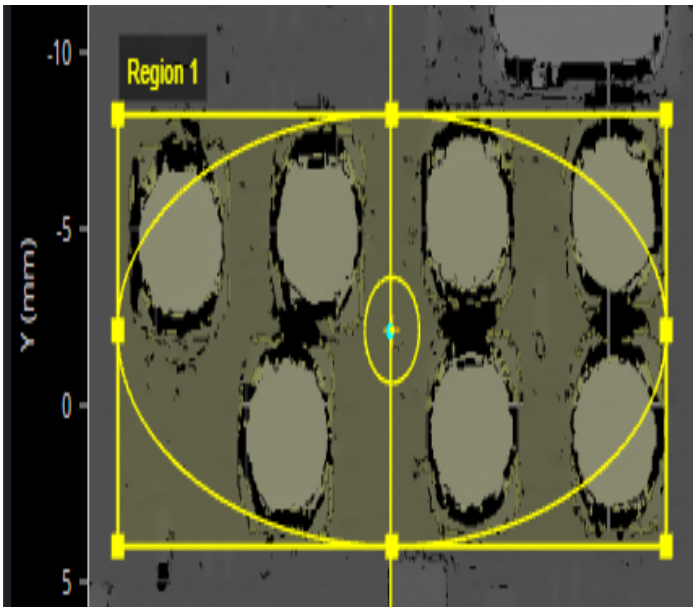


The following parameters are available in tools that support flexible regions.

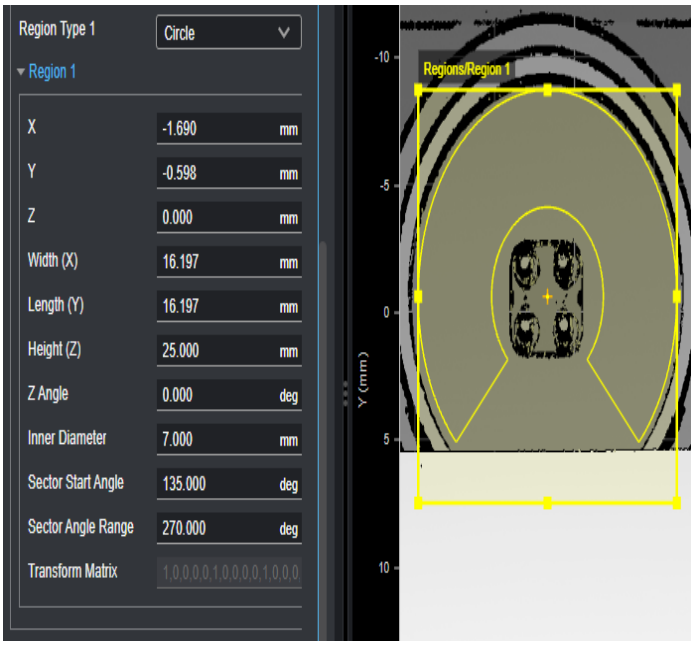
Flexible Region Parameters

Parameter	Description
Mask Mode	One of the following: **Include Data in Region**: The tool uses only the data inside the region or regions. **Exclude Data in Region**: The tool ignores the data inside the region or regions, using only the data outside.
Number of Regions	The number of regions or masks the tool uses to extract data. You can define up to 64 regions. This parameter is not available in some tools. To set a number of regions greater than 16, choose "Custom" in this parameter and set **Custom Region Count**. When you specify more than one region, the regions are initially stacked on top of one another, in the same location.
Custom Region Count	The number of regions (from 17 to 64) when **Number of Regions** is set to "Custom".

Parameter	Description
Region {n}	An expandable section containing region definitions for each region.
Region Type {n}	The type of each region. Regions can overlap. (For more information on the settings you use with the Circle and Ellipse types, see *Working with Circular and Elliptical Regions* on page 221.) One of the following: **Rectangle** Extracts a rectangular region from the acquired data.  **Circle** Extracts a circular region from the acquired data, constrained by a square region. Set the region's inner circle using the **Inner Diameter** parameter to 0 to extract circular data. Set it to a non-zero to extract annular data.

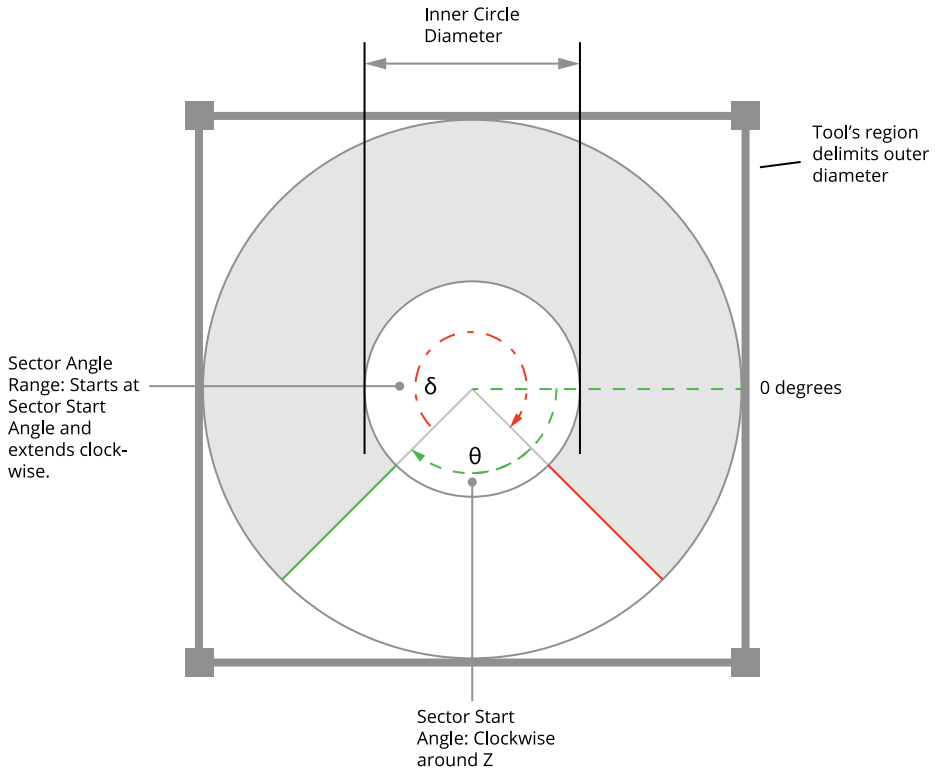
Parameter	Description
	  Use the **Sector Start Angle** and **Sector Angle Range** parameters to extract a partial circular or elliptical region. (See below.) **Ellipse** Extracts an elliptical region from the acquired data, constrained by a square or rectangular region. Set the region's inner ellipse (inner cyan ellipse below) using the **Inner Major Axis** and **Inner 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. (See below.) **Surface** Uses the Surface data you select in **Mask Source {n}**, in the **Inputs** panel, to

Parameter	Description
	create a mask. **Surface Intensity** Uses the intensity data you select in **Mask Source {n}**, in the **Inputs** panel, to create a mask. Set the **Low Threshold** and **High Threshold** parameters as required.
X, Y, Z Width (X), Length (Y), Height (Z) Z Angle	These parameters define the size, location (center of the region), and orientation of the region. Not available when **Region Type {n}** is set to the following: **Surface** or **Surface Intensity**.
Inner 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 Major Axis Inner 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.
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.
High Threshold Low Threshold	The high and low thresholds the tool uses in combination with the intensity mask. Only available when **Region Type {n}** is set to the following: **Surface Intensity**.

Working with Circular and Elliptical Regions

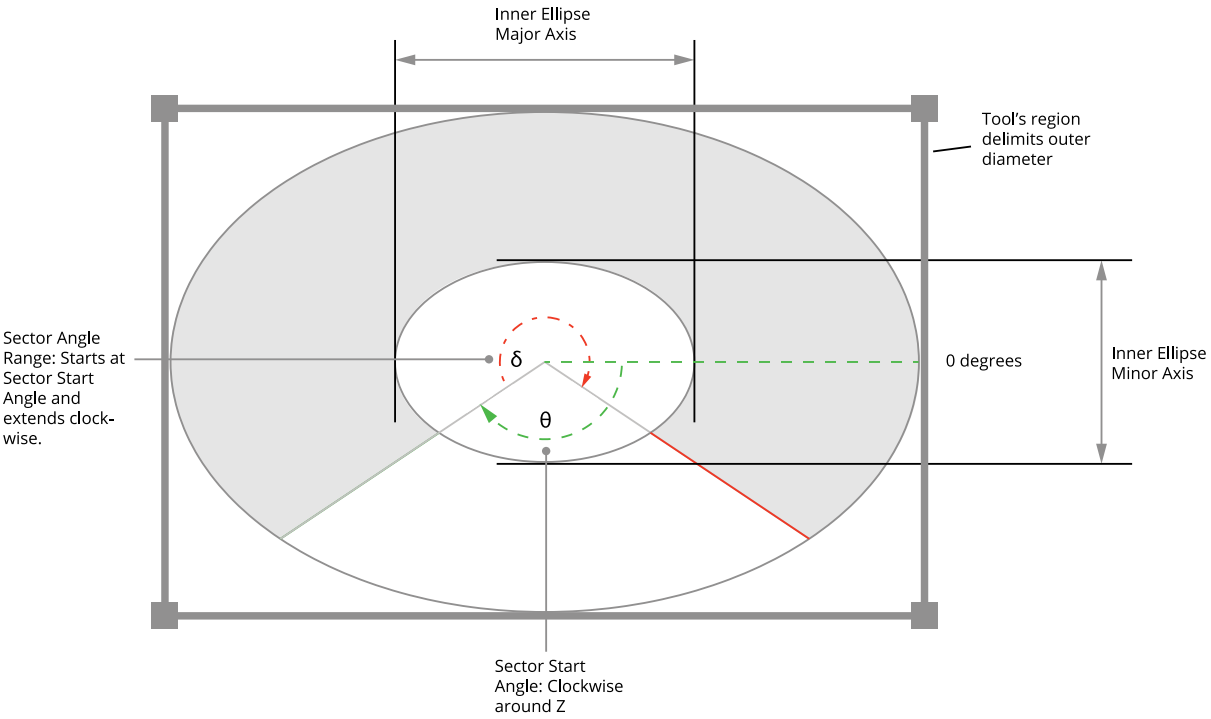
When you set a region's type to Circle or Ellipse, the tool displays several additional settings that work together to define the region. **Sector Start Angle** and **Sector Angle Range** work together to define the start and end of a partial circular/elliptical region (solid or annular). A region will 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.

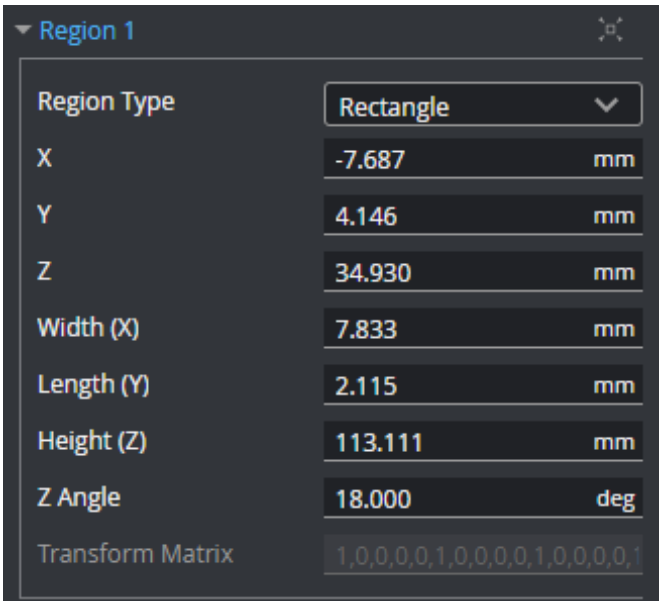
Note that the angles defining a partial circular/elliptical region are relative to the *region*, and not the sensor's coordinate system. So a region rotated 30 degrees using its **Z Angle** setting rotates the start angle and angle range by 30 degrees.

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.

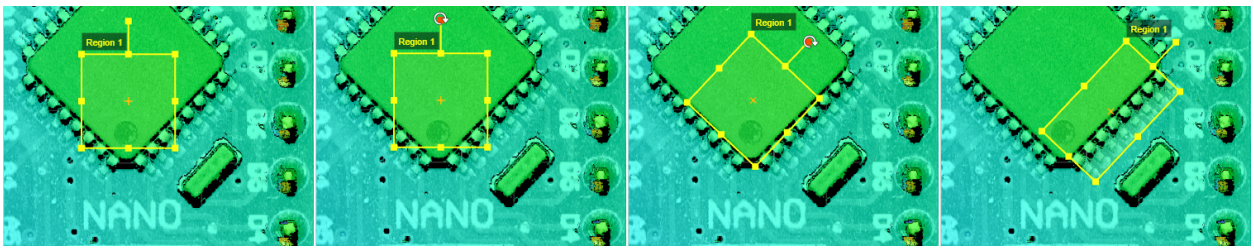


Region Rotation

The regions of some tools can be rotated by setting the region's **Z Angle** (or **Angle** for some models) to better accommodate angled features on a target. By rotating the measurement region, data not related to the feature can often be excluded, improving accuracy of measurements. You can modify the size and position of a rotated region.

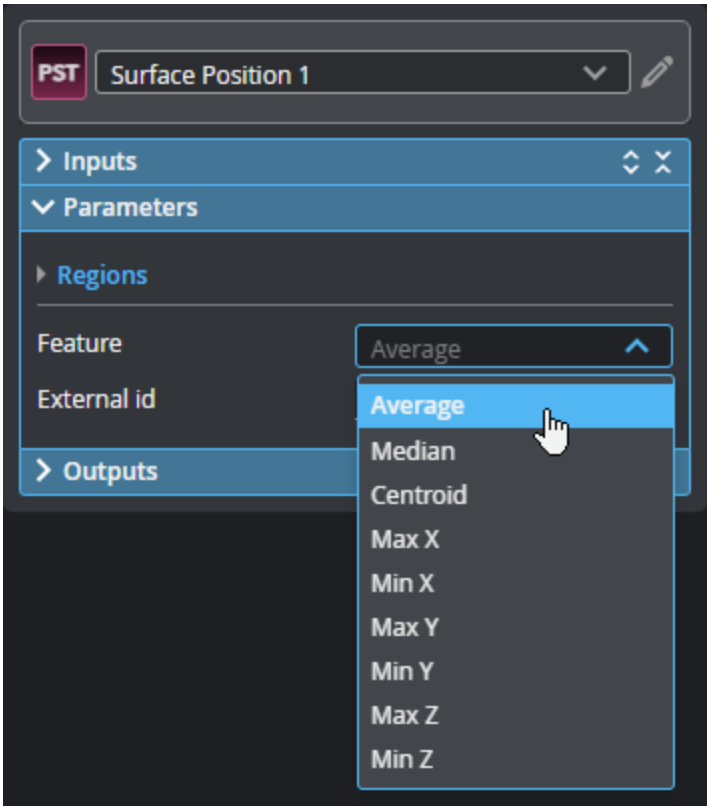


You can also rotate a region using a mouse in the data viewer. To do this, hover the mouse over the region's grabber and click and hold the left mouse button. Then rotate the region as needed. Finally, resize the region.



Feature Points

Dimensional and positional measurements detect *feature points* found within the defined region (whether it's a measurement region, feature region, reference region, and so on) and then compare measurement values taken at the selected point with minimum and maximum thresholds to produce a pass / fail decision. Feature points are selected in one or more **Feature** parameters in a tool's **Parameters** section, and are used for all of the tool's measurements.

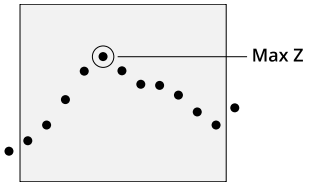
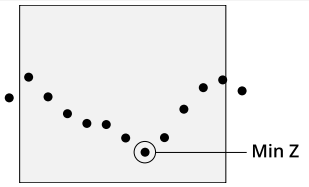
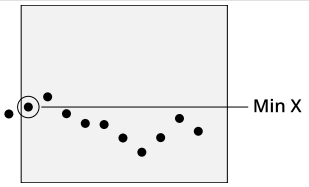
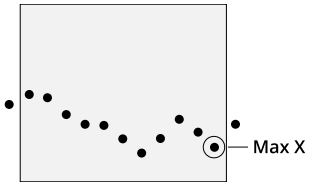
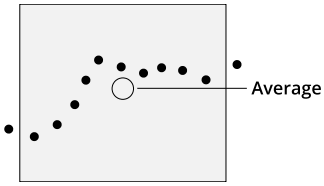
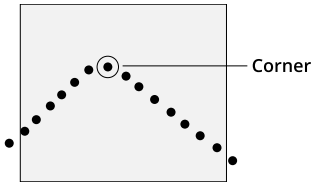


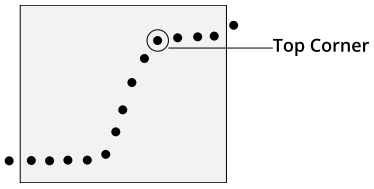
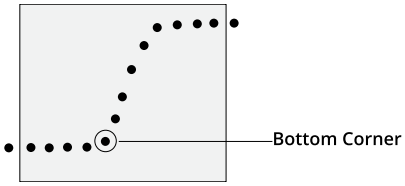
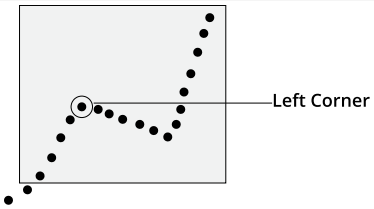
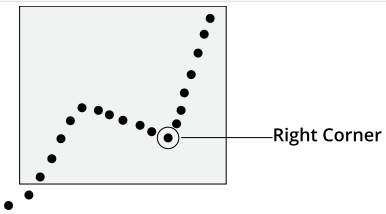
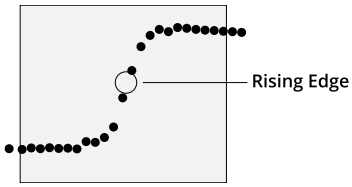
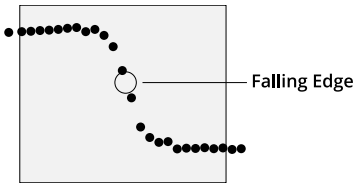
It is important to note that the feature points described here are distinct from the geometric features generated by some tools, and which can be taken as inputs by other tools. Feature points are simply the location in the data at which a measurement tool performs a measurement. So, for example, if you set a Profile Position tool's **Feature** parameter to Max Z, it will return the X and Z positions of the point that is at the highest Z value. Geometric features are a data type produced as

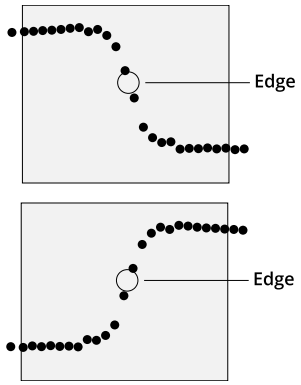
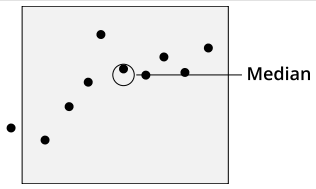
output or taken as input by a tool. (For more information on geometric features, see *Geometric Features* on page 227.)

The following types of points can be identified in a measurement region. The availability of a feature point type is tool-dependent.

> The following are represented as profiles for clarity. Many Surface tools use feature points.

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

Point Type	Examples
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.	
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).	

Point Type	Examples
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 Tools* on page 252.)

The measurement tools can currently generate the following kinds of geometric features:

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

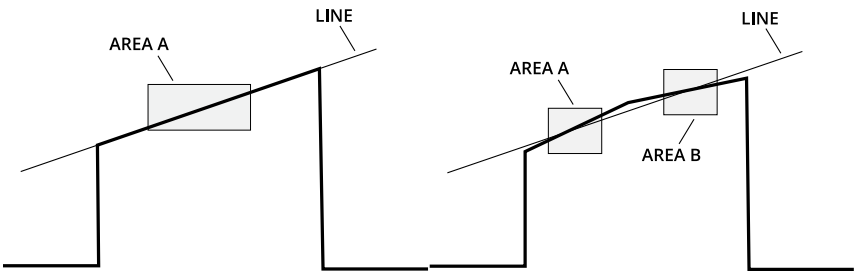
Lines: A straight line that is infinitely long. Useful for locating the orientation of an enclosure or part, or to intersect with another line to form a reference point that another tool can take as input.

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.

Fit Lines

Some Profile measurement tools estimate line fits in order to measure angles or intersection points. A fit line can be calculated using data from either one or two fit areas.

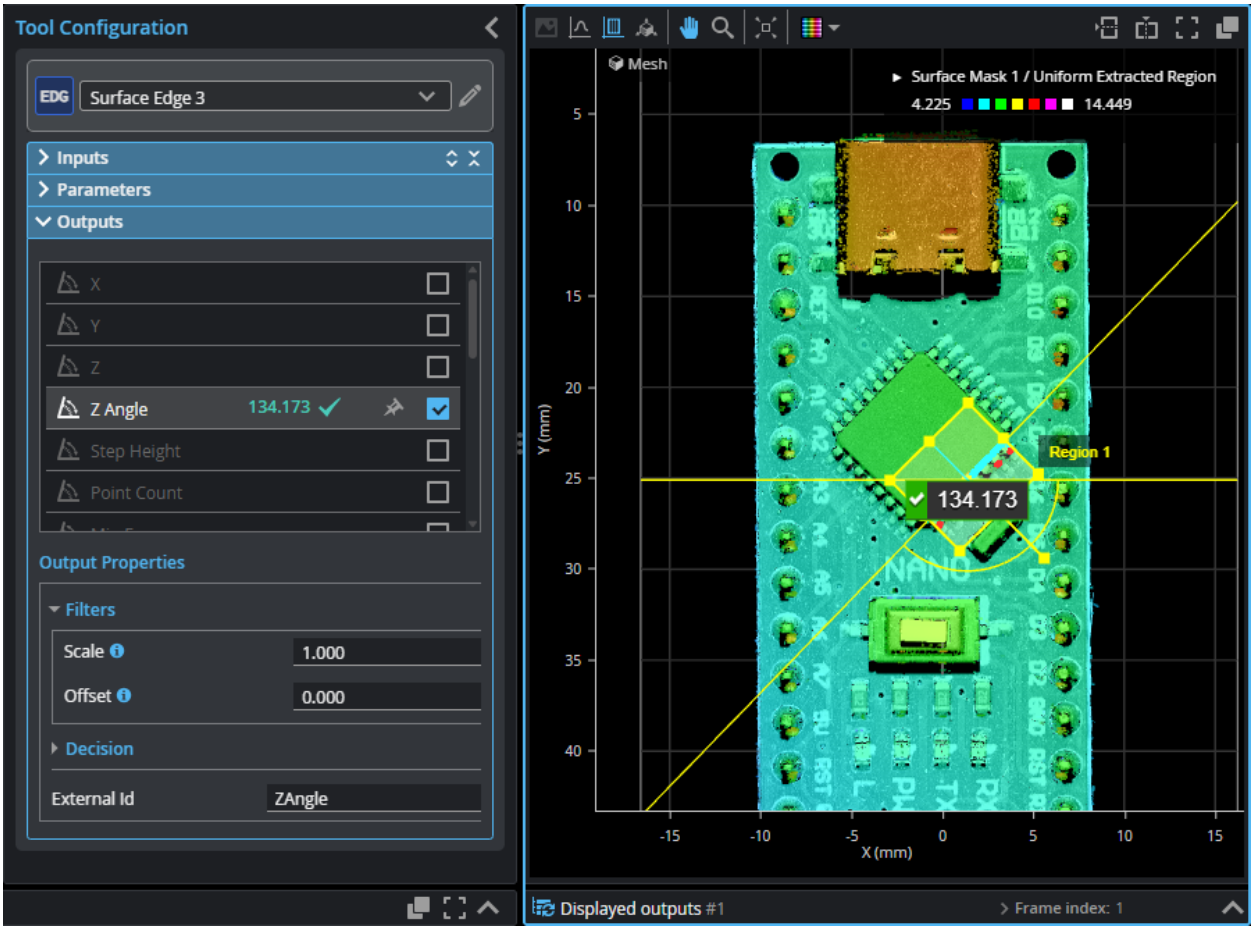


A line can be defined using one or two areas. Two areas can be used to bypass discontinuity in a line segment.

Output Filters

Results from a measurement can be scaled or offset before they are output from the GoPxL. You set these values in a tool's **Output Properties** > **Filters** section.

Note that measurement values are output with six decimal places. However, you can configure the GoPxL interface to hide some decimal places, using the "Decrease precision" and "Increase precision" buttons (and , respectively), from zero to six decimal places. Keep this in mind when configuring thresholds.



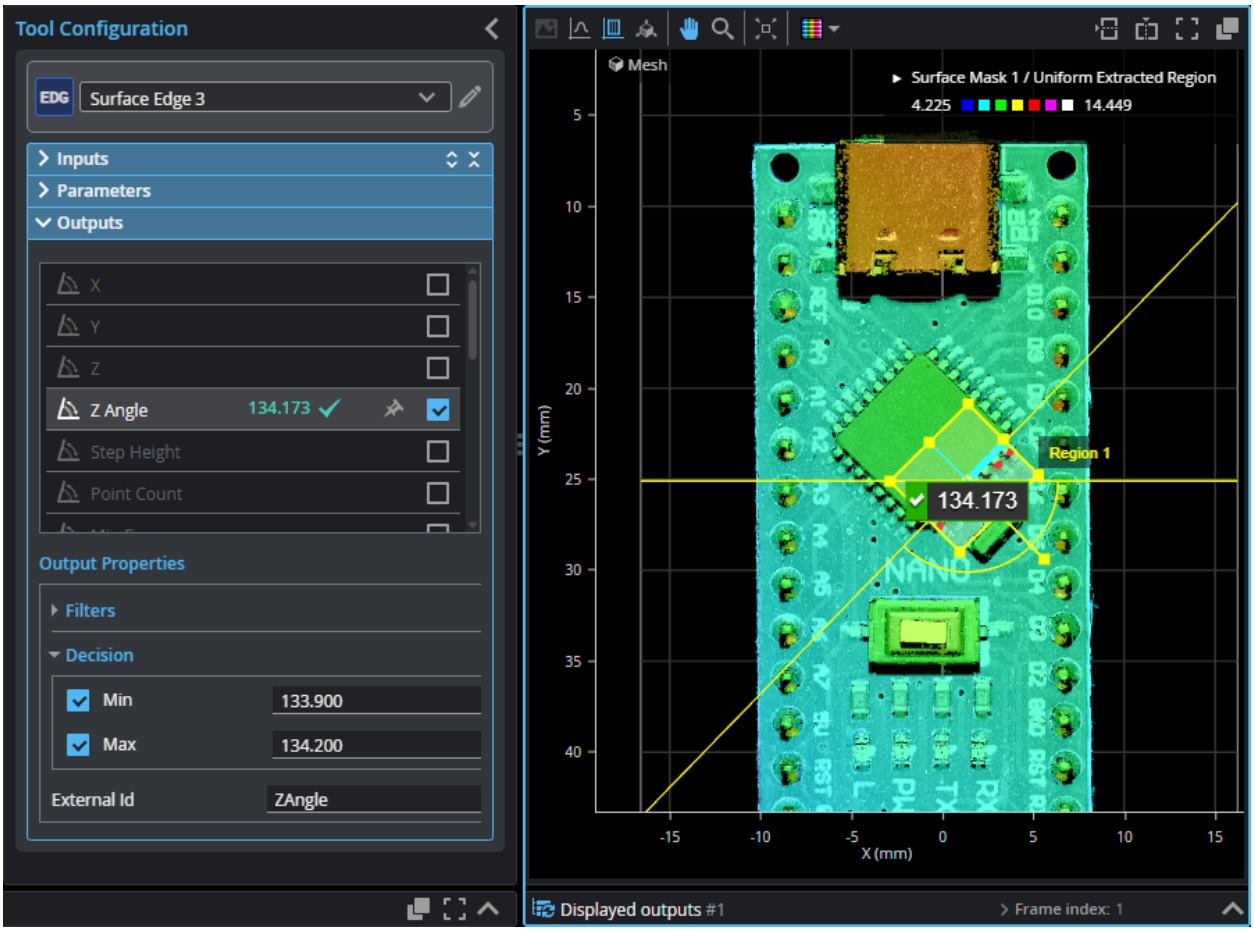
Filter	Description
Scale and Offset	The **Scale** and **Offset** settings are applied to a measurement value according to the following formula: Scale * Value + Offset **Scale** and **Offset** can be used to transform the output without the need to write a script. For example, to convert the measurement value from millimeters to thousands of an inch, set **Scale** to 39.37. To convert from radius to diameter, set **Scale** to 2.

Decisions

Results from a measurement can be compared to minimum and maximum thresholds you set in a tool's **Output Properties** > **Decision** section to generate *pass* / *fail* decisions. The decision state is *pass* if a measurement value is within the minimum / maximum range. Output values that pass are green in the interface. Output values that fail are red in the interface.

By default, **Min** and **Max** are disabled in new tool instances, and are displayed as passes (green) in the interface.

Note that measurement values are output with six decimal places. However, you can configure the GoPxL interface to hide some decimal places, using the "Decrease precision" and "Increase precision" buttons (and , respectively), from zero to six decimal places. Keep this in mind when configuring thresholds.



Decisions (and measurements) can be sent to external programs and devices. For more information on sending values and decisions as output, see *Communicate - PLC, Digital, GDP, HMI* on page 1018.

External AI Models

Although you can train models entirely within GoPxL, you can also do the following:

- Import and use compatible Ultralytics models, including YOLO 26, developed outside the GoPxL training environment.
- Develop and train models using Roboflow or other AI frameworks and the PC/GoMax infrastructure. For instructions on using Roboflow, see *Roboflow* on the next page.

Roboflow

Array Tools

Array tools let you create and work with arrays, or extract individual elements from an array.

> Arrays can contain any type of data supported in GoPxL (such as Surface, Profile, or Image data, or measurements). For more information on this and on arrays in general, see *Arrays, Batching, and Aggregation* on page 199.

The following Array tools are available:

- Array Decision: Takes an *array* of *measurements* and lets you apply different minimum/maximum thresholds to each measurement in the array.

  The tool is typically used in electronics pin inspection applications.

  For more information on this tool, see *Array Decision* on page 236.

- Array Index: Takes an *array* of data as input and extracts one or more pieces of data at the user-provided index. You can then perform measurements on the individual outputs, or perform measurements using more than one piece of data, such as measuring the thickness represented by two profiles or surfaces.

  Usage examples include separating the multiple layers produced by a confocal sensor into individual profiles or surfaces.

  For more information on this tool, see *Array Index* on page 240.

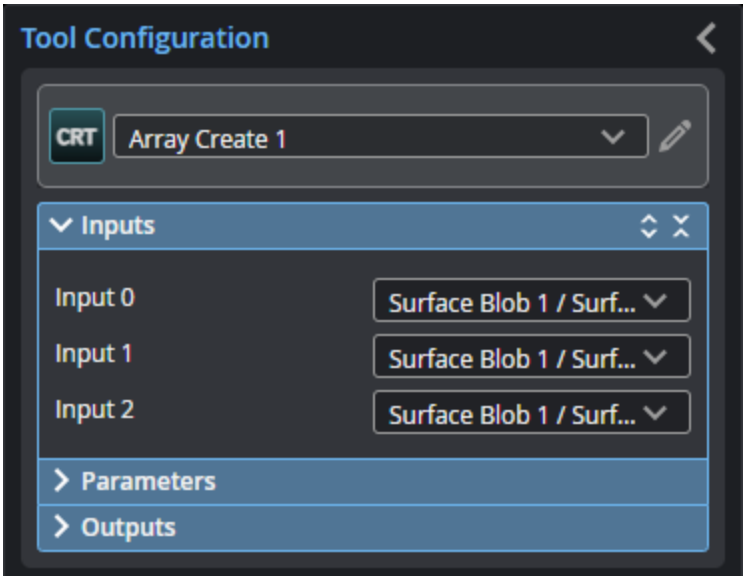
- Array Create: Takes *individual* inputs and creates an array containing those pieces of data as elements. For more information on this tool, see *Array Create* on the next page.

Array Create

The Array Create tool combines individual pieces of data (Profile, Surface, measurements, and so on) into an array. For example, a number of Surface scans could be combined into an array made up of those pieces of Surface data. Compatible measurement and processing tools can then take the array as input and perform batch operations on the elements in the array, such as measurements on each element's data or processing each element's data. For more information, see *Arrays, Batching, and Aggregation* on page 199.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

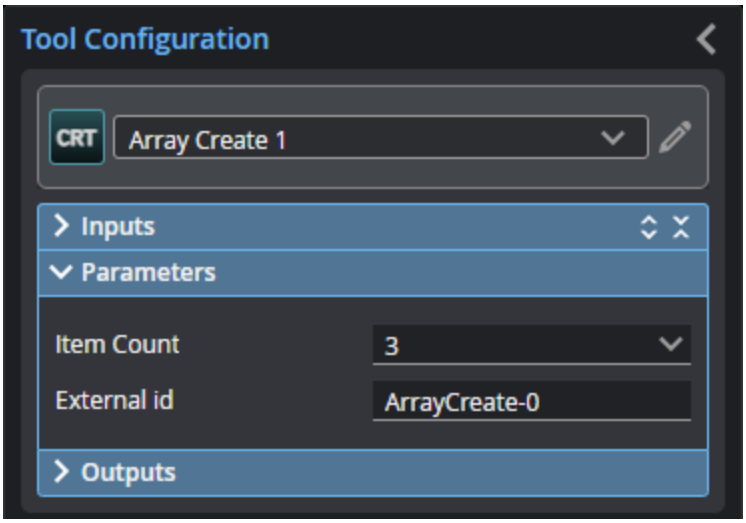


Inputs

Name	Description
Input {n}	An input that the tool will add to the output array. The number of inputs depends on the **Item Count** parameter. All inputs must be of the same type. The tool supports taking arrays as inputs and creates and array containing all elements in the inputs.

Parameters

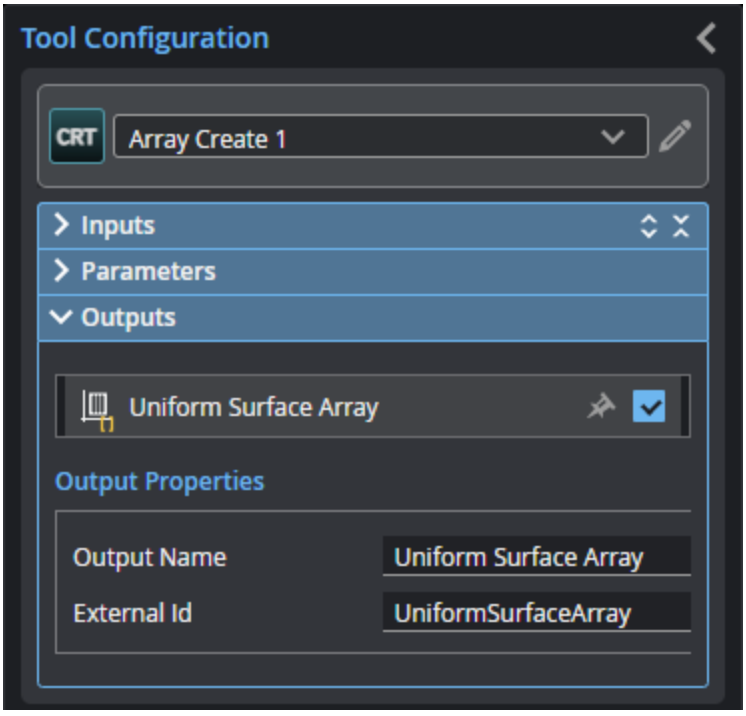
The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

Parameter	Description
Item Count	The number of items expected in the array.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



The type of the output depends on the type of the inputs.

Array Decision

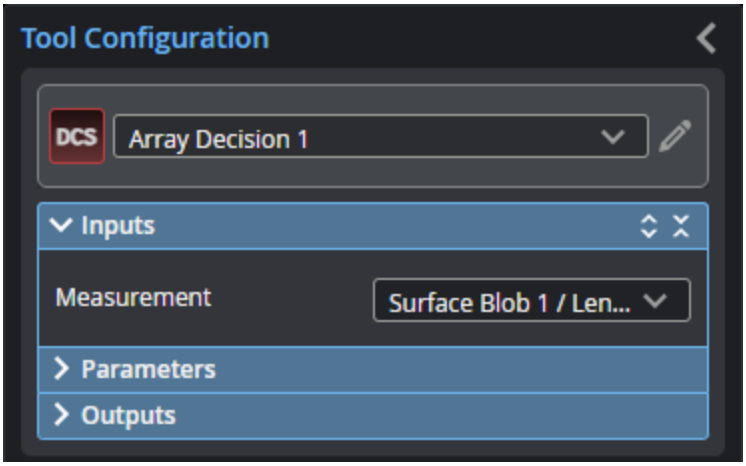
The Array Decision tool takes an array of measurements as input and applies individual minimum / maximum thresholds to one or more of the array elements. The tool returns results related to fails (number of fails, index and value of max/min fail, as well as arrays of failed measurements and their indices).

A common use case of Array Decision is to make pass/fail decisions for electronics pin inspection applications.

For general information on the Array tools, see *Array Tools* on page 233.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

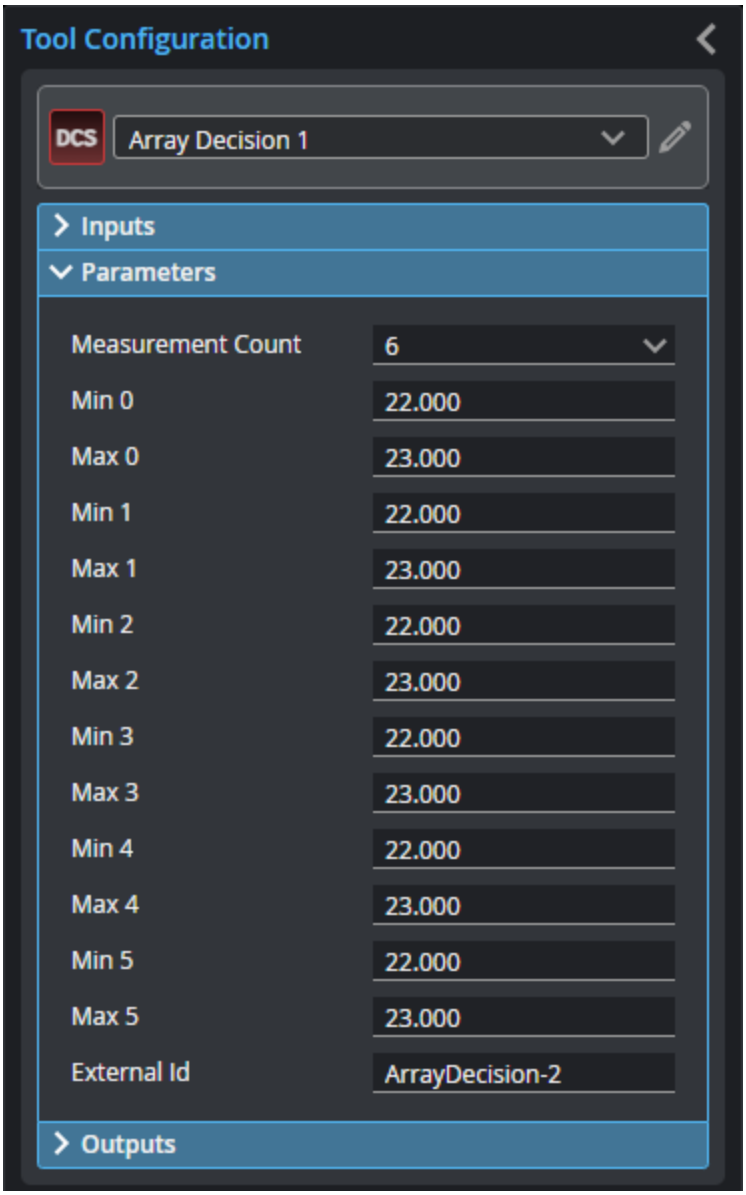


Inputs

Name	Description
Measurement	The array of measurements the tool takes as input.

Parameters

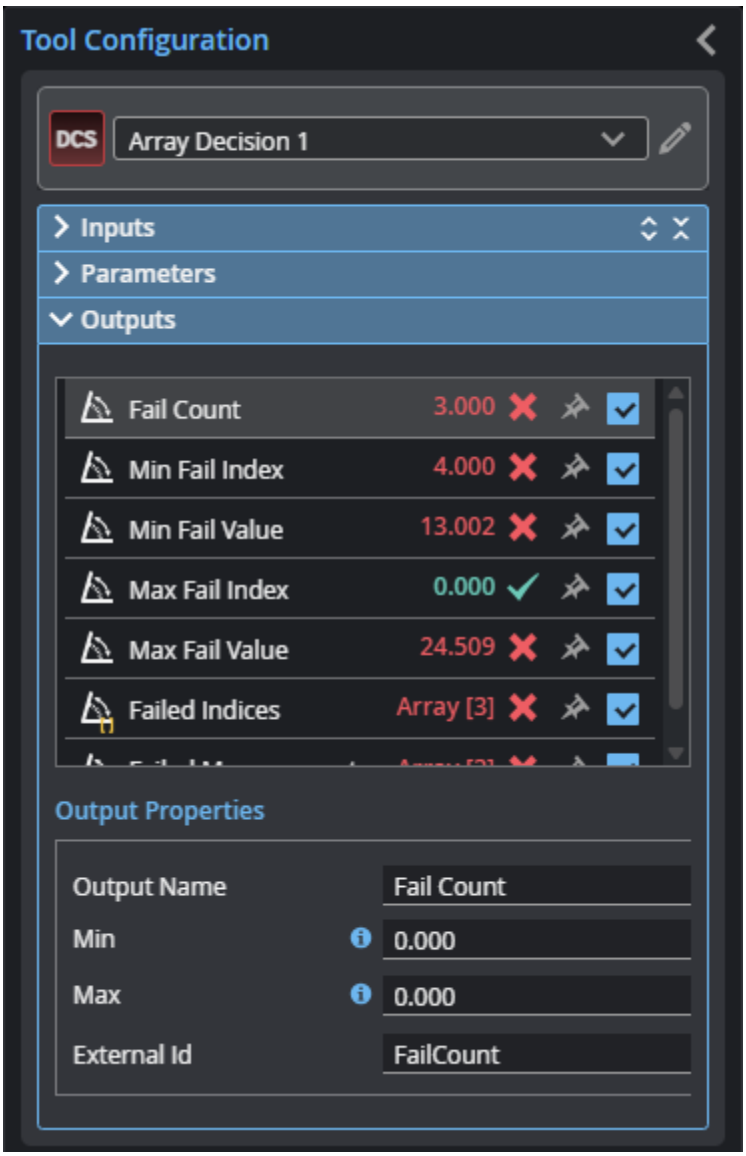
The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

Parameter	Description
Measurement Count	The maximum number of array elements expected in the input array. The tool displays Min{n} and Max{n} fields for each element.
Min {n} Max {n}	The minimum and maximum threshold for measurement element {n} in the input array of measurements.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



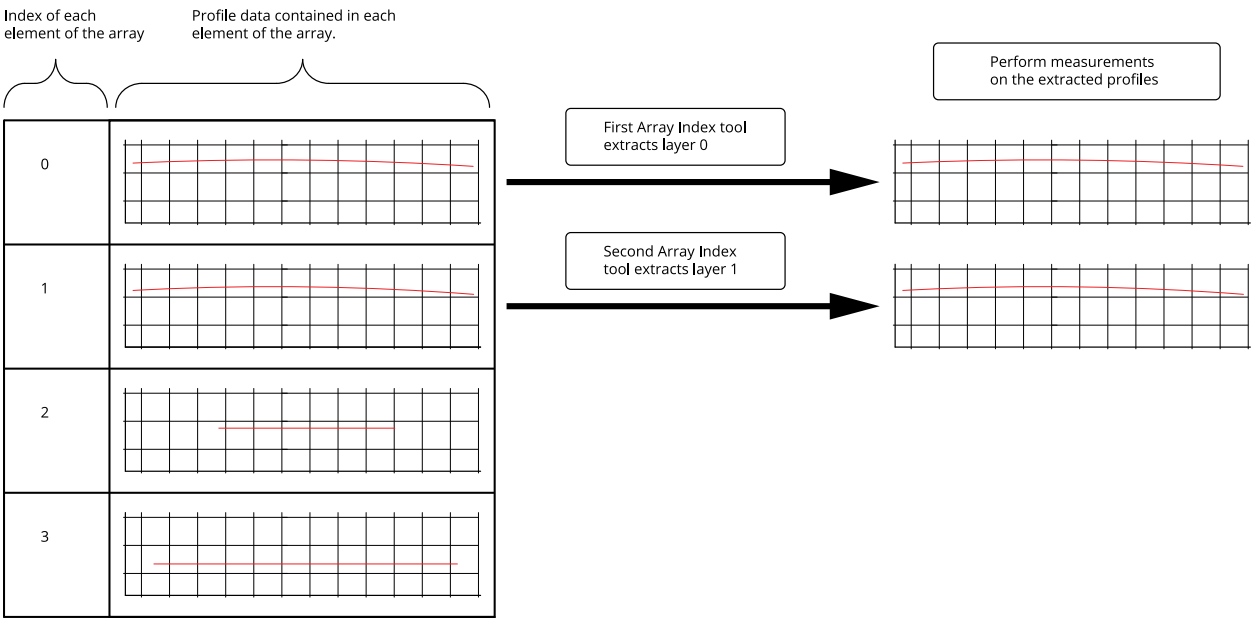
Measurements

Measurement
Fail Count The number of array values that are outside their Min/Max thresholds.
Min Fail Index The index of the array element with the greatest difference with respect to its Min threshold.
Min Fail Value The value of the array element at Min Fail Index.
Max Fail Index The index of the array element with the greatest difference with respect to its Max threshold.

Measurement
Max Fail Value The value of the array element at Max Fail Index.
Failed Indices An array of the indices of the failed measurements.
Failed Measurements An array of the failed measurement values.

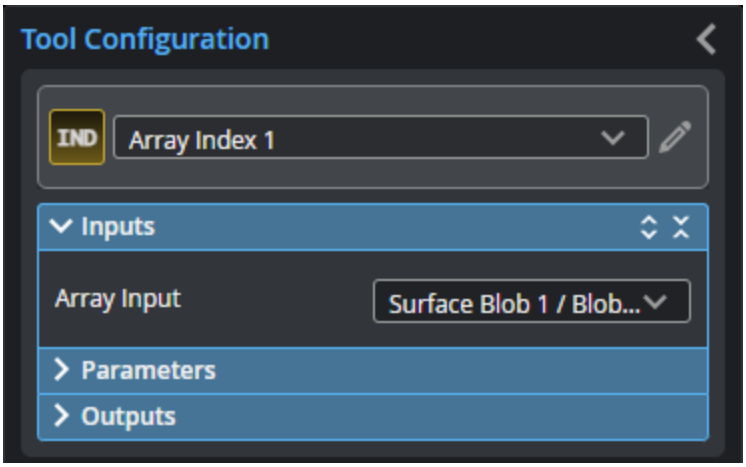
Array Index

The Array Index tool extracts a one or more pieces of data (such as a profile or a surface) from the array it takes as input starting from the position (the index) you specify.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

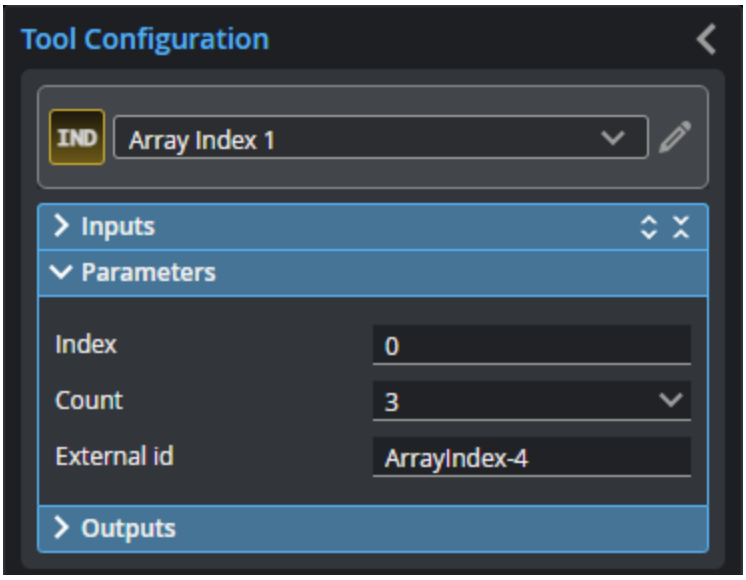


Inputs

Name	Description
Array Input	The data the tool applies measurements to or processes.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

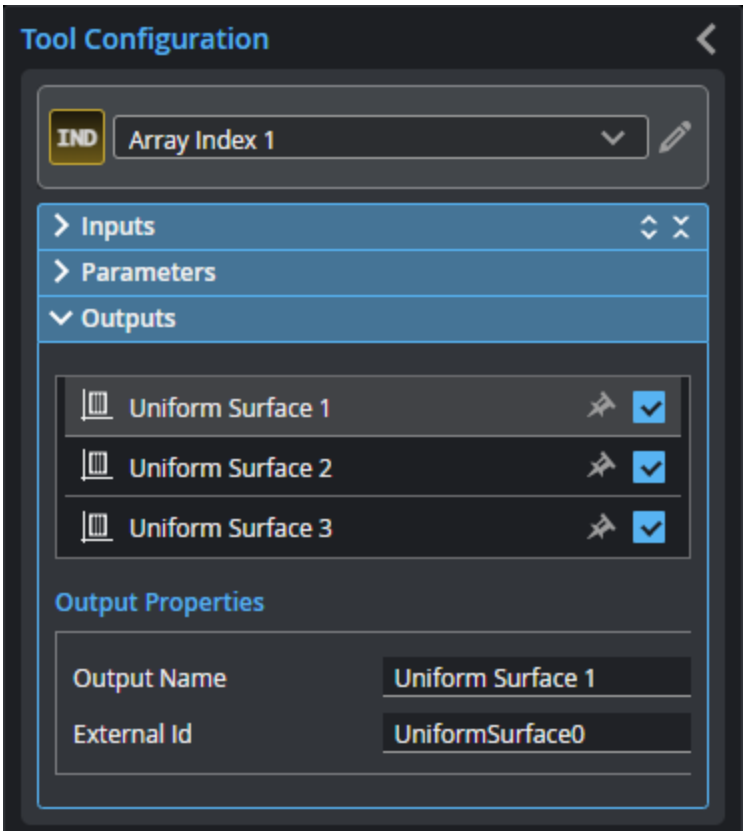


Parameters

Parameter	Description
Index	The position (the index) in the array at which the tool extracts and outputs. If **Count** is set to a value greater than 1, the index is the starting point for array element outputs.
Count	The number of individual outputs the tool produces, starting at the provided index (see above). This setting makes sending multiple outputs over protocols easier.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

If **Count** is set to a value greater than 1, the tool produces multiple outputs.



The type of the output depends on the type of the input array.

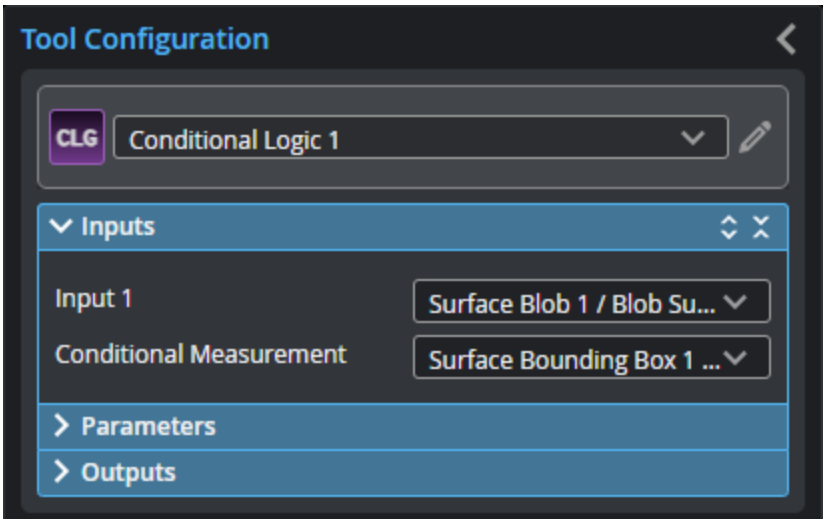
Conditional Logic Tool

The Conditional Logic tool sends input data to different outputs based on a conditional measurement value. The tool can take multiple inputs. This lets you create complex processing and inspection tool chains that depend on the conditions you set. By limiting the tool's output to a subset of the input it receives, for example, job runtime can be reduced.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

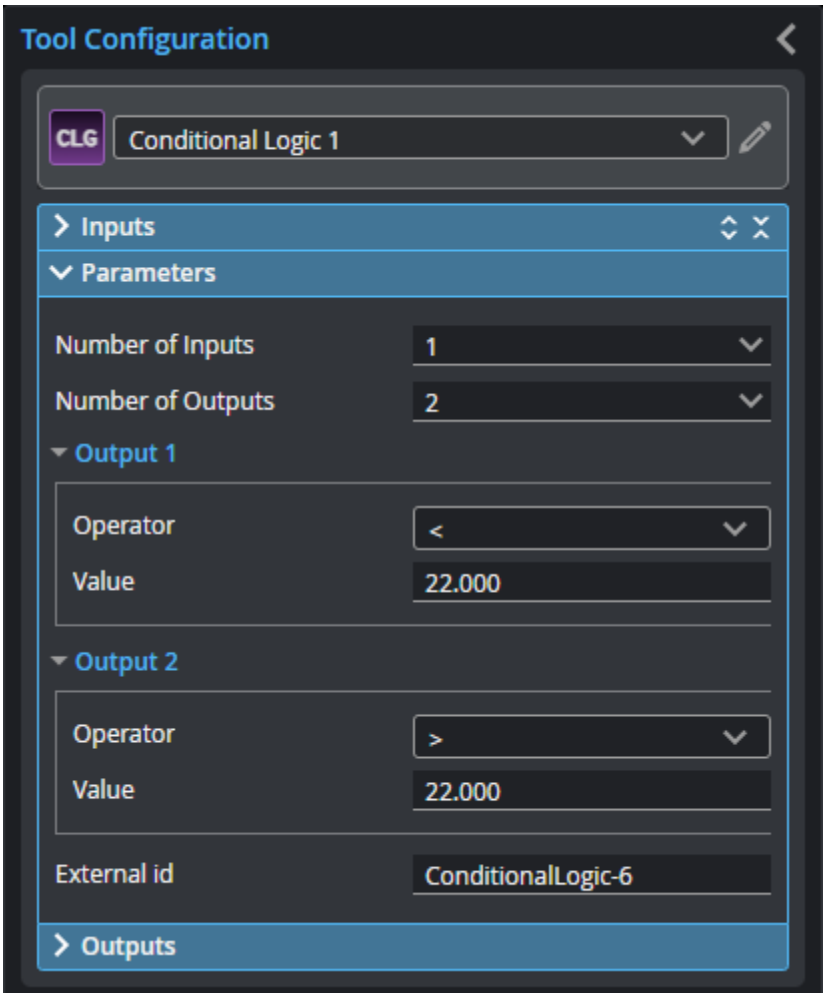


Inputs

Name	Description
Input {n}	An input the tool routes to an output, based on the conditions (an operator and a value) set in the **Parameters** section. This tool can optionally take an array as input. For more information, see Arrays, Batching, and Aggregation.
Conditional Measurement	The measurement whose value the tool compares to a value set in **Parameters** > **Output {n}**, using the operator selected in the **Output {n}** section.

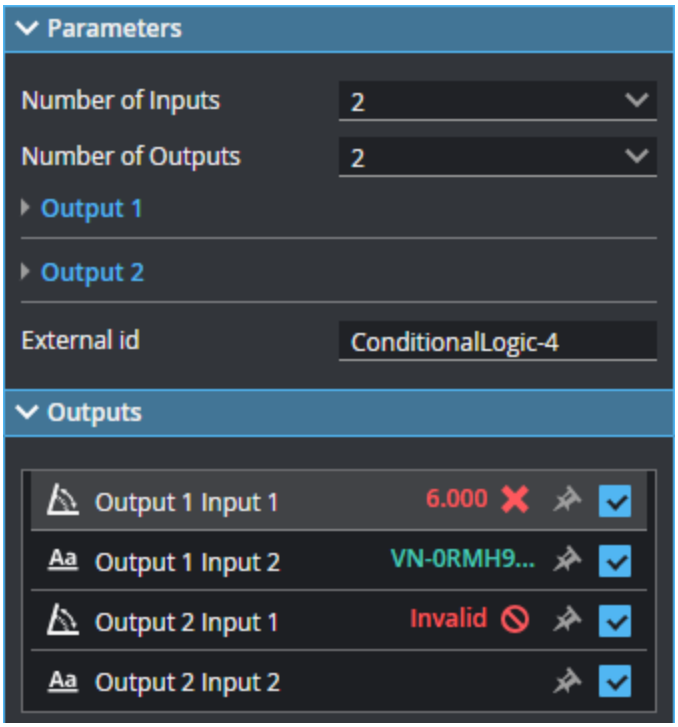
Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Name	Description
Number of Inputs	The number of data inputs the tool accepts. Set the inputs in the **Inputs** section. If the tool takes more than one input, the tool outputs all possible combinations.

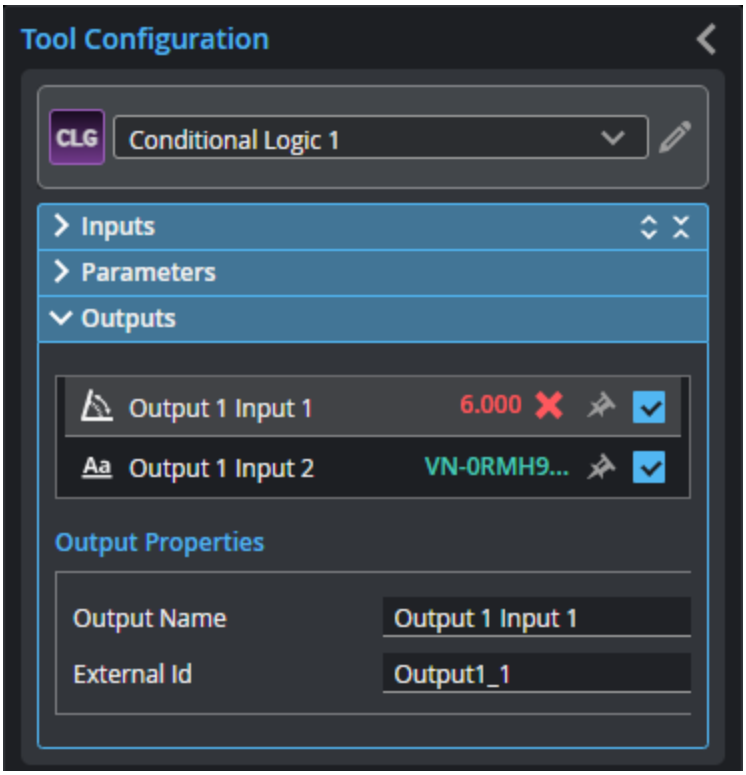
Name	Description
	
Number of Outputs	The number of data outputs the tool potentially produces. For each output, configure Operator and Value in the expandable **Output {n}** section.
Output {n}	The relational operator (<, >, ≤, ≥, =, ≠) and value the tool uses to compare with the value of the conditional measurement. Set **Operator** and **Value**.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.



Data

Type	Description
Output {n} Output {n} Input {n}	Data satisfying the comparison configured in **Output {n}** in the **Parameters** section. If the tool takes more than one input, the tool outputs all possible combinations, and names them using "Output {n} Input {n}" as the convention.

Data Export Tool

The Data Export tool saves frames of scan data to local PC storage (by default, `C:\GoTools\DataExport`), letting you import and evaluate scan data in 3rd-party applications.

> This tool is only available in a PC instance of GoPxL. It is not available on a sensor or on GoMax.

You can optionally use a measurement as a condition for data export. For example, you could configure the tool to only export data if a Surface Position's Z measurement is a Fail (or a Pass).

The tool supports exporting the following formats. The available formats depend on the tool's input.

Output formats

Format	Surface	Profile	Image	Description
CSV (.csv)	✓	✓		CSV format. For more information, see *CSV File Format* on page 1295.
PRO (.pro)		✓		MountainsMap format. Uniform data only.
Wavefront OBJ (.obj)	✓			Mesh with comma-separated vertices and facets in text format.
ODSCAD OMC (.omc)	✓			ODSCAD heightmap.
PCL PCD (.pcd)	✓			Point Cloud Data (PCD) file format for the Point Cloud Library (PCL).
PNG (.png)	✓		✓	**Surface** The heightmap as a PNG image. The image is either a 24-bit RGB image or a 16-bit grayscale image, depending on whether **16-bit Depth** is enabled (see below). **Image** An image from the sensor camera. The bit-depth depends on the camera.
STL ASCII (.stl)	✓			Mesh in the standard STL text format (can become very large).
STL Binary (.stl)	✓			Mesh in the binary STL format.
MountainsMap SUR (.sur)	✓			MountainsMap SUR heightmap.
TIFF (.tif)	✓		✓	**Surface** The heightmap as a TIFF image. The image is either a 24-bit RGB image or a 16-bit grayscale image, depending on whether **16-bit Depth** is enabled (see below).

Format	Surface	Profile	Image	Description
				Image An image from the sensor camera. The bit-depth depends on the camera.
GoPxL Recording (.gprec)	✓	✓	✓	GoPxL replay data.

The file naming convention is as follows:

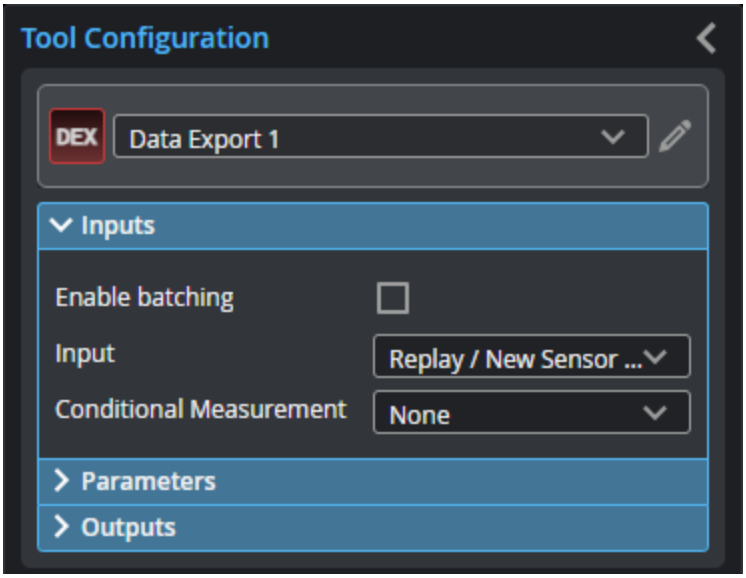
`<frame>_<timestamp>_<counter>.<extension>`

`<counter>` is optional: It is only present if a file with the same name is already in the output folder.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



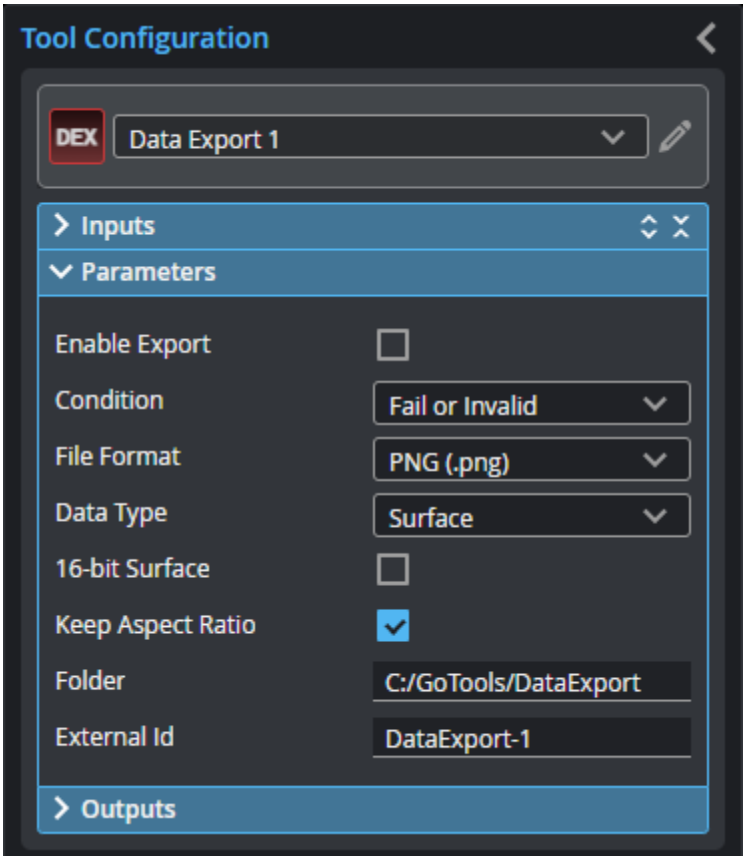
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and exports the data for each item in the array individually. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array

Name	Description
	of surfaces with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Input	The data the tool exports. This tool can optionally take an array as input. For more information, see Arrays, Batching, and Aggregation.
Conditional Measurement	The measurement the tool uses to determine whether to export data. Can be None, in which case the tool exports all frames. In the **Parameters** section, set **Condition** to the condition type (see below).

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



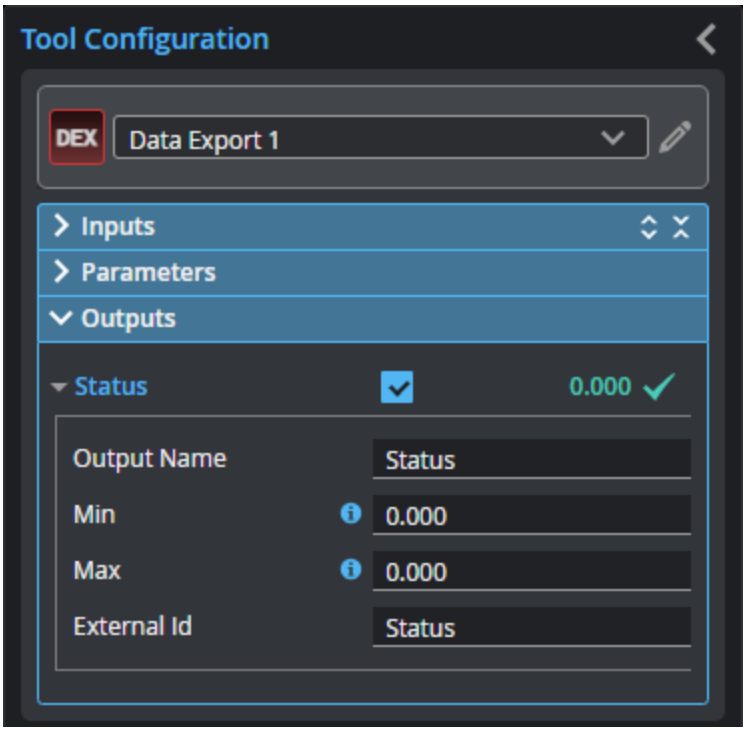
Parameters

Name	Description
Enable Export	When this parameter is checked, the tool exports frames in the format you select

Name	Description
	in **File Format** to the folder specified in **Folder**.
Condition	If the **Conditional Measurement** input is set to a measurement, this determines which state of the measurement causes the tool to export data. One of the following: • Fail or Invalid • Invalid • Pass The setting is only displayed if the **Conditional Measurement** input is set to a measurement.
File Format	The format the tool uses to export data. For a list of the formats, see *Output formats* on page 247.
Data Type	The type of data the tool outputs, using the format selected in **File Format**. One of the following: **Surface** Outputs Surface data. **Intensity** If intensity data is available, the tool outputs this data. Otherwise, no output is produced. **Surface and Intensity** Outputs Surface data and, if it is available, intensity data. The tool appends "_surface" and "_intensity" to the file names of the outputs.
16-bit Surface	The bit depth of the heightmap output as an image. When disabled, the tool outputs a 24-bit RGB image. When enabled, the tool outputs a 16-bit grayscale image. Only available for PNG and TIFF outputs.
Keep Aspect Ratio	Preserves the aspect ration of the image outputs (PNG and TIFF) by resampling to equal XY spacing.
Folder	The folder in which the tool saves exported data. If the folder doesn't exist, the tool creates the folder. By default, `C:/GoTools/DataExport`.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

This tool provides no outputs other than saving exported data in `C:\GoTools\DataExport`.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	
Status	
Indicates whether the current frame has been exported. One of the following values: 1: The frame is saved. 0: The frame is not saved because the condition is not met. For more information, see *Condition* on the previous page. -1: The frame save fails because of insufficient disk space, incorrect file permissions, and so on.	

Feature Tools

The following sections describe the Feature tools.

Feature tools generate or measure geometric features; for more information on geometric features, see *Geometric Features* on page 227. The tools let you work with more complex geometry, letting you implement applications more quickly by reducing dependence on scripts and GDK tools to perform these kinds of measurements.

The "Create" tools (such as Feature Circle Create) generate geometric features from other tools or using values you provide.

The Feature Dimension tool lets you take measurements using geometric features.

The Feature Robot Pose tool is for use with robots.

Feature Circle Create

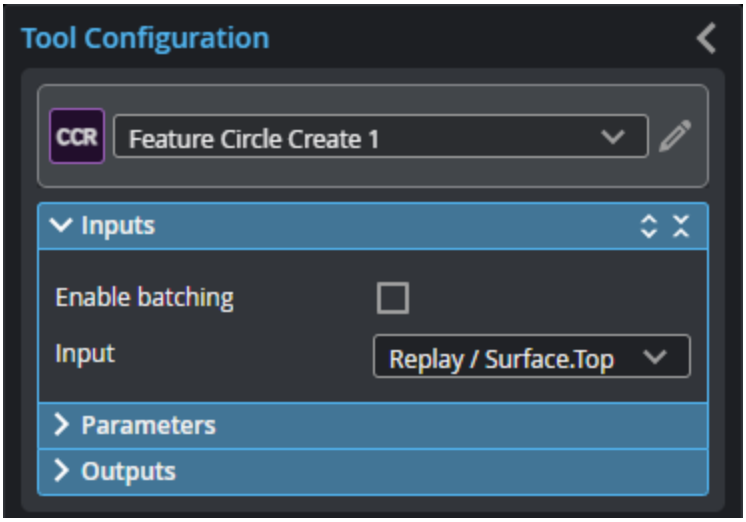
The Feature Circle Create tool lets you generate a Circle geometric features from geometric features generated by other tools. You can also generate a circle from an arbitrary position and radius. The following outputs are available (for more information, see *Output Types* on the next page):

- Constant circle
- Circle from points
- Circle from point and plane
- Circle tangent to two lines

For more information on geometric features, see *Geometric Features* on page 227.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

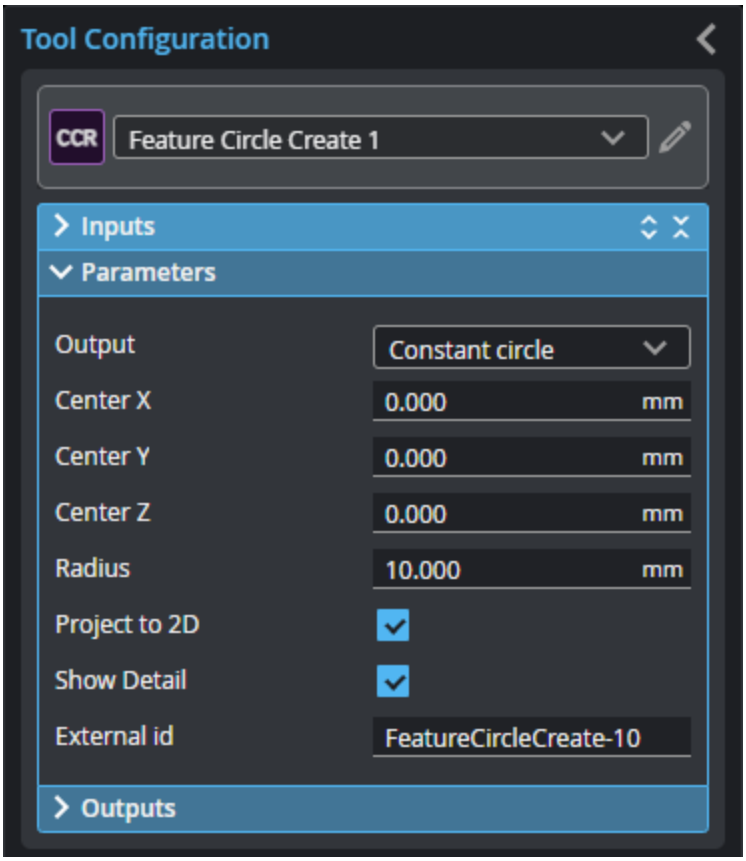


This tool can take an array as input. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.

The tool's geometric feature inputs depend on the output type you choose in the **Output** parameter. For information on the required inputs and parameters of each output type, see *Output Types* on the next page.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Output	The type of output the tool generates. Switching between the options changes the input types displayed in Input section and the parameters displayed in the Parameters section. For information on the output types and the required inputs and related parameters, see *Output Types* below.
Project To 2D	If checked, generates a circle in the XZ plane for Profile mode and in the XY plane for Surface mode. Otherwise, generates a 3D circle. Not shown when **Output** is set to “Circle from point and plane”.
Show Detail	Toggles the display of the input geometric features in the data viewer.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Output Types

The following sections describe the output types available in the tool's **Output** parameter, in addition to the required inputs and parameters.

CONSTANT CIRCLE

The **Constant circle** output type is useful if scan data from frame to frame is reliably fixed and you want to measure from an arbitrary circle to another feature. When you choose this output type, the tool displays the following parameters you can manually fill in to create a constant circle.

Disable **Project To 2D** to input an arbitrary normal vector (using the **Normal** parameters) to get a tilted 3D circle.

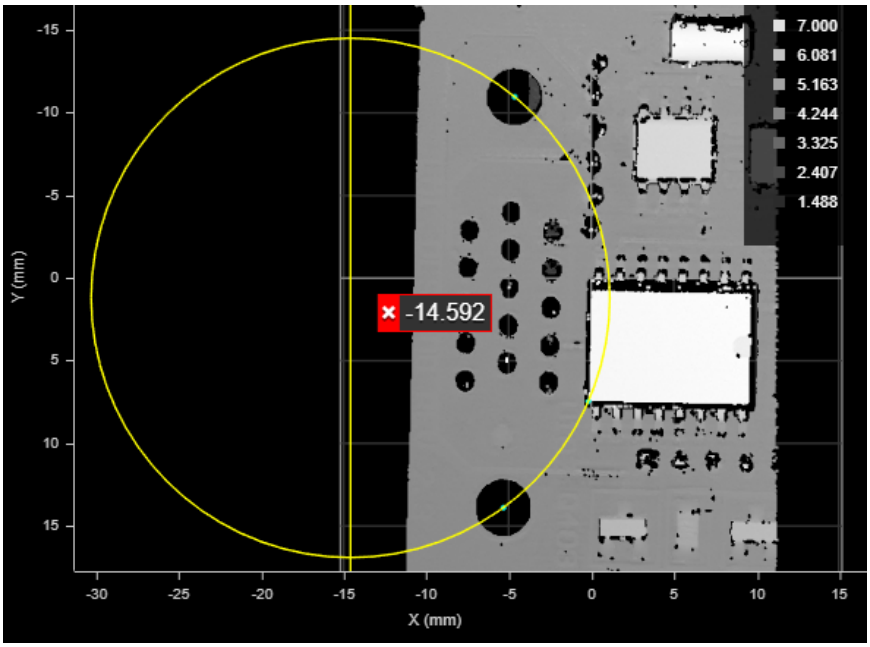
Constant Circle Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	**Center X**, **Center Y**, **Center Z** The X, Y, and Z position, respectively, of the center of the constant circle.
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z offsets the tool will apply to the generated circle.	**Normal X**, **Normal Y**, **Normal Z** The X, Y, and Z for the normal vector. Only displayed when **Project To 2D** is disabled.
	Radius The radius of the constant circle.

CIRCLE FROM POINTS

The **Circle from points** output type takes three point geometric features generated by another tool and fits a circle to those points. The circle is always on the XY plane.

When **Project To 2D** is enabled, the circle is projected to the XY plane. When **Project To 2D** is disabled, circle is tilted to fit the plane the three point inputs create.



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.)

The X, Y, and Z measurements return the center of the circle. The X, Y, and Z Angle measurements return arbitrary values.

Circle from points Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	None.
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z offsets the tool will apply to the generated circle.	
Point 1, **Point 2**, **Point 3** The three Point geometric features the tool uses as the center of the circle.	

CIRCLE FROM POINT AND PLANE

The **Circle from point and plane** output type generates a circle using Point and Plane geometric features and a user-defined radius. The circle is on the Plane received as input.

Circle from point and plane Output Type - Inputs and Parameters

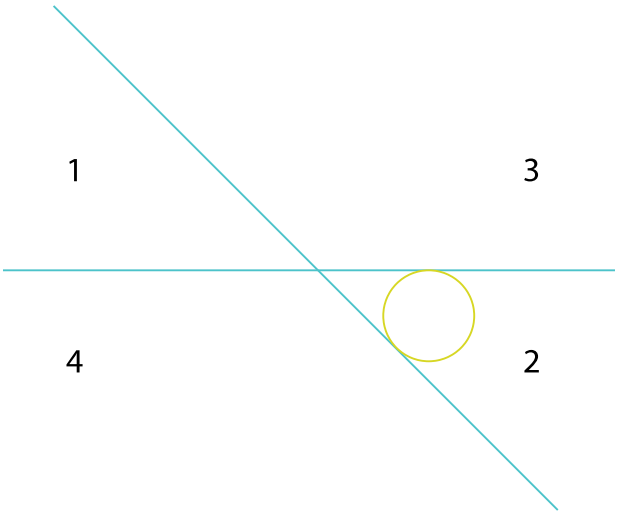
Inputs	Type-Related Parameters
Input The Surface or Profile input.	**Radius** The radius of the circle.
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z offsets the tool will apply to the generated circle.	
Point The Point geometric features the tool uses as the center of the circle.	
Plane The Point geometric features the tool uses as the center of the circle.	

CIRCLE TANGENT TO TWO LINES

The **Circle tangent to two lines** takes two Line geometric features and an optional Plane geometric feature to create a circle that is tangent to the two provided lines. The circle is on the Plane received as input.

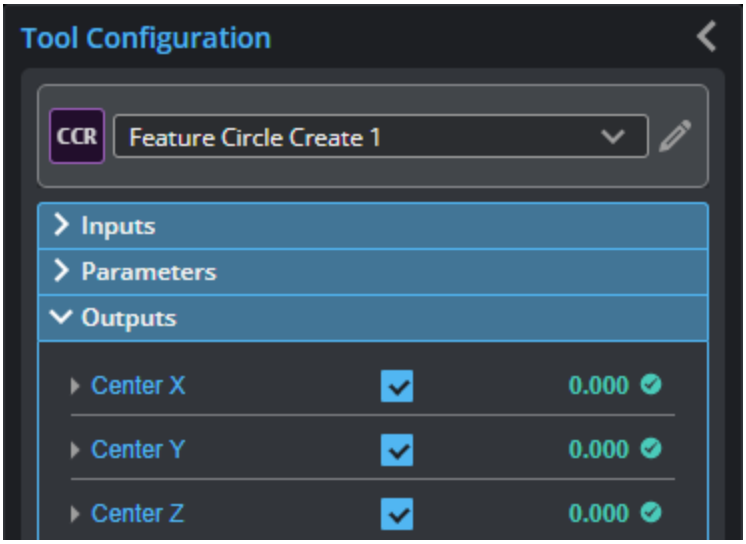
Circle from point and plane Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	**From Acute Angle** Toggles between using the acute angle between the lines (enabled) or the obtuse angle (disabled).
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z offsets the tool will apply to the generated circle.	**Circle Selection** Selects whether the tool fits the circle between the upper or lower angle.
Point The Point geometric features the tool uses as the center of the circle.	**Circle Radius** The radius of the circle.
Plane The Point geometric features the tool uses as the center of the circle.	The tool uses **From Acute Angle** and **Circle Selection** together to determine where to fit the circle. For example, given these two lines, the tool fits the circle between the line segments as described below:

Inputs	Type-Related Parameters
	 1: **From Acute Angle** enabled, "Upper circle" selection. 2: **From Acute Angle** enabled, "Lower circle" selection. (Circle fitted here.) 3: **From Acute Angle** disabled (uses obtuse), "Upper circle" selection. 4: **From Acute Angle** disabled (uses obtuse), "Lower circle" selection.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

For information specific to the different output types, see *Output Types* on page 254.

Measurements

Measurement
Center X, **Center Y**, **Center Z** The X, Y, and Z positions of the center of the circle.
Radius The radius of the circle.
Z Offset The Z position of the intersection between the circle plane and the Z axis.
Distance The distance from the origin to the circle plane.
Normal X, **Normal Y**, **Normal Z** The X, Y, and Z normal vectores of the circle.

Features

Type	Description
Circle	The generated Circle geometric feature. For more information, see *Output Types* on page 254.

Feature Dimension

The Feature Dimension tool returns dimensional measurements from a point geometric feature to a reference point, line, or plane geometric feature. Geometric features are available as outputs from many tools, for example a Point geometric feature from Surface Position, or a line from Surface Edge. The feature creation tools such as Feature Circle Create, Feature Line Create, and so on, can also create geometric feature. For more information on geometric features, see *Geometric Features* on page 227.

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.

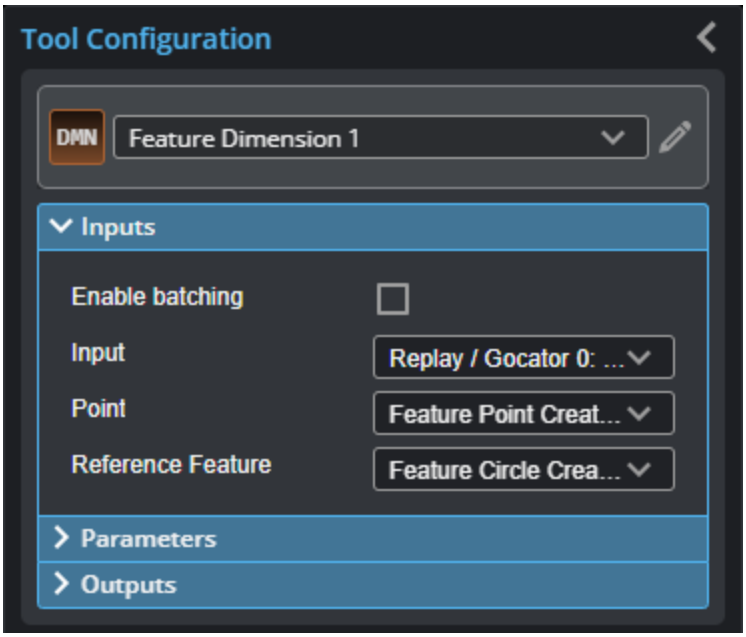
See *Adding a Tool* on page 164 for instructions on how to add measurement tools.

> 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).

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

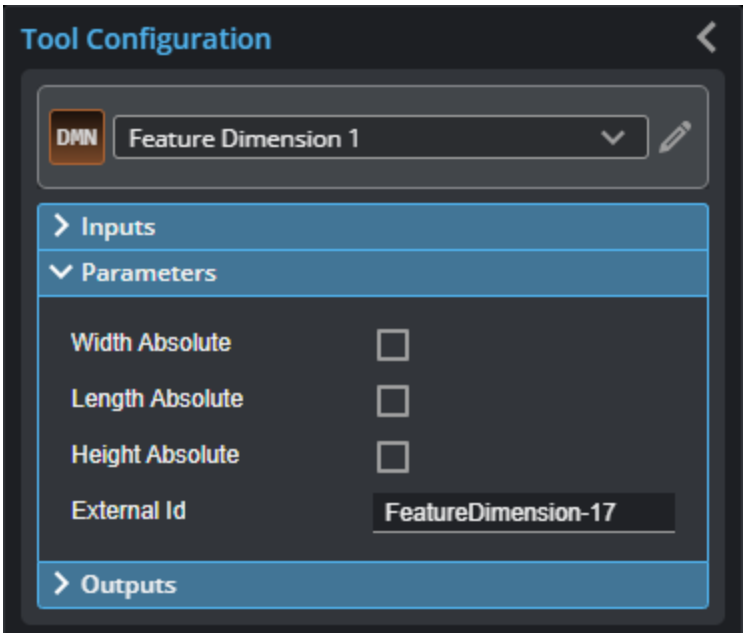


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Input	The data the tool applies measurements to.
Point	A point geometric feature generated by another tool.
Reference Feature	A geometric feature generated by another tool. Dimensional measurements are calculated *from* the reference feature *to* the point in the **Point** input.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

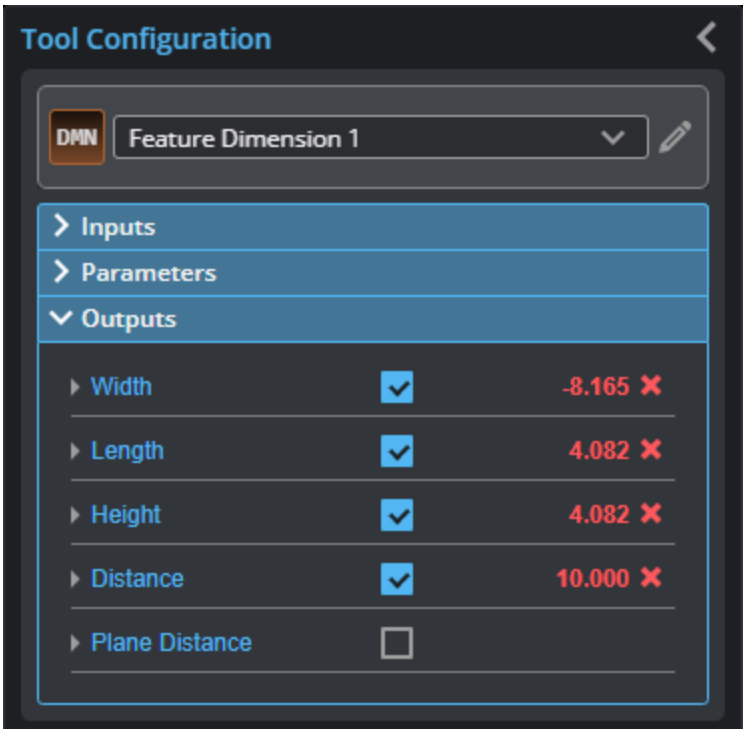


Parameters

Parameter	Description
Width Absolute Length Absolute Height Absolute	The tools returns measurements as absolute values.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

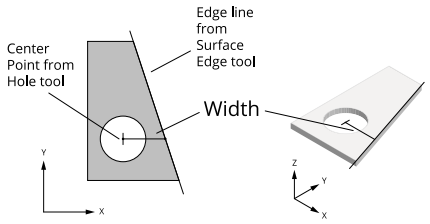


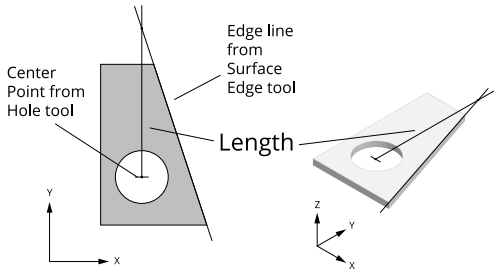
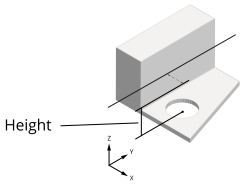
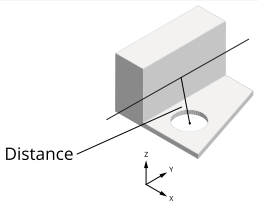
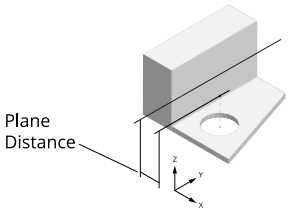
To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.	

Measurement	Illustration
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.	

Feature Intersect

The Feature Intersect tool returns the intersection of a line or plane geometric feature 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'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.

For more information on anchoring, see *Measurement Anchoring* on page 209.

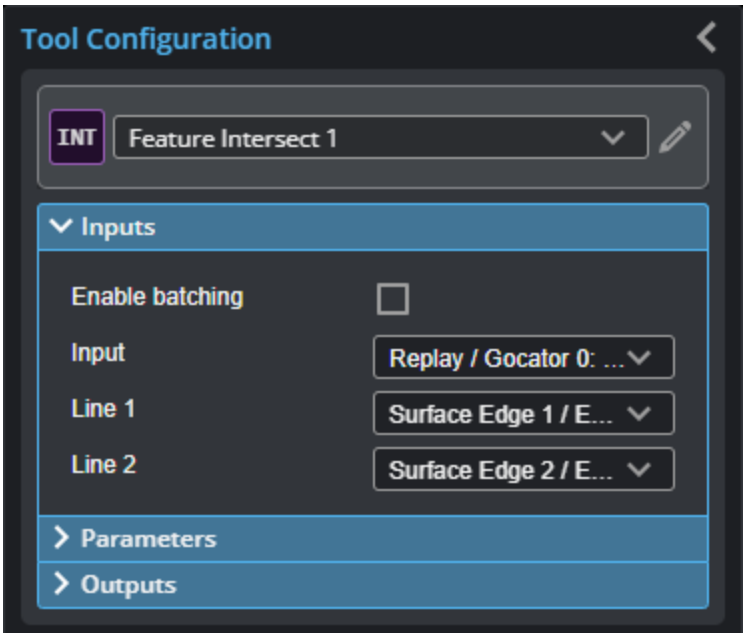
The Feature Intersect tool can also generate a point geometric feature representing the point of intersection of the lines that the Feature Dimension tool can use in measurements.

See *Adding a Tool* on page 164 for instructions on how to add measurement tools.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

The tool's inputs depend on the intersection type you choose in the **Intersection Type** parameter. For information on the required inputs and parameters of each output type, see *Feature Intersect* above.

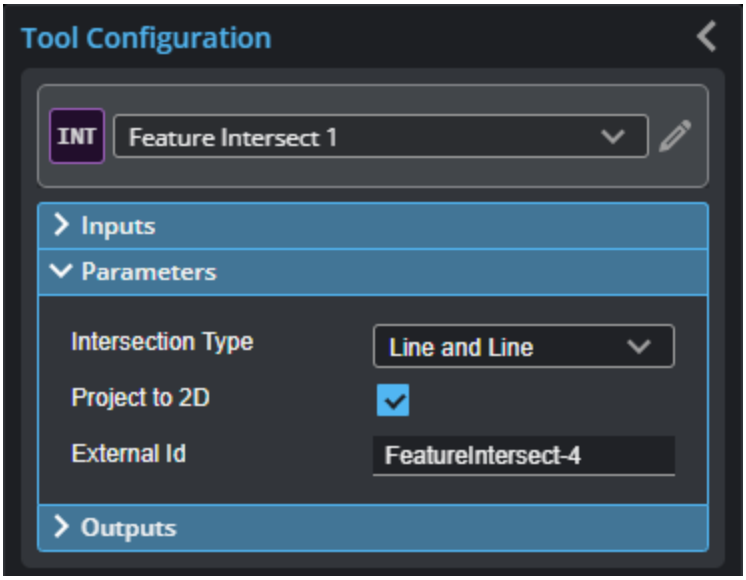


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Input	The data the tool applies measurements to or processes.
Line, Circle, or Plane	The geometric features the tool uses to calculate an intersect. A combination of Line, Circle, or Plane geometric features, depending on the **Intersection Type** parameter.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

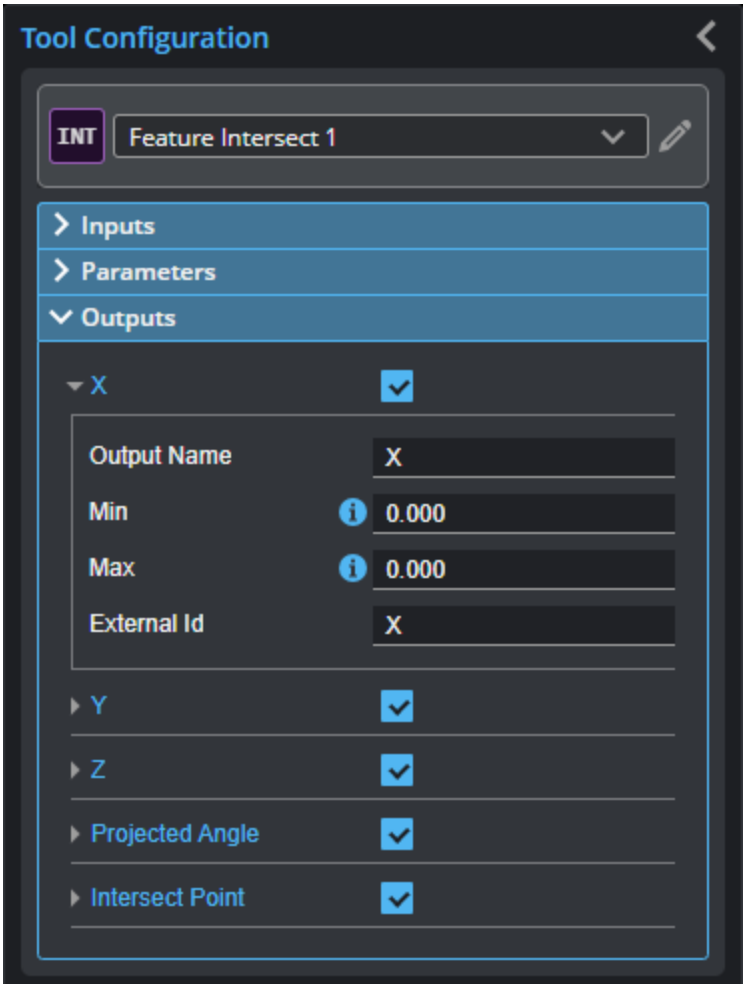


Parameters

Parameter	Description
Intersection Type	One of the following: • Line and Line • Line and Plane • Plane and Plane • Line and Circle The intersection type determines which inputs are available in the **Inputs**.
Intersection Option	When **Intersection Type** is Line and Circle, provide the location of the option (intersection on the circle with the lowest versus the highest X position, and so on).
Project to 2D	Only shows when Intersection Type is “Line and Line”. If checked, the intersection

Parameter	Description
	angle is projected to the XZ plane for Profile mode and XY plane for Surface mode. Otherwise, outputs the 3D angle.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

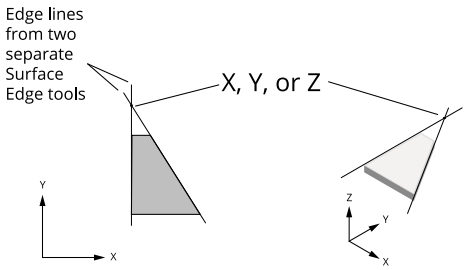
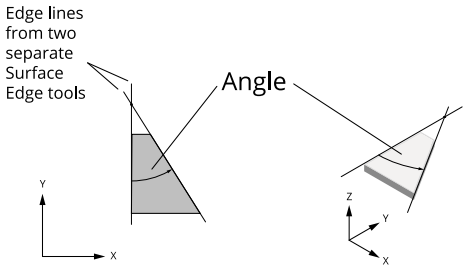
Outputs

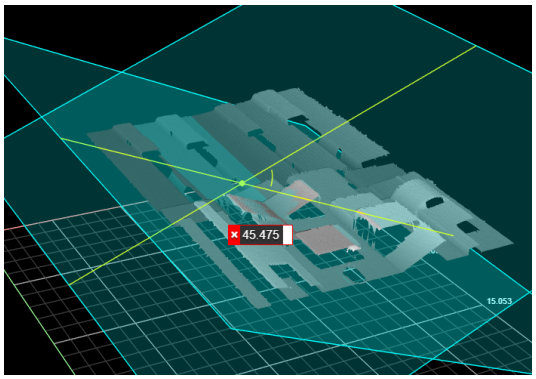


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.	
Angle or **Projected Angle** (*Line-Line*: The angle between the lines, as measured from the line selected in **Reference Feature** to the line selected in **Line**. If **Project To 2D** is enabled, the output is Projected Angle. *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**.	 In the following image, the angle is measured between two planes (the small angled surfaces facing each other in the center of the image).

Measurement	Illustration
	

Features

Type	Description
Intersect Point	The intersect point of the two features.
Intersect Line	The line representing the intersection of two planes.

Feature Line Create

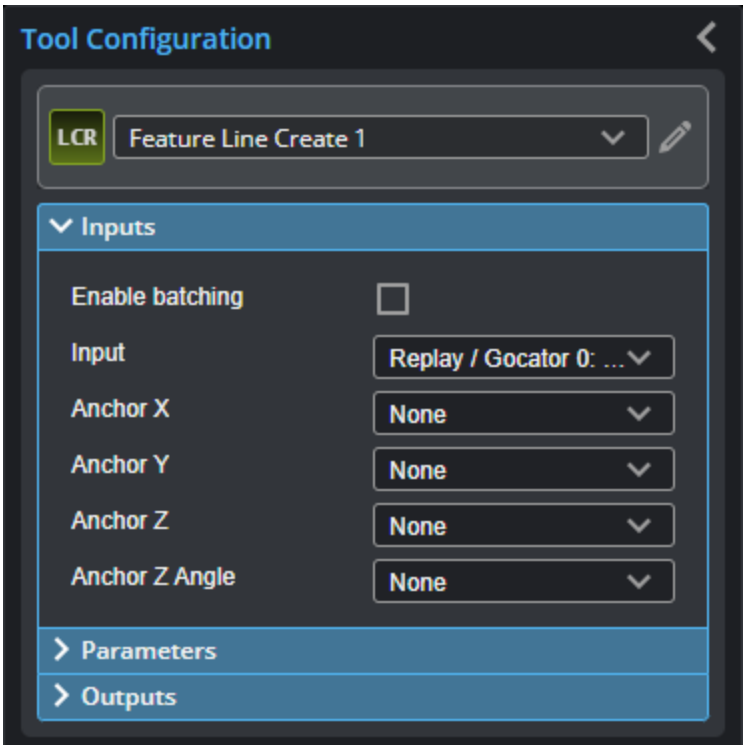
The Feature Line Create tool lets you generate a Line geometric features from geometric features generated by other tools. You can also generate a line from arbitrary positions you provide. The following outputs are available (for more information, see *Output Types* on page 273):

- Line
- Constant line
- Line from offset
- Line from two points
- Perpendicular line from point and line
- Perpendicular line from point and plane
- Parallel line from point and line
- Intersect line of two planes
- Projected line on plane
- Center line from two lines
- Projected center line from two lines and a plane
- Rotate line around point

For more information on geometric features, see *Geometric Features* on page 227.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

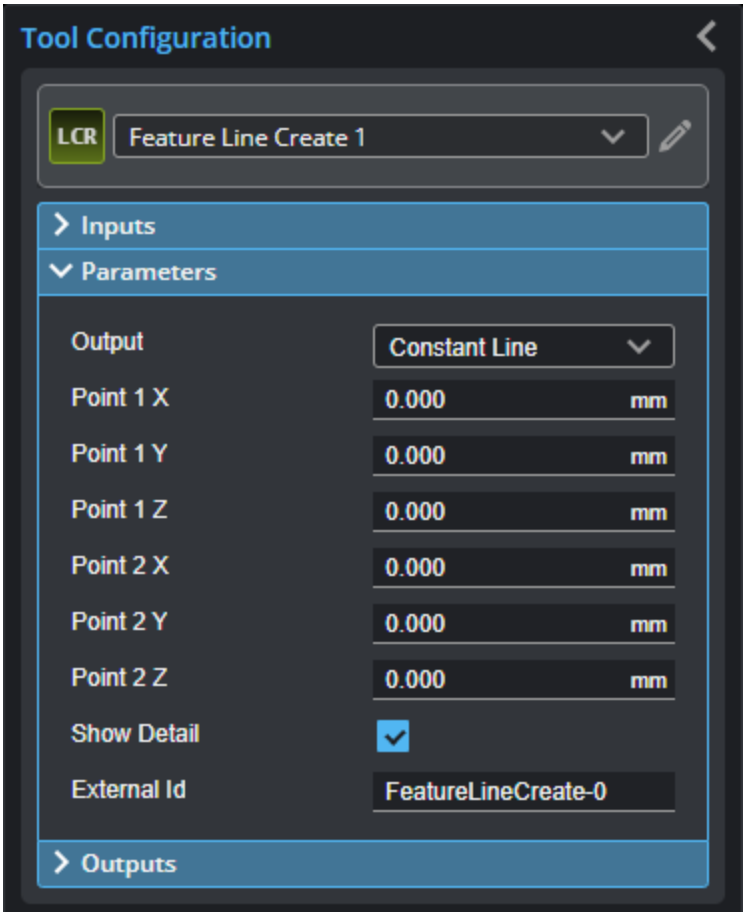


This tool can take an array as input. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.

The tool's inputs depend on the output type you choose in the **Output** parameter. For information on the required inputs and parameters of each output type, see *Output Types* on the next page.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Output	The type of output the tool generates. Switching between the options changes the input types displayed in Input section and the parameters displayed in the Parameters section. For information on the output types and the related parameters, see *Output Types* on the next page.
Show Detail	Toggles the display of the input geometric features in the data viewer.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Output Types

The following sections describe the output types available in the tool's **Output** parameter, in addition to the required inputs and parameters.

LINE

The **Line** output type takes a Line geometric feature as input. This output is useful if the tool takes a feature generated by another Feature Create tool as input, on which you want to perform measurements in a second Feature Create tool. This tool can also be useful if you have developed GDK tools that only generate geometric features (no measurements): you can use this tool to extract those measurements.

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 and Y angle measurements return arbitrary values.

Line Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	None.
Line The Line geometric feature the tool uses to create another Line geometric feature.	

CONSTANT LINE

The **Constant line** output type displays parameters you can manually fill in to create a geometric feature. This output type is useful if the scan data from frame to frame is reliably fixed and you want to measure from an arbitrary line to another feature.

Line Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	**Point 1 X**, **Point 1 Y**, **Point 1 Z** The X, Y, and Z position, respectively, of the first end point of the constant line.
	Point 2 X, **Point 2 Y**, **Point 2 Z** The X, Y, and Z position, respectively, of the second end point of the constant line.

LINE FROM OFFSET

The **Line from offset** output type takes a Line geometric features as input.

The X, Y, and Z measurements return the midpoint of the line.

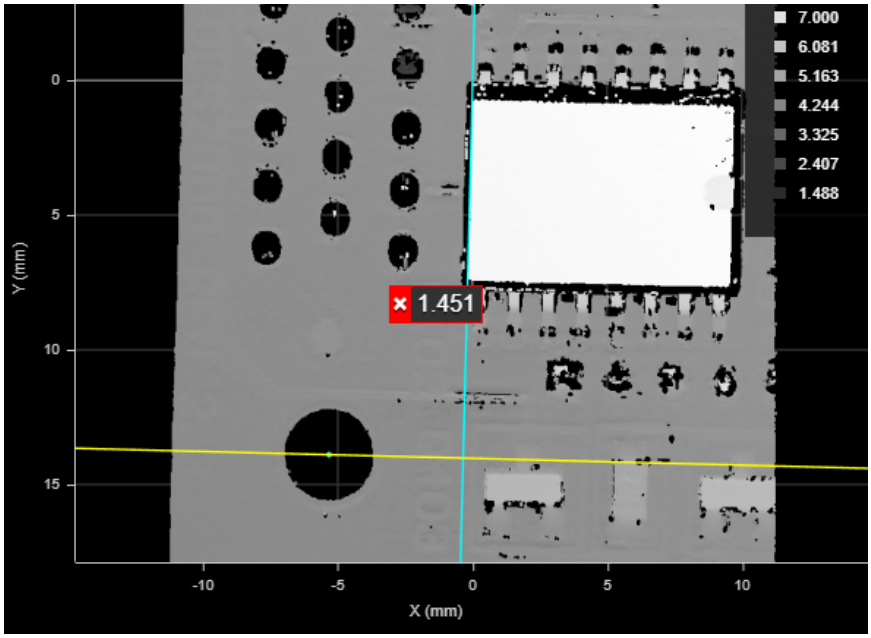
Line from offset Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input. **Line** The Line geometric feature the tool uses to create another Line geometric feature.	For Surface data: **X Distance / Y Distance** Offsets the input line by the provided X and Y distance. For Profile data: **Offset Direction** The offset direction. Either Positive X or Positive Z. **Offset Distance** The distance along the axis specified in **Offset Direction**.

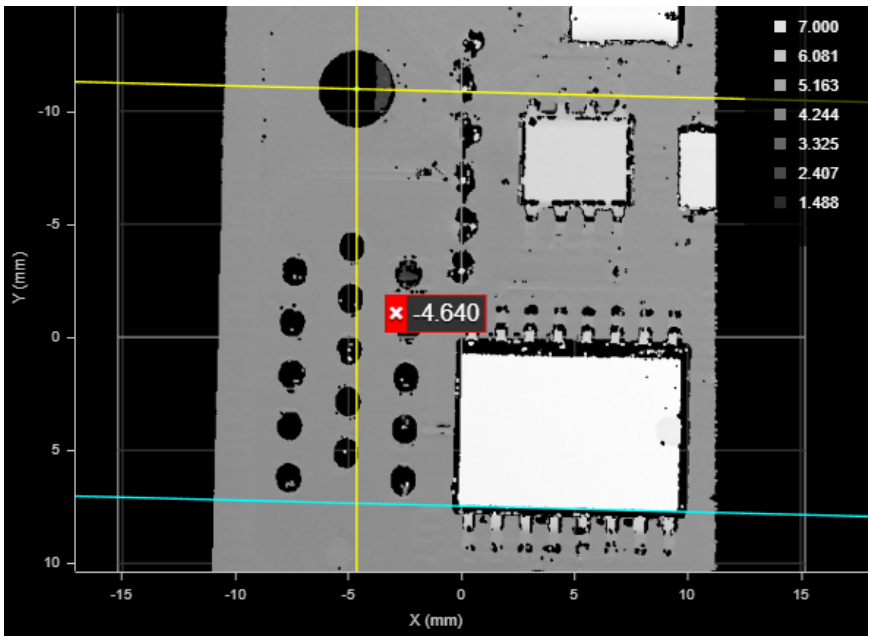
PERPENDICULAR OR PARALLEL LINE FROM POINT AND LINE

The **Perpendicular line from point and line** and **Parallel line from point and line** output types take a Point and a Line geometric feature as input to create another line.

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).



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 arbitrary values.

Line from Point and Line Output Types - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	None.
Point The Point geometric feature the tool uses to create the line.	
Line The Line geometric feature the tool uses to create the line.	

PERPENDICULAR LINE FROM POINT TO PLANE

The **Perpendicular line from point to plane** output type generates a perpendicular line from a point up to a plane.

Perpendicular line from point to plane Output Types - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	None.
Point The Point geometric feature the tool uses to create the line.	
Plane The Plane geometric feature the tool uses to create the line.	

INTERSECT LINE OF TWO PLANES

The **Intersect line of 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.

Intersect line of two planes Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	None.
Plane 1, **Plane 2** The two Plane geometric feature the tool uses to create the line.	

PROJECTED LINE ON PLANE

The **Projected line on a plane** output type takes a Line and a Plane geometric feature to create a line projected onto a plane.

Projected line on a plane Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	**Projection Direction** One of the following: **Perpendicular to Plane** or **Along Z Axis**.
Line The Line geometric feature the tool uses to create the	

Inputs	Parameters
line. **Plane** The Plane geometric feature the tool uses to create the line.	

PROJECTED CENTER LINE FROM TWO LINES AND A PLANE

The **Projected Center Line from Two Lines and a Plane** output type takes creates a two Line and a Plane geometric feature as input and creates a line.

Projected center line from two lines and a plane - Inputs and Parameters

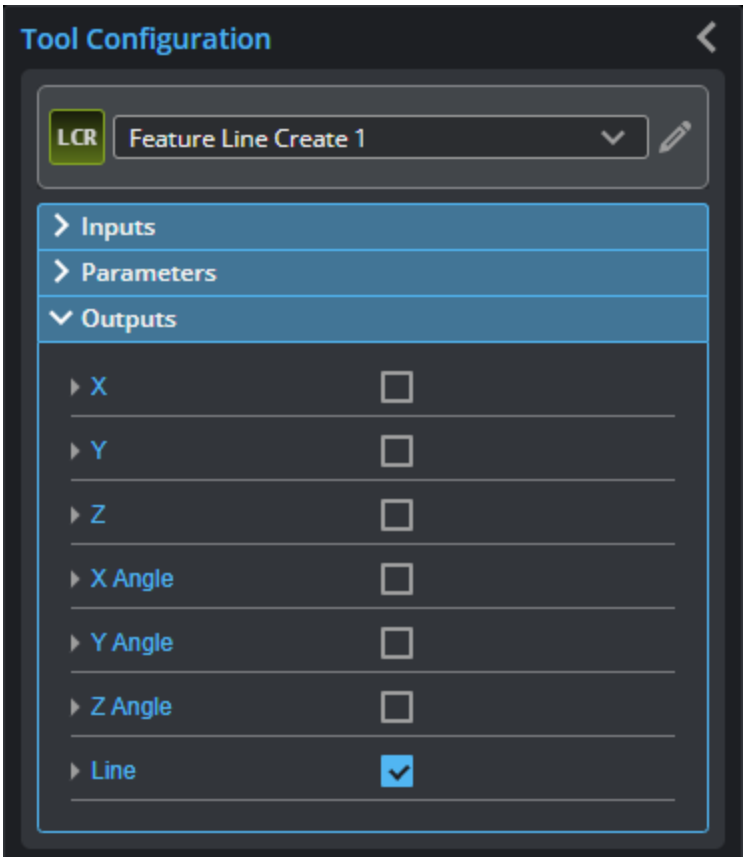
Inputs	Parameters
Input The Surface or Profile input. **Line** The Line geometric feature the tool uses to create the line. **Plane** The Plane geometric feature the tool uses to create the line.	None.

ROTATE LINE AROUND A POINT

This output type lets you choose a Line geometric feature and a Point geometric feature around which the line is rotated by the value you set in **Rotation** angle.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

For information specific to the different output types, see *Output Types* on page 273.

Measurements

Measurement
X, **Y**, **Z** The X, Y, and Z positions of the midpoint of the line described by the Point geometric features. For more information, see the sections above.
X Angle, **Y Angle**, **Z Angle** The X, Y, and Z angles of the line. For more information, see the sections above.

Features

Type	Description
Line	The generated Line geometric feature.

Feature Plane Create

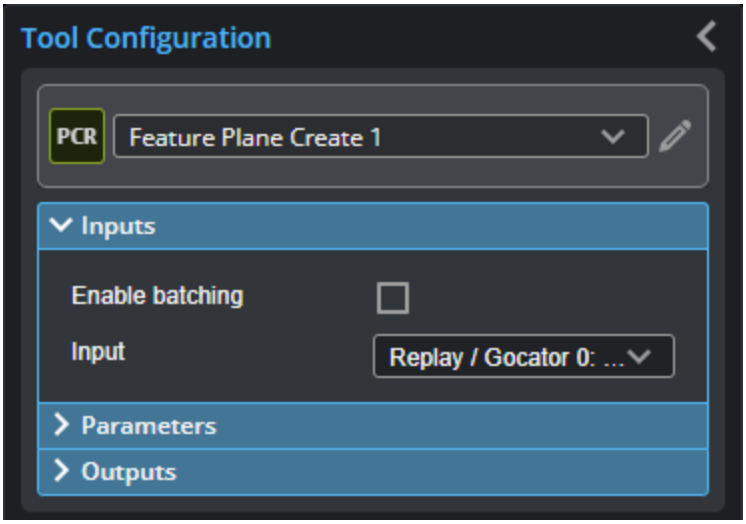
The Feature Plane Create tool lets you generate a Plane geometric features from geometric features generated by other tools. You can also generate a plane from arbitrary values you provide. The following outputs are available (for more information, see *Output Types* on the next page):

- Constant plane
- Plane from point and normal
- Plane from point and line
- Plane from three points
- Plane from circle
- Parallel plane from point and plane
- Perpendicular plane from point and plane
- Perpendicular plane from line and plane
- Bisect plane from two planes

For more information on geometric features, see *Geometric Features* on page 227.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

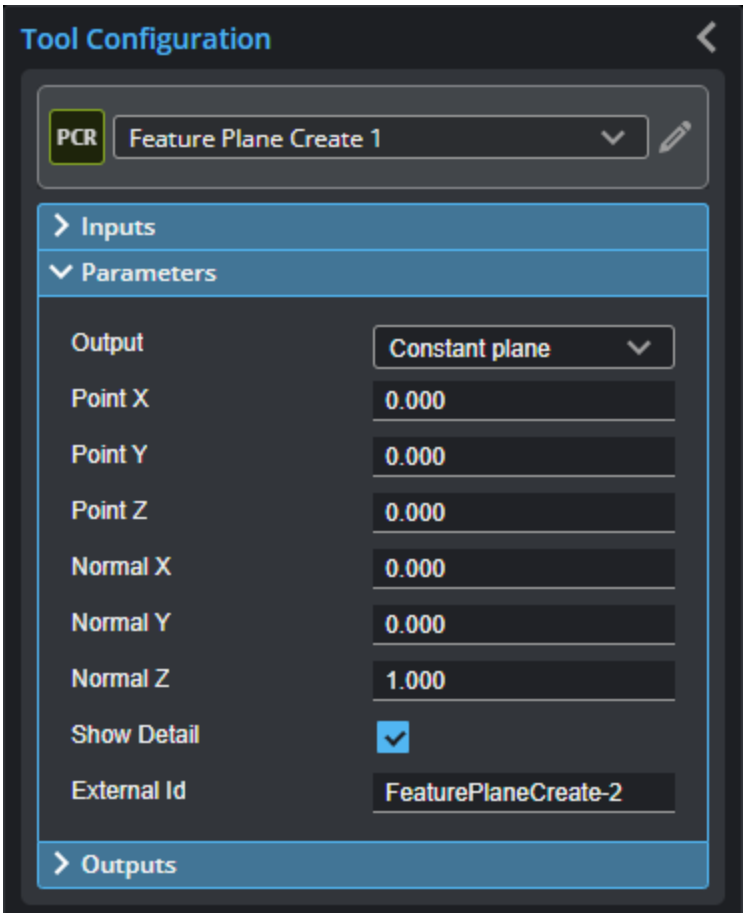


This tool can take an array as input. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.

The tool's inputs depend on the output type you choose in the **Output** parameter. For information on the required inputs and parameters of each output type, see *Output Types* on the next page.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Output	The type of output the tool generates. Switching between the options changes the input types displayed in Input section and the parameters displayed in the Parameters section. For information on the output types and the related parameters, see *Output Types* below.
Show Detail	Toggles the display of the input geometric features in the data viewer.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Output Types

The following sections describe the output types available in the tool's **Output** parameter, in addition to the required inputs and parameters.

CONSTANT PLANE

The **Constant plane** output type is useful if scan data from frame to frame is reliably fixed and you want to measure from an arbitrary plane to another feature. When you choose this output type, the

tool displays the parameters listed below you can manually fill in to create a constant plane.

Constant Plane Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	**Point X**, **Point Y**, **Point Z** The X, Y, and Z points that define the plane. **Normal X**, **Normal Y**, **Normal Z** The X, Y, and Z normals, respectively, of the constant plane.

PLANE FROM A POINT AND NORMAL

The tool uses the Line geometric feature as the normal of a plane, and positions the plane so that it intersects the Point geometric feature.

Plane from Point and Normal Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input. **Point** The Point geometric feature with which the plane intersects. **Line** A Line geometric feature representing the normal that the tool uses to create the plane.	None.

PLANE FROM A POINT AND LINE

The tool uses Line and Point geometric features to create a plane.

Plane from Point and Line Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input. **Point** The Point geometric feature with which the plane intersects.	None.

Inputs	Type-Related Parameters
Line A Line geometric feature with which the plane intersects.	

PLANE FROM THREE POINTS

The **Plane from three points** output type creates a plane that intersects the three Point geometric features passed to the tool as input.

Plane from Three Points Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	None.
Point 1, **Point 2**, **Point 3** The three Point geometric features the tool uses to create the plane.	

PLANE FROM CIRCLE

The **Plane from circle** output type creates a plane that intersects the Circle geometric feature passed to the tool as input.

Plane from Three Points Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	None.
Circle The Circle geometric features the tool uses to create the plane.	

PARALLEL PLANE FROM POINT AND PLANE

The **Parallel plane from point and plane** output type creates a plane parallel to the provided Plane geometric feature that intersects the provided Point geometric feature.

Parallel Plane from Point and Plane Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	None.

Inputs	Type-Related Parameters
Point The Point geometric feature with which the created plane intersects.	
Plane The Plane geometric feature with which the created plane is parallel.	

PERPENDICULAR PLANE FROM POINT AND PLANE

The **Perpendicular plane from point and plane** output type creates a plane perpendicular to the provided Plane geometric feature that intersects the two provided Point geometric features.

Perpendicular Plane from Point and Plane Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	None.
Point 1, **Point 2** The Point geometric features with which the created plane intersects.	
Plane The Plane geometric feature with which the created plane is perpendicular.	

PERPENDICULAR PLANE FROM LINE AND PLANE

The **Perpendicular plane from line and plane** output type creates a plane perpendicular to the provided Plane geometric feature, which intersects the provided Line geometric feature.

Perpendicular Plane from Line and Plane Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	None.
Line The Line geometric feature that bisects the created plane.	

Inputs	Type-Related Parameters
Plane The Plane geometric feature with which the created plane is perpendicular.	

BISECT PLANE FROM TWO PLANES

The **Bisect plane from two planes** output type creates a plane from the two provided Plane geometric features.

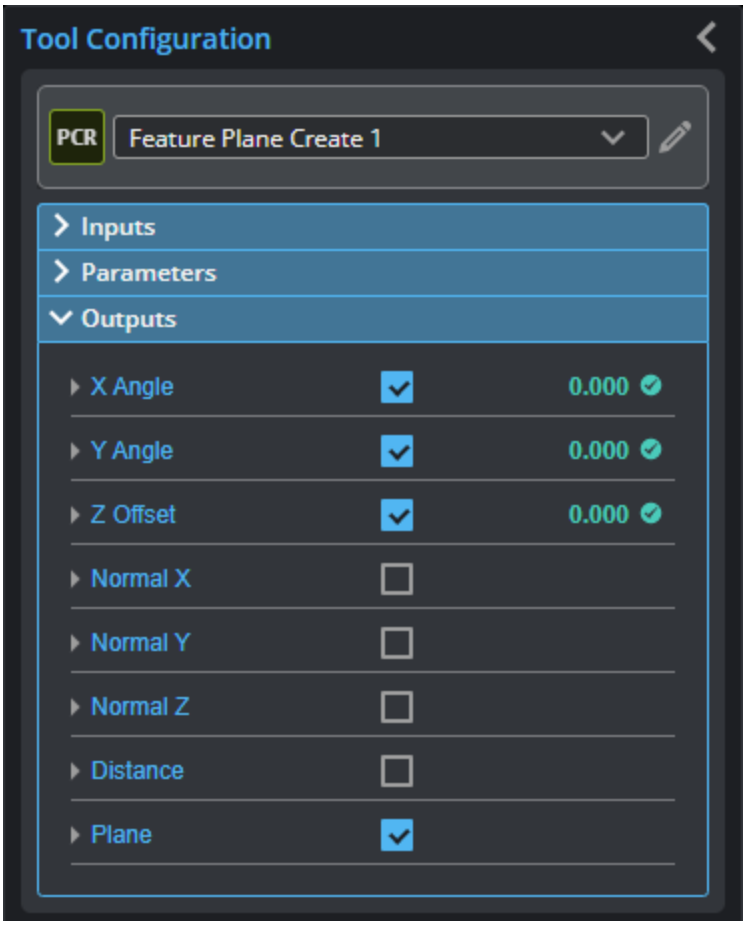
Bisect Plane from Two Planes Output Type - Inputs and Parameters

Inputs	Type-Related Parameters
Input The Surface or Profile input.	None.
Plane 1, **Plane 2** The Plane geometric features the tool uses to create a bisecting plane	

See *Adding a Tool* on page 164 for instructions on how to add measurement tools.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

For information specific to the different output types, see *Output Types* on page 281.

Measurements

Measurement
X Angle, **Y Angle** Rotation angles of the plane, that is, angles such that rotating (0,0,1) around the Y axis by the Y Angle, and then around X axis by X Angle produces the normal.

Measurement
Z Offset The Z value of intersection of the plane and the Z axis.
Normal X, **Normal Y**, **Normal Z** The normal X, Y, and Z vectors of the plane.
Distance The distance from the origin to the plane.

Features

Type	Description
Plane	The generated Plane geometric feature. For more information, see *Output Types* on page 281.

Feature Point Create

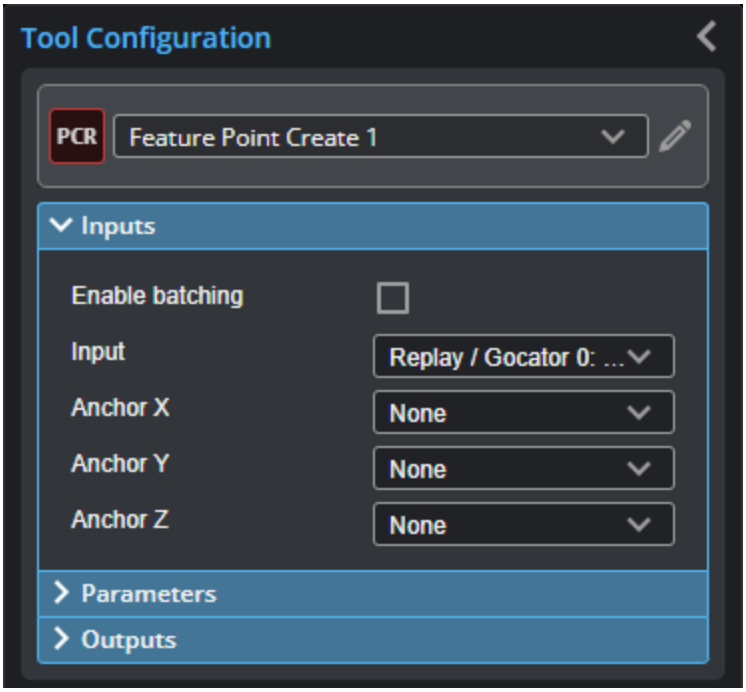
The Feature Point Create tool lets you generate a Point geometric features from geometric features generated by other tools. You can also generate a point from an arbitrary position you provide. The following outputs are available (for more information, see *Output Types* on the next page):

- Point
- Constant point
- Point from offset
- Point from three planes
- Point from line and circle
- Point from line and line
- Point from line and plane
- Projected point on line
- Projected point on plane

For more information on geometric features, see *Geometric Features* on page 227.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

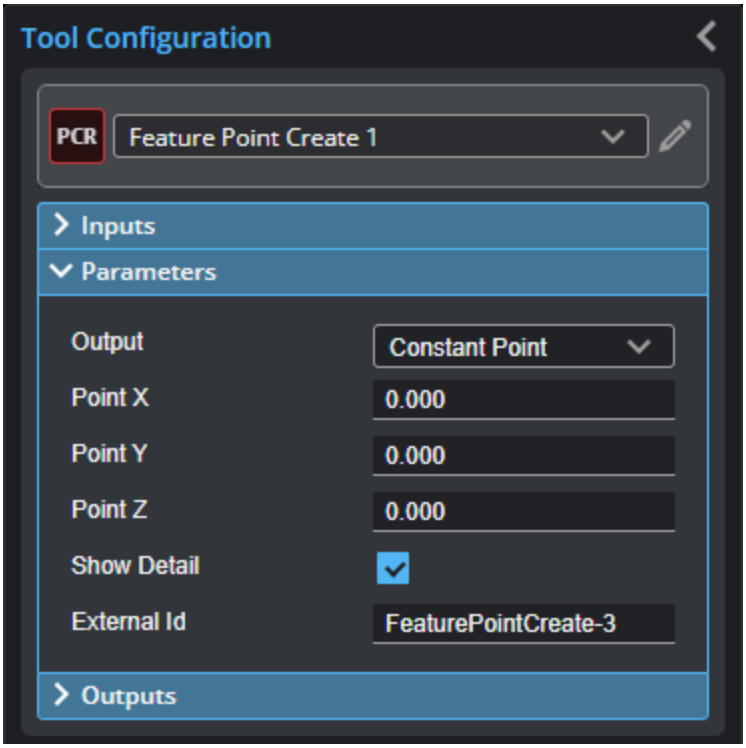


This tool can take an array as input. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.

The tool's inputs depend on the output type you choose in the **Output** parameter. For information on the required inputs and parameters of each output type, see *Output Types* on the next page.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Output	The type of output the tool generates. Switching between the options changes the input types displayed in Input section and the parameters displayed in the Parameters section. For information on the output types and the related parameters, see *Output Types* below.
Show Detail	Toggles the display of the input geometric features in the data viewer.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Output Types

The following sections describe the output types available in the tool's **Output** parameter, in addition to the required inputs and parameters.

POINT

The **Point** output type takes a Point geometric feature as input. This output is useful if the tool takes a feature generated by another Feature Create tool as input, on which you want to perform measurements in a second Feature Create tool. This tool 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.

Point Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	None.
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z position measurements the tool uses as anchors.	
Point The Point geometric feature the tool uses to create another Point geometric feature.	

CONSTANT POINT

The **Constant point** output type displays parameters you can manually fill in to create a geometric feature. This output type is useful if the scan data from frame to frame is reliably fixed and you want to measure from an arbitrary point to another feature.

Constant Point Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	**Point X**, **Point Y**, **Point Z** The X, Y, and Z position, respectively, of the first end point of the constant line.
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z position measurements the tool uses as anchors.	

POINT FROM OFFSET

The **Point from offset** output type takes two Point geometric feature as input, and generates a point on the line connecting the two input Point features. You specify the offset direction and the distance along the line.

Point from Offset Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	**Offset Direction** The direction along the line running through the two input Point geometric features that the point is offset.

Inputs	Parameters
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z position measurements the tool uses as anchors. **Point 1**, **Point 2** The Point geometric features the tool uses to create the line along which the offset Point geometric feature is placed.	One of two values: Forward or Backward. A Forward offset direction offsets the point along the line moving from Point 1 to Point 2. A Backward offset direction offsets the point along the line moving from Point 2 to Point 1. **Offset Distance** The distance along the line running through the Point geometric features that the point is offset.

POINT FROM THREE PLANES

The **Point from three planes** output type takes three Plane geometric features as input and generates a Point geometric feature at the intersection of those planes.

Point from Three Planes Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input. **Anchor X**, **Anchor Y**, **Anchor Z** The X, Y, and Z position measurements the tool uses as anchors. **Plane 1**, **Plane 2**, **Plane 3** The Plane geometric features the tool uses to generate the Point geometric feature. The point is at the intersection of the planes.	None.

POINT FROM LINE AND CIRCLE

The **Point from line and circle** output type takes a Line and a Circle geometric feature as input and generates a Point geometric feature where the line intersects the circle. The tool uses an intersection option to decide which intersection point to use.

Point from Line and Circle Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input. **Anchor X**, **Anchor Y**, **Anchor Z**	**Intersection Option** The intersection between the line and the circle that the tool uses to generate the Point geometric feature. One of the following: Upper, Lower, or Middle.

Inputs	Parameters
The X, Y, and Z position measurements the tool uses as anchors. **Line** The Line geometric feature the tool uses to generate the Point geometric feature. **Circle** The Circle geometric features the tool uses to generate the Point geometric feature.	**Upper**: The uppermost intersection point in the data viewer. **Lower**: The lowermost intersection point in the data viewer. **Middle**: The midpoint between the two intersection points.

POINT FROM LINE AND LINE

When the **Output** parameter is set to **Point from line and line**, the tool takes two Line geometric feature as input and generates a Point geometric feature where the lines intersect. The tool also returns an angle measurement for the angle between the lines.

Point from Line and Line Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input. **Anchor X**, **Anchor Y**, **Anchor Z** The X, Y, and Z position measurements the tool uses as anchors. **Line 1**, **Line 2** The Line geometric features the tool uses to generate the Point geometric feature.	**Angle Range** The range returned by the XY Angle measurement. One of the following: "-180 to 180" or "0 to 360".

POINT FROM LINE AND PLANE

When the **Output** parameter is set to **Point from line and plane**, the tool takes a Line and a Plane geometric feature as input and generates a Point geometric feature where the lines intersect. The tool also returns an angle measurement for the angle between the lines.

Point from Line and Plane Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	**Angle Range** The range returned by the Angle measurement. One of the following: "-180 to 180" or "0 to 360".

Inputs	Parameters
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z position measurements the tool uses as anchors.	
Line The Line geometric features the tool uses to generate the Point geometric feature.	
Plane The Plane geometric features the tool uses to generate the Point geometric feature.	

PROJECTED POINT ON LINE

The **Projected point on line** output type takes a Point and a Line geometric feature as input and generates a Point geometric feature.

Projected Point on Line Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	None.
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z position measurements the tool uses as anchors.	
Point The Point geometric features the tool uses to generate the Point geometric feature.	
Line The Line geometric features the tool uses to generate the Point geometric feature.	

PROJECTED POINT ON PLANE

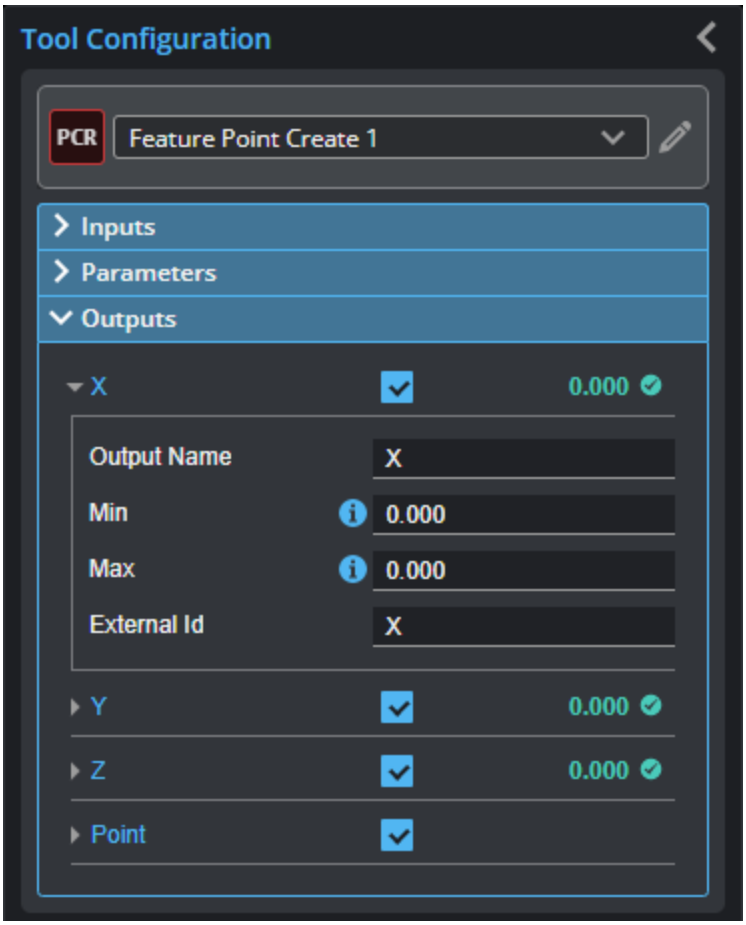
The **Projected point on plane** output type takes a Point and a Plane geometric feature as input and generates a Point geometric feature.

Projected Point on Plane Output Type - Inputs and Parameters

Inputs	Parameters
Input The Surface or Profile input.	**Projection Direction** The direction of the projection. One of the following: Perpendicular to Plane or Along Z Axis.
Anchor X, **Anchor Y**, **Anchor Z** The X, Y, and Z position measurements the tool uses as anchors.	
Point The Point geometric features the tool uses to generate the Point geometric feature.	
Plane The Plane geometric features the tool uses to generate the Point geometric feature.	

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

For information specific to the different output types, see *Output Types* on page 289.

Measurements

Measurement
X, **Y**, **Z** The X, Y, and Z positions of the Point geometric feature. For more information, see the sections above.
XY Angle The angle measurement.

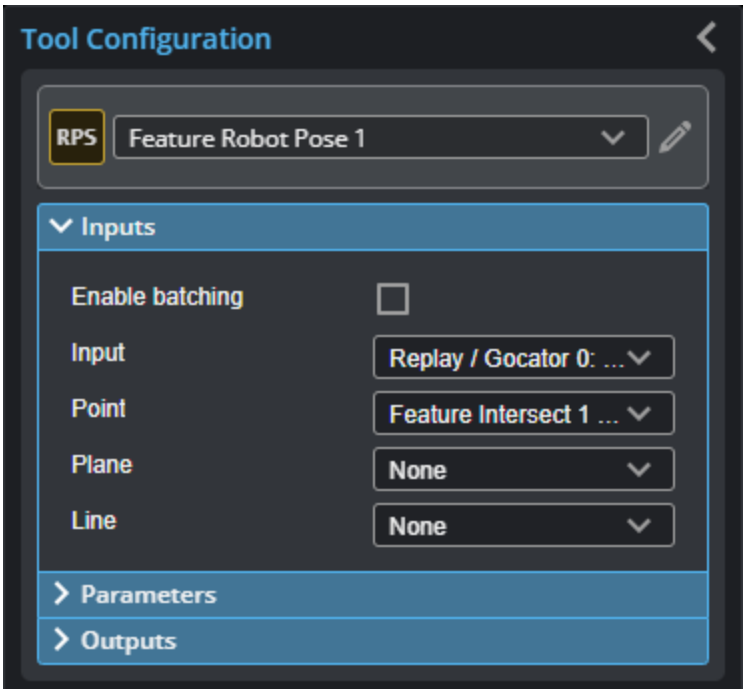
Features

Type	Description
Point	The generated Point geometric feature.

Feature 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.

Inputs



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

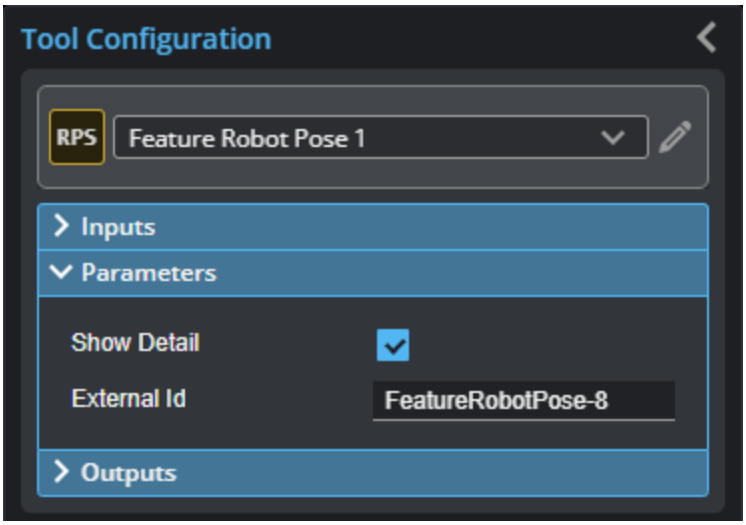
Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Input	The data the tool applies measurements to or processes.
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

Name	Description
	from. This input is optional. If you omit it, the tools uses the XY plane for calculations.
Line	The Line geometric feature the tool extracts the Yaw measurement from. This input is optional. If you omit it, the tool uses the X axis for calculations.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

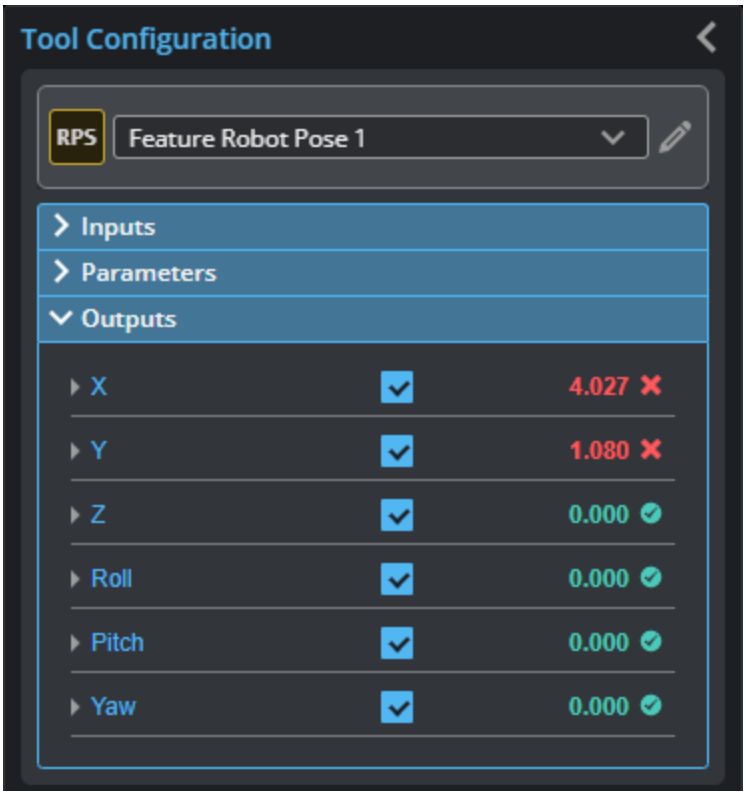


Parameters

Parameter	Description
Show Detail	Toggles the display of additional visualizations in the data viewer.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



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.	

Image Tools

The following sections describe the Image tools.



The Image tools are only available on Gocator 2D Smart Cameras and GoMax Smart Vision accelerators, and in GoPxL for PC. To be able to use the Image tools with this sensor family, you need a GoPxL Pro Tools license and dongle.

You also need to use the Surface Image Converter tool to convert either Surface data or intensity date to the Image format that the Image tools require. For more information on the Surface Image Converter, see *Surface Image Converter* on page 883.



On color sensors, all Image tools except Image Intensity convert color pixels in images to a monochrome value internally for processing and measurement.

When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value:

```
pixel = 0.299R + 0.587G + 0.114B
```

When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run.

When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.

Image Anomaly Detector

> This tool requires a special dongle and license to function with live sensor data. (Available for purchase from LMI.) You must also install the dongle's supporting CodeMeter runtime to license the tool, which you do in GoPxL Manager. For more information, see *Installing Licensing Software* on page 1195.

To exceed the capabilities of GoMax ORIN in terms of training times and inference, an RTX 4070 or higher is recommended.

If you choose to run this tool on a PC instance of GoPxL, the PC must have a graphics card with a recent Nvidia GPU that supports Nvidia GeForce Game Ready Driver version 576.57 or higher. To verify the driver version of your video card, launch the NVIDIA Control Panel, and in the main panel of the application, make note of the driver version.

The GoPxL Anomaly Detector toolset, which is part of LMI's GoPxL Pro Tools offering, lets you train and deploy AI-powered anomaly detection in features and parts in image data acquired by a compatible Gocator 2D Smart Camera.

You can download the Pro Tools package for PC, which contains the toolset (`14658-x.x.x.x_GoPxL_Pro_Utilities.exe`), from the downloads area of the LMI Technologies website: https://lmi3d.com/product-downloads/. Run the installer, and provide a location where the content of the package will be unzipped.

The toolset includes the following:

- The Image Anomaly Detector tool: This tool provides an AI-based model training routine and an inference engine that uses a trained model to predict whether something in production image data is an anomaly. The training routine steps you through the process of training on example images you must upload. To train a model, the tool requires a set of "good" (OK) images, roughly 15 to 20 images typically. Optionally, you can include "bad" (NG) example frames to help the tool set a good threshold in the model. The tool can also simulate part orientation and acquisition variations, as well as synthetic NG frames to improve model performance.
- The Training Archive tool: An auxiliary tool that conditionally saves frames to a project created in the Anomaly Detector tool, letting you improve the model on scan data acquired at a later time. For information on the Training Archive tool, see *Training Archive* on page 742.

Sample datasets are available in a package on LMI's Scenario Download page (https://lmi3d.com/scenario-downloads/).

If you are using this tool on a PC, you must install the CodeMeter runtime. For more information, see *Pro Tools* on page 113.

The following hardware and software is required to use the GoPxL Anomaly Detector in production:

1. A compatible Gocator 2D Smart Camera.
2. A GoPxL Anomaly Detector license and dongle, inserted into a USB port on the GoMax unit or PC.



A license and dongle are required for running the tool on live scan data. Without the dongle, you can only evaluate the tool on previously acquired images. The tool will also display an error message without a dongle.

Plugging in the license dongle will license the tool. If the camera system is running when you insert the dongle (that is, acquiring images), you must stop the system before it will license the tool.

3. Optionally, a GoMax NX, GoMax ORIN, or GoMax ORIN+ accelerator, depending on scan resolution, cycle time and pre- or post-processing requirements.

4. If you are running GoPxL Anomaly Detector on a PC, a version of the CodeMeter runtime that is compatible with the dongle and license is required.

   You install the runtime in GoPxL Manager; for more information, see *Running GoPxL on a Windows PC* on page 1188.

   To exceed the capabilities of GoMax ORIN in terms of training times and inference, an RTX 4070 or higher is recommended.If you choose to run this tool on a PC instance of GoPxL, the PC must have a graphics card with a recent Nvidia GPU that supports Nvidia GeForce Game Ready Driver version 576.57 or higher. To verify the driver version of your video card, launch the NVIDIA Control Panel, and in the main panel of the application, make note of the driver version.



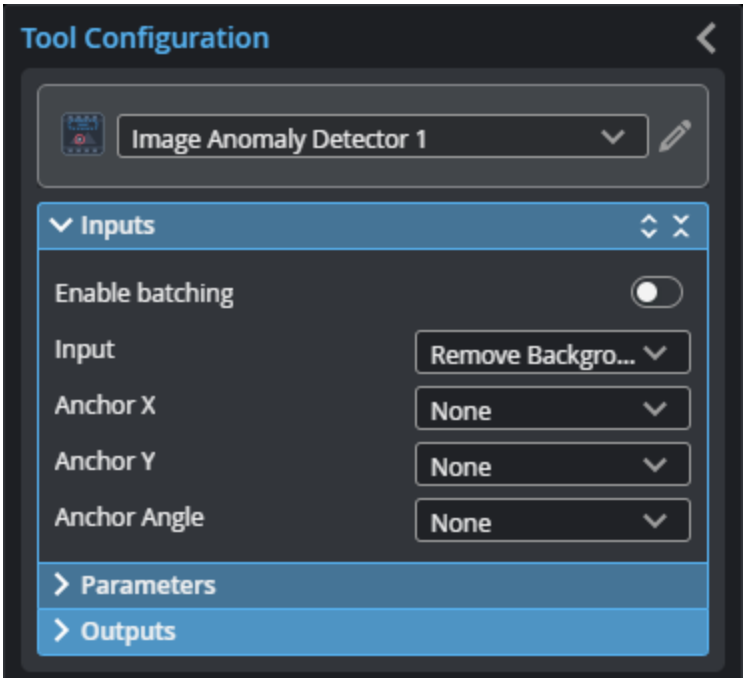
   Note the following general requirements for running GoPxL on a PC:

   - Processor: x86-64 (Core i3 or equivalent)
   - RAM: 8 GB
   - Hard drive: 64 GB

- Operating system: Windows 10 or 11 (64-bit only)
- Display: 1366 x 768

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

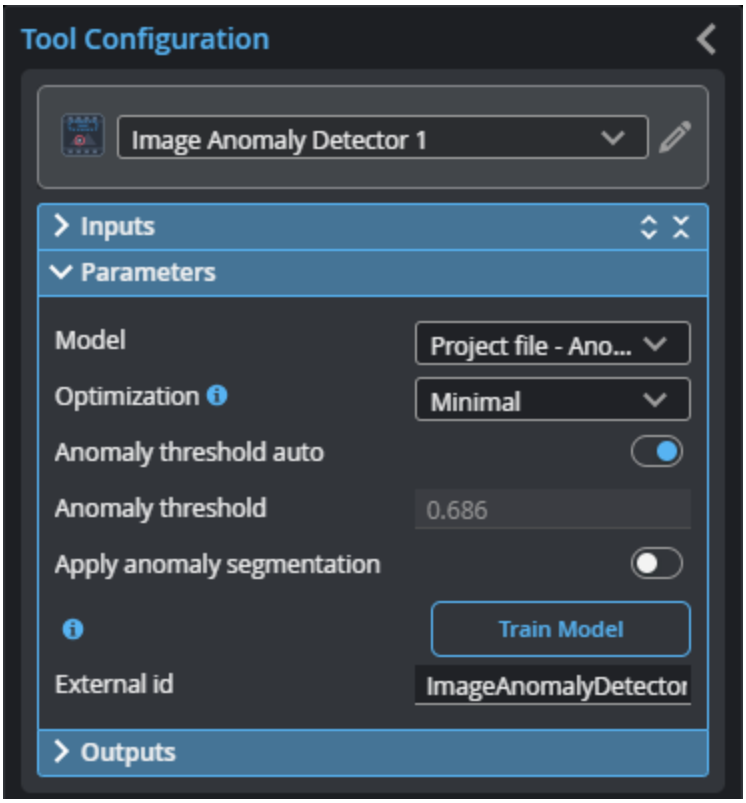
Inputs

Name	Description
Enable Batching	Batching (when a tool takes an array as input and processes each element in the array individually) is currently not fully supported by this tool. If you choose to use batching, keep the following limitations in mind: • The Label page only shows the first element of the input array. • For anomaly detection, blobs representing the anomalies are not displayed. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Name	Description
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

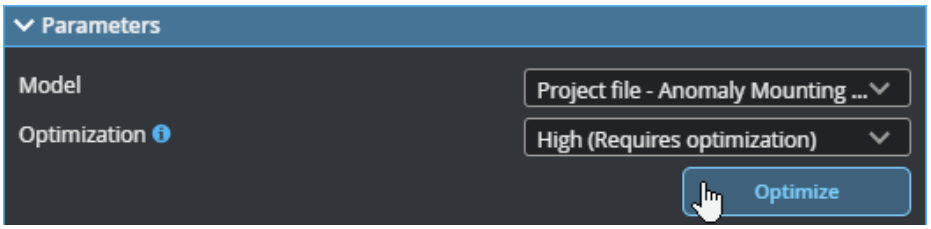
Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Model	The trained model the tool uses to detect anomalies during production.
Optimization	Determines how much optimization is applied to the model during inference. A higher setting provides better performance, One of the following: Minimal, Balanced, High. When you set to this to "High", GoPxL may need to re-optimize the model. Optimization may take several minutes.

Parameter	Description
	
Anomaly threshold auto	When this setting is enabled, the tool uses a threshold calculated during training. When it is disabled, provide a threshold in **Anomaly threshold**.
Apply anomaly segmentation	When you set **Model** to a previously trained model, this parameter is available. When you enable this parameter, configure the parameters in **Segmentation parameters** (see below). The tool produces additional outputs when this is enabled; for more information, see *Anomaly Segmentation Outputs* on page 319.
Segmentation parameters	The parameters that control anomaly output. **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 anomalies 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. **Use area filter** If **Use Area Filter** is enabled, the tool applies an area filter to anomalies using the values you provide in **Max area** and **Min area**. Use the filter to ignore small features such as debris or dust or large irrelevant regions. **Ordering** Orders the measurements, features, and surface data of the individual anomalies output by the tool. One of the following: • Position - X increasing

Parameter	Description
	• Position - X decreasing • Position - Y increasing • Position - Y decreasing • Area - Large to small • Area - Small to large
Train Model	Starts the model training routine, which steps you through process. For more information, see *Train Model* on the next page.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Development Workflow

The high level workflow is as follows:

1. Collect representative images of the features or parts you need to inspect in order to train an Anomaly Detector model. In addition to "good" parts, you can scan "bad" parts if these are available, that is, ones with typical defects, to help set a good threshold.
2. Create a pre-processing toolchain that excludes unimportant background and train an Anomaly Detector model.
3. Deploy the Anomaly Detector model and connect post-processing tools.
4. Optionally, collect new production data for model retraining.

Collect dataset

Unless you are using a previously created dataset, you must create the dataset that the GoPxL Anomaly Detector will use for training a model before it can detect anomalies in new production data. To do this, perform the following steps:

1. Record representative scans of 20 or more OK targets (targets with no flaws) and 20 or more NG targets (targets with flaws).

   You can record the scans to a single .gprec file or to multiple files. If you record scans to a single file, grouping each type (OK and NG) together will make it easier to label them later in the wizard. Make sure there are no empty frames: each frame should contain either an OK or an NG target. Do not rely on .gpsup or .rec for this as the Anomaly Detector wizard requires .gprec files when importing from PC.
2. Save a separate .gpsup file with the sensor configuration settings such as exposure time and related imaging parameters.

   Ensure the same parameters are used on the production system. Otherwise, you will need to collect new training data.

Implement a Preprocessing Toolchain

The preprocessing tool chain limits variation in the image data to the region of interest (with and without anomalies). With less variation in the image data, the total number of frames needed for

training is reduced.

The preprocessing tool chain must not change between training and inference. Otherwise, the system will not function properly. You must save your preprocessing toolchain in the .gpsup file created previously or using a separate .gpjob file.

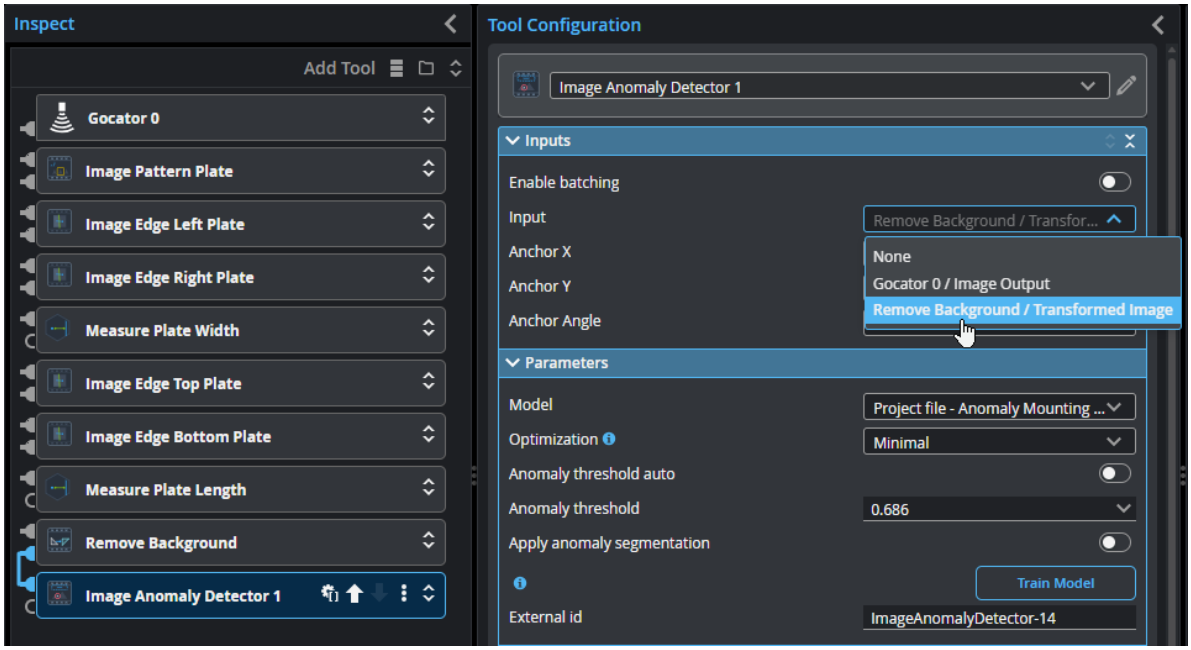
> You *must* use the same preprocessing toolchain for training and inference to ensure proper detection of anomalies.

Train Model

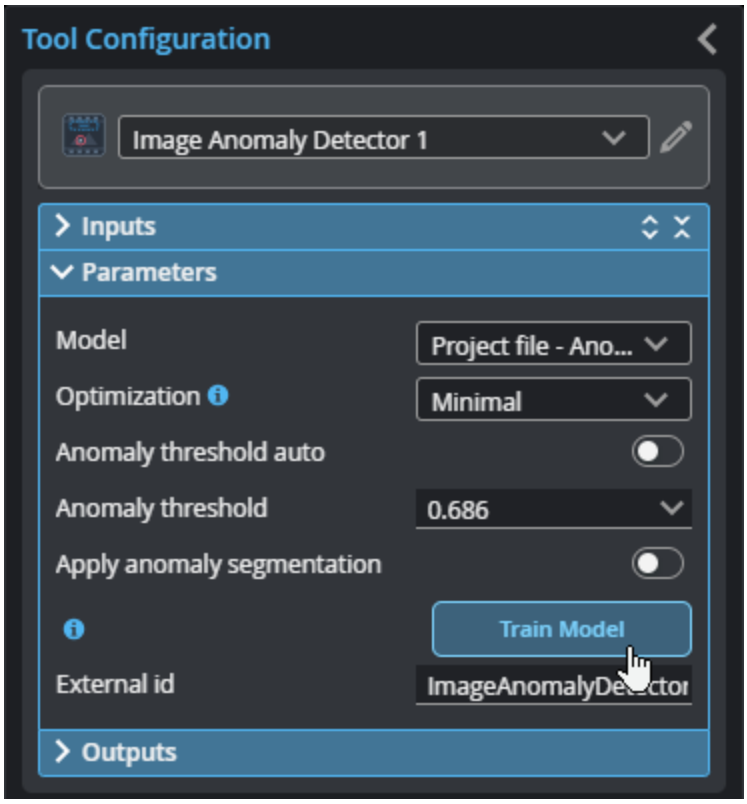
ADD IMAGE ANOMALY DETECTOR TOOL

1. Add and connect an Image Anomaly Detector tool to the output of the preprocessing tool chain.

   In the following, the first four tools in the Tool Diagram on the left are the preprocessing tool chain. The last tool in the chain, Surface Filter, is used as input for the Anomaly Detector tool.



2. In the tool's **Parameters** section, click the **Train Model** button.

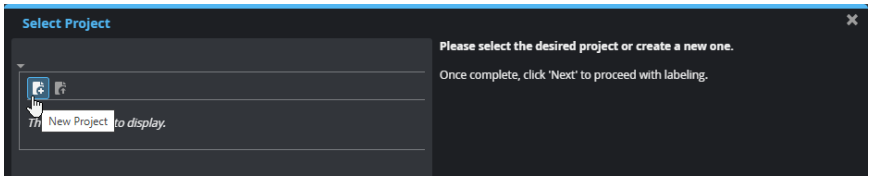


The routine starts and the **Select Project** page opens. You will create a new project in this page, or open a previously created project.

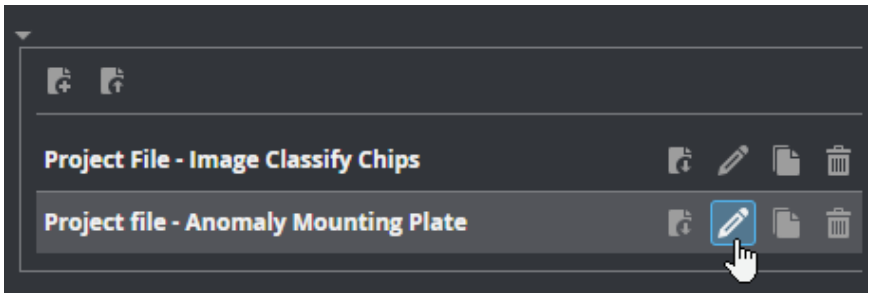
CREATE NEW OR OPEN PROJECT

A project is a collection of frames used for training, labels you apply to those frames (OK and NG), and the models trained on those frames. You should choose a meaningful project name, such as the part name, especially when working with multiple projects.

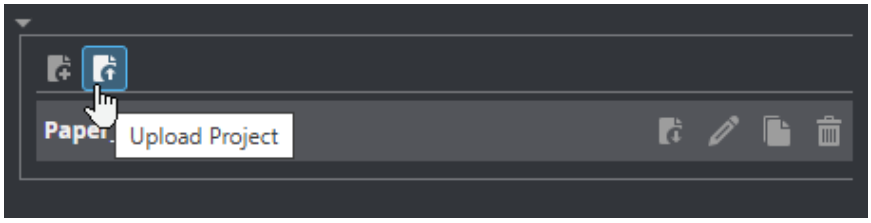
1. On the **Select Project** page, click the New Project icon, provide a project name, click **Continue**, and then click **Next**.



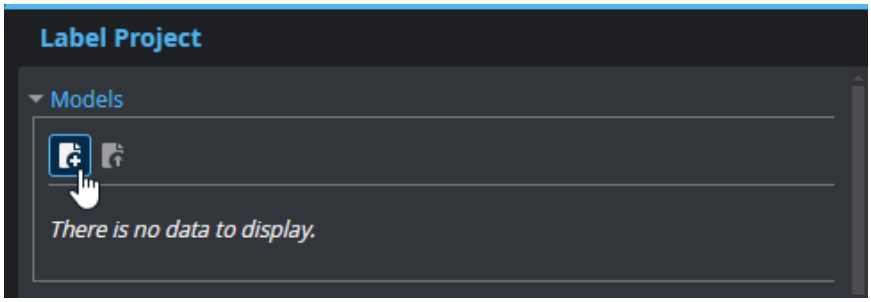
You can use the download icon on a project's row to download this to your PC and upload to another device.

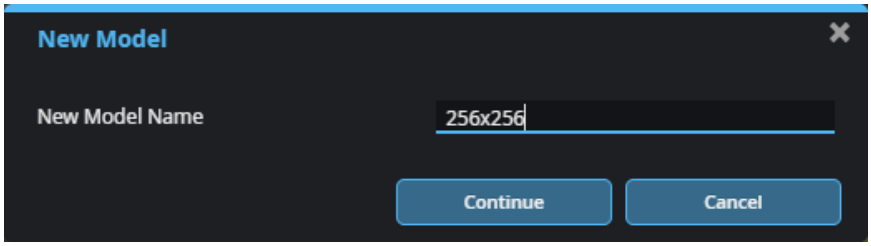


You can use the upload icon to import a previously downloaded project.

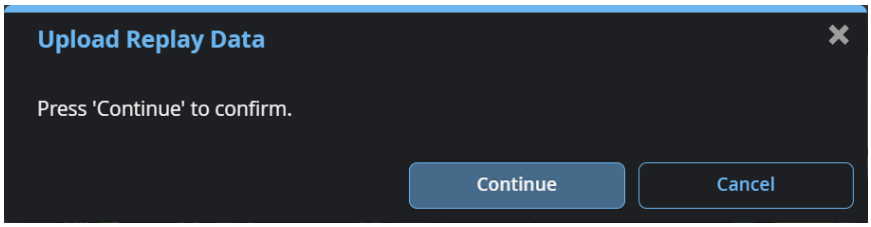


2. In the **Label Project** page, in the **Models** section, create a new model and click **Continue**.





3. In the **Frames** section, do one of the following
   - To upload currently loaded replay data, click the Upload Replay Data button ( ) , and then click **Continue**.



   - To upload the data from a .gprec file on the local PC, click the Upload From Disk button ( ), choose a file in the Open dialog a, click Open.

The tool loads the frames into the model.

> ⚠ Loading replay data into the project can take several minutes. If a "Delayed Response" message displays, click "Wait." In this case, add a Decimation Filter to the pipeline.

Note that you can also use the Replay Editor to upload image files from a PC to create a dataset; for more information, see *Recording and Working with Scan Data* on page 104.

LABEL FRAMES AND CHOOSE SET

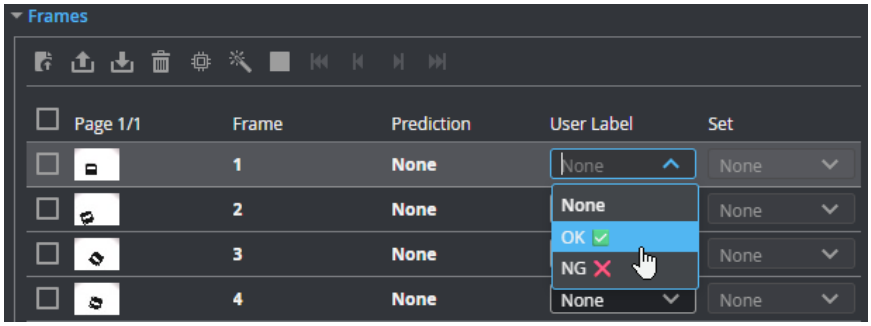
You must now label each frame as "OK" or "NG" (under the **User Label** column). You must also choose the set each frame belongs to (Train or Test, under the **Set** column). By default, the tool automatically places frames in the Train or Test set, using the proportions listed below. You can disable auto-split, but you should use the same proportions.

80% of OK is in Train

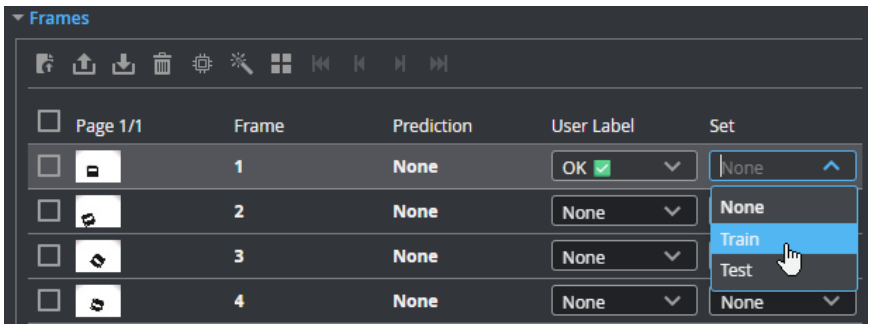
20% of OK is in Test

100% of NG is in Test

1. In the row of each frame, under **User Label**, choose the correct label.



If you have not enabled auto-split, also choose the frame's set under **Set**. Make sure you use the proportions described in step .



To quickly choose the label and set for multiple frames, do one of the following to first select the frames:

- To select a range of frames, click the checkbox of the first frame, press Shift, and click the checkbox of the last frame of the range.
- To select multiple frames, click the checkbox of the first frame, press Ctrl, and click the checkbox for

each frame you want to select.

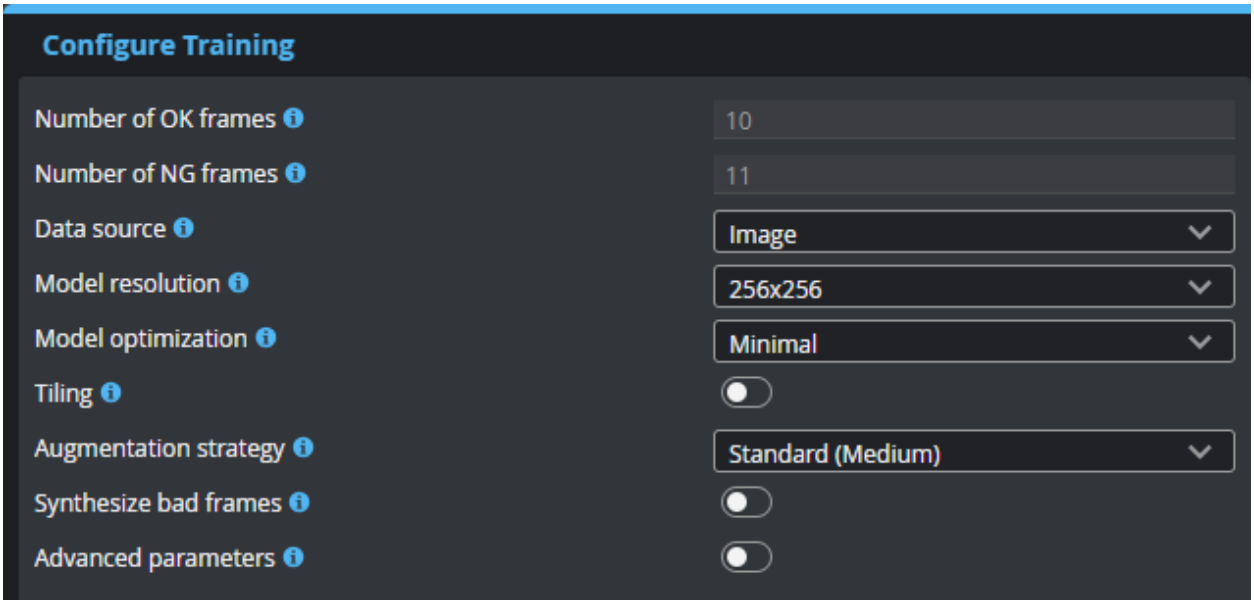
- To select non-contiguous ranges, alternate between clicking Ctrl and Shift.

After selecting the frames, choose the label and the set as described above. To deselect all frames, click anywhere between the checkbox and the User Label and Set drop-down menus.

2. Click **Train Model** to configure training.

CONFIGURE TRAINING

After labeling the frames and choosing the set they belong to, you must configure training. For descriptions of the settings, see below.

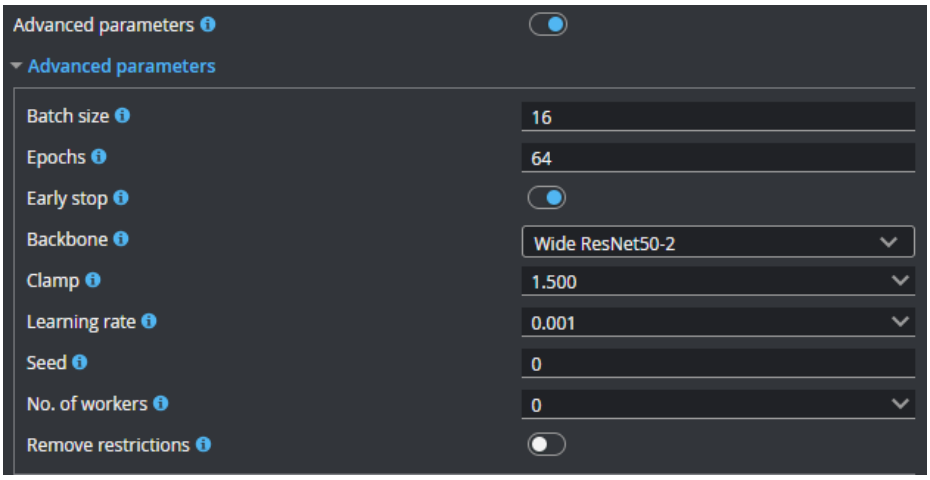


Training Parameters

Parameter	Description
Number of OK frames Number of NG frames	Read-only fields that display the number of frames you have labeled OK and NG, respectively.
Data source	The source the tool uses during training.
Model resolution	Decimates the input data to the specified resolution for processing. A higher resolution improves anomaly location accuracy but increases processing time for training. For most applications, leave this at 256x256.
Model optimization	Determines how much optimization is applied to the model during inference. A higher setting provides better performance, One of the following: Minimal, Balanced, High.
Tiling	Enables tiling. Tiling divides the frames of training data into sections. The tool uses the same model for each tile, which increases the resolution of the model. This lets the tool potentially find smaller defects or increase detection accuracy. You should

Parameter	Description
	only use tiling with homogenous targets (for example, paper towel, wood decking, and so on) or symmetric targets. For discrete parts with a complex or asymmetric shape, disable this setting. After enabling tiling, set the number of vertical and horizontal tiles. You can also set tiles to overlap in **Tile margin**. Use this to detect anomalies that might fall on a boundary between tiles. Generally, you should use a value between 12.5% and 25%. Note that using tiling and tiling margins increases the memory requirements for training considerably. For example, with 5x5 tiling, the tool uses 25 times more memory.
Augmentation strategy	When **Augmentation strategy** is enabled, the tool generates random variations to simulate the kinds of variations encountered in production use. This improves the training data, which in turn improves the tool's ability to detect anomalies in production. The following options are available: **Precision (Low)**: Minimal transformations for controlled environments with fixed lighting and camera. **Standard (Medium)**: Moderate variations for typical factory conditions. **Robust (High)**: Aggressive distortion for high-glare or cluttered scenes. You can visualize example augmentations on the Model Performance page, which is available after training is complete; for more information, see *Test Model Performance* on page 315.
Synthesize bad frames	When this setting is enabled, the tool synthesizes bad (NG) frames from the good (OK) frames in the project. The frames are used internally during training and are not added to the frames in the project or shown in the tool. Enable this setting if your project contains very few NG frames. The setting is automatically enabled if the project contains *no* NG frames.
Advanced parameters	If the training results are generally good but the tool is reporting a small number of false positives or false negatives, you can modify these parameters to further improve results.

Parameter	Description



Batch size

The number of frames in a batch (a subset of the entire dataset passed to the training algorithm for an iteration).

If you are using a model with a 512x512 resolution or tiling and the tool has memory issues during training, decrease this value.

Epochs

The number of times the entire dataset passes through the training algorithm. To reduce false positives or negatives, increase this value. Note that training will take longer.

Early stop

Leave enabled for the initial training runs. To help model performance, disable to improve model performance; note that training will take longer.

Backbone

The feature extractor used by the tool. One of the following:

ResNet18: A lighter model that is fast and a good starting point.

Wide ResNet50-2: Slow and larger memory, with better for complex textures.

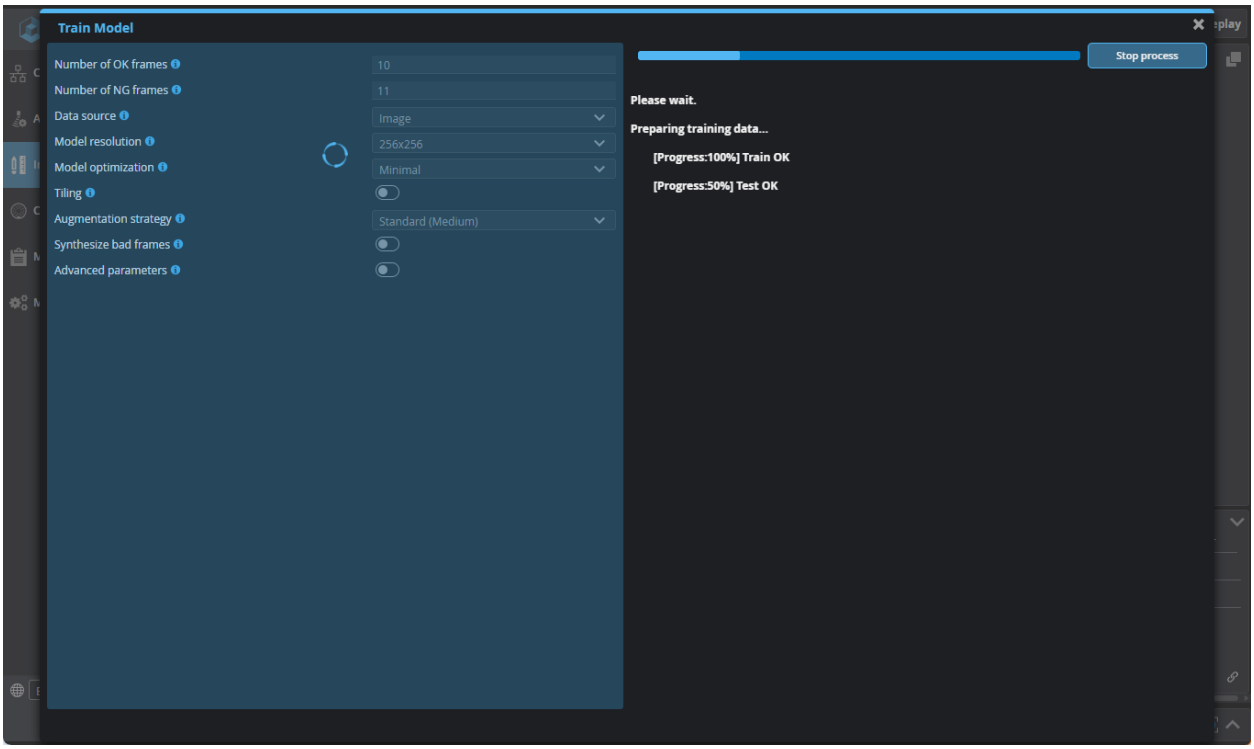
Clamp

The range within which the data values should be clamped. Reduce this value to help reduce false positives or false negatives.

Learning rate

Parameter	Description
	Set this to a higher value to reduce training time. Note that a value that is too high can cause the model to miss the point where the difference between predicted and actual OK / NG is at its smallest (that is, overshooting it). A lower value causes the model to learn more slowly, but potentially more accurately; if the value is too low, the model can get stuck. **Seed** Set this to a non-zero number to ensure your results are consistent and reproducible between training runs. Set the value to 0 use a random seed for each training run. **No. of workers** The number of worker threads dedicated to background image data preparation. Increasing the number of workers can improve data throughput and make sure the GPU isn't waiting for data, which may speed up training. However, it also increases CPU load, RAM usage, and sometimes overhead or instability if set too high on systems with a limited number of cores or memory. **Remove restrictions** By default, this option is disabled, and the following limitations are in place to ensure the system trains successfully and does not run out of memory: • The maximum number of tiles under the **Tiling** parameter is limited to 6. • The highest resolution in **Model resolution** is 512x512.

After you configure training, click **Start Training** to train the model. The tool will start by preparing the training data, providing progress feedback.



After the data is prepared, the tool will start generating Loss and AUROC graphs.

Loss: This graph represents the cumulative error for each frame in the Test set, determined after each training run iteration or epoch. A decreasing number (the graph's line is moving down) implies the model is improving.

AUROC: This graph defines how well the model can distinguish between the OK and NG user labels. A value of 1 (the graph's line is moving up) represents the best possible result.

As the tool trains the model on the labeled frames, the graphs should converge, that is, Loss will decrease, and AUROC will increase, approaching 1 (the best possible result). After the training has completed, click **Next** to test the model. Note that training can take several minutes.

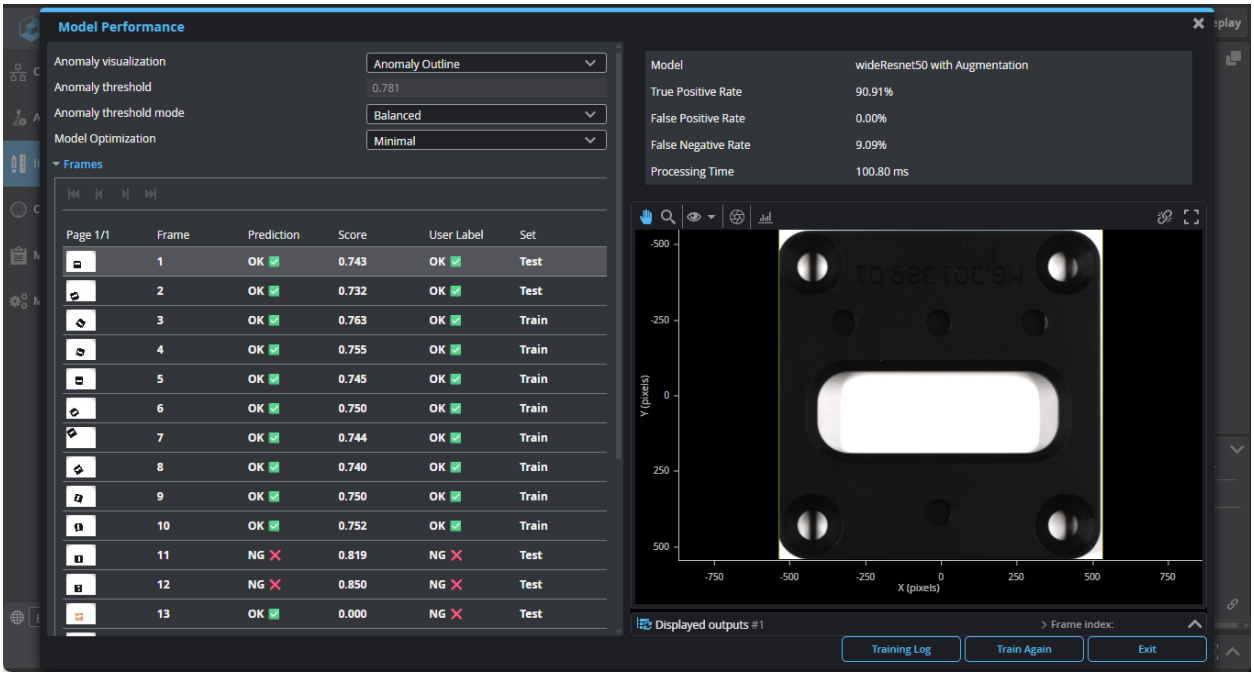
TEST MODEL PERFORMANCE

After the tool has finished training the model, it will use the model to make predictions on the frames to determine the anomaly score for each frame, as well as the anomaly threshold. The Model Performance page displays the following results:

Performance Results

Name	Description
Model	The trained model the tool used to detect anomalies during the model performance phase.
True Positive Rate	The percentage of frames correctly predicted as OK.
False Positive Rate	The percentage of frames incorrectly predicted as OK.

Name	Description
False Negative Rate	The percentage of frames incorrectly predicted as NG.
Processing Time	The average time required for inference. Includes the pre-processing tool chain.
Anomaly Threshold	The threshold to which a frame's Anomaly Score measurement is compared to determine whether the frame is an anomaly. An anomaly score above this value is an anomaly. (Displayed as a read-only field to the left of the wizard.)

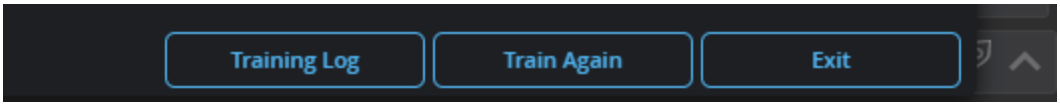


You can use the **Anomaly visualization** drop-down to view different visualizations of the selected frame:

Anomaly visualizations

Name	Description
Anomaly Outline	Displays the Image data of the frame with anomalies outlined, if present.
Heatmap	A diagnostic Image output overlay on the input data. This output can make it easier to see whether the brighter areas in the anomaly map correspond with flaws in the target.

The Model Performance page also lets you download an HTML file containing reports and other diagnostics. LMI recommends checking the visual results in this file.



You can also retrain the model with different training configuration if you wish.

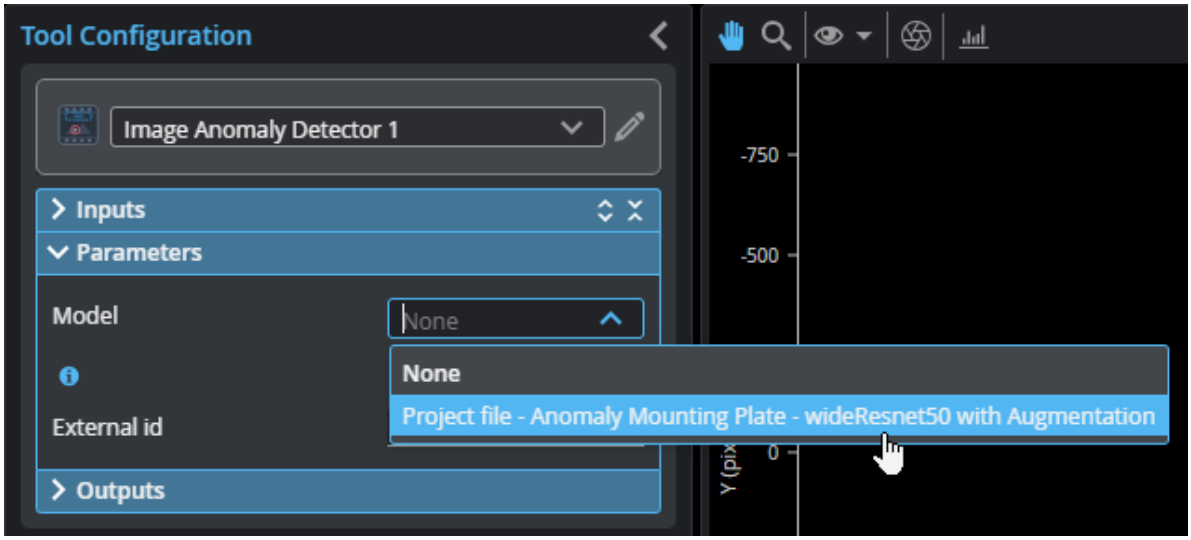
You are now ready to use the trained model on new scan data in production.

Deploy in production

ANOMALY TOOL SETUP

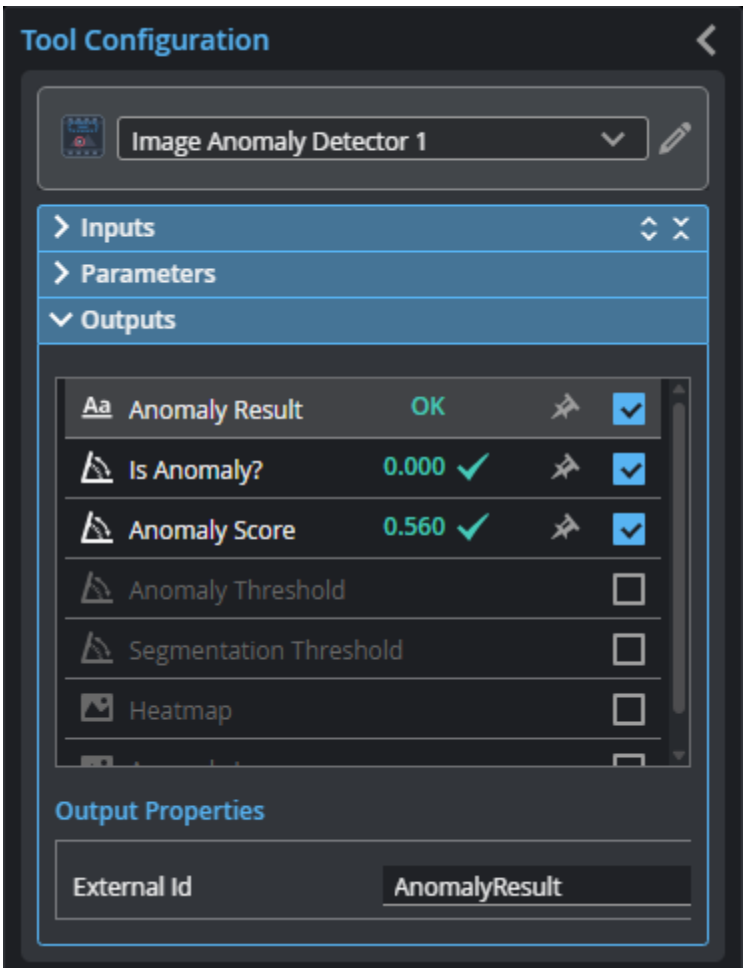
1. Connect to the device using a PC on the same network.
2. Upload a .gpsup or .gprec file created during the dataset collection.

   Ensure the recording contains the same preprocessing pipeline used during training.
3. Discover the Gocator sensor and create a new sensor group.
4. Insert the GoPxL Anomaly Detector USB dongle to a USB port on the GoMax.
5. In the Image Anomaly Detector tool, in **Model**, choose the model you trained and want to use in production.



6. Configure the outputs as required, and implement post-processing to isolate and prepare anomalies for measurement. For more information, see *Outputs* on the next page and *Image Anomaly Detector* on page 301.

Outputs



Although the tool has several outputs, the "Is Anomaly" measurement and "Anomaly Surface" output are the main outputs intended for processing by tools later in the tool chain.

To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Some outputs are only displayed if **Apply anomaly segmentation** is enabled in the **Parameters** section.

Measurements

Measurement
Anomaly Result A string representing the result: OK or NG, corresponding to 0 and 1, respectively, in the "Is Anomaly?" measurement.
Is Anomaly? Indicates whether the tool has detected an anomaly in the current frame of scan data.
Anomaly Score Indicates how strongly the tool believes the current frame contains an anomaly. When the value is higher than the Anomaly Threshold measurement, this measurement is set to 1.0.
Anomaly Threshold Threshold value (0..1) of the 'Anomaly Score' above which the model considers an input image anomalous. This value is determined during training to achieve the best possible detection rate.
Segmentation Threshold A grayscale value in the Image data above which the model considers input intensity data anomalous. The value is constant for a trained model.
Intensity Threshold Threshold value (0..1) of the 'Anomaly Score' above which the model considers an input image anomalous. This value is determined during training to achieve the best possible detection rate.

Data

Type	Description
Heatmap	Diagnostic Image output overlaying the Anomaly Map output on the input scan data. This output can make it easier to see whether the brighter areas in the anomaly map correspond with flaws in the target.
Anomaly Image	Contains the Image data of the frame, with the anomaly map (see Anomaly Map below) added as intensity data. The intensity data represents a per-pixel score of how anomalous that pixel is.

Anomaly Segmentation Outputs

The following outputs are only displayed when **Apply anomaly segmentation** is enabled.

Measurements

Measurement
Anomaly Count The number of anomalies detected in the frame.
Center X
Center Y

Measurement
Arrays containing the X and Y positions of the centers of the anomaly segments.
Width
Height
Arrays containing the width and height values of the anomaly segments.
Area
Array containing the areas of the anomaly segments.

Geometric features

Type	Description
Center Point	A point geometric feature representing the center of the segment (blob).

Data (with anomaly segmentation)

Type	Description
Blob Image	An image of the segment (blob).

Store production data

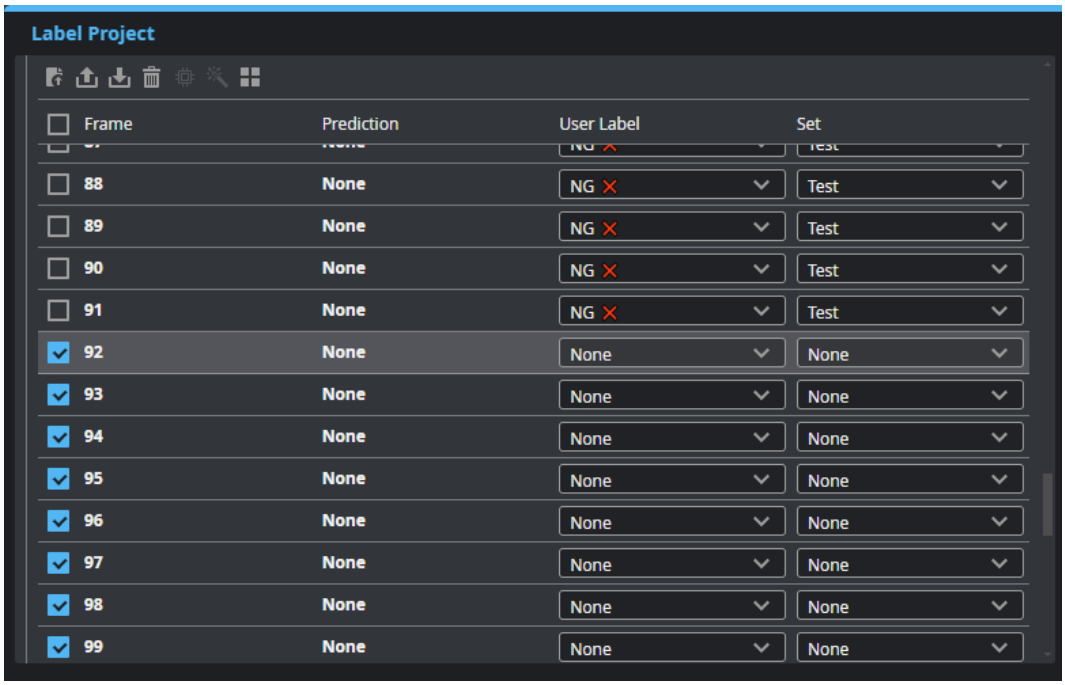
You can use the Training Archive tool to conditionally add frames to a Surface Anomaly Detector project. For more information, see *Training Archive* on page 742.

Train using additional data

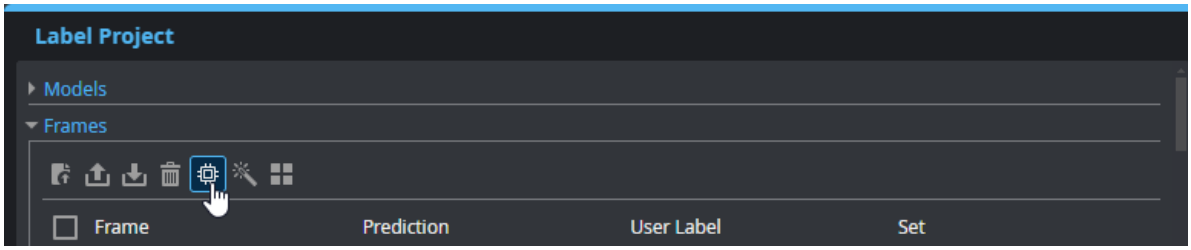
After adding new frames to a project using the Training Archive tool (see *Training Archive* on page 742) or uploading additional frames manually (from the replay buffer or from a .gprec file), you can further train the model with new frames, taking advantage of the trained model's prediction capabilities to automatically label frames as OK or NG.

To train using new data:

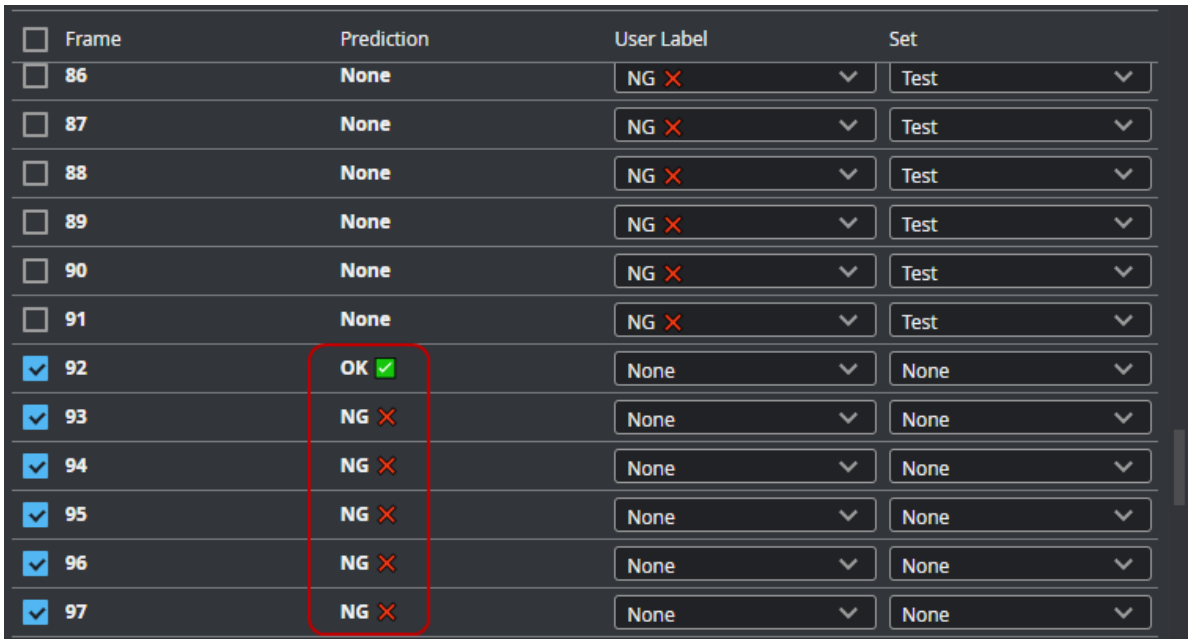
1. After adding frames to a project, go to the Label Project page.
2. Select the new frames.



3. Choose the correct model you want to use for prediction.
4. At the top of the Frames section, click the Generate Prediction button.

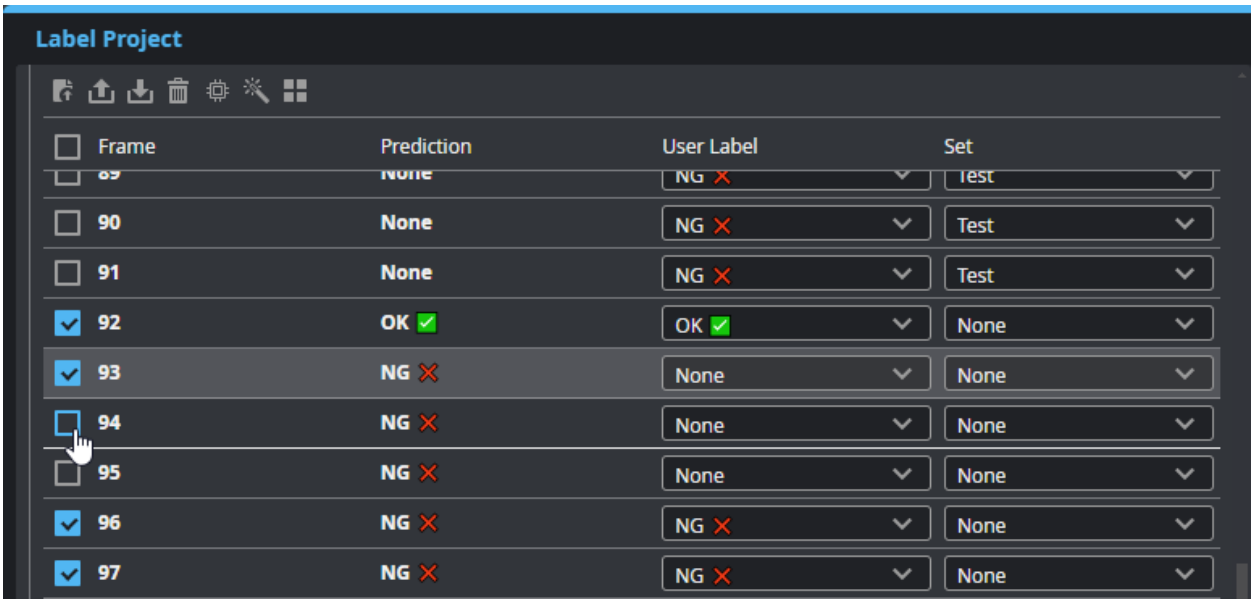


The tool uses the selected model to generate predictions (OK or NG) for the selected, unlabeled frames. Note that the user labels of the frames are unchanged at this point.

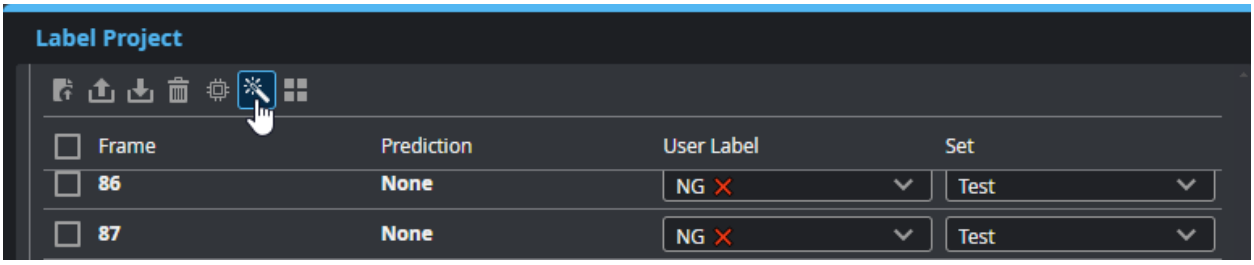


5. Review the predictions and do the following:

   a. Uncheck any invalid predictions.



   b. Click the Apply Prediction button.



c. Toggle Auto-Split to On to automatically set the Set of the selected frames.

d. Manually set the user label of any incorrectly predicted frames.

The predictions of the selected frames are copied to the user label drop-down.

6. Retrain the model with the newly added frames.

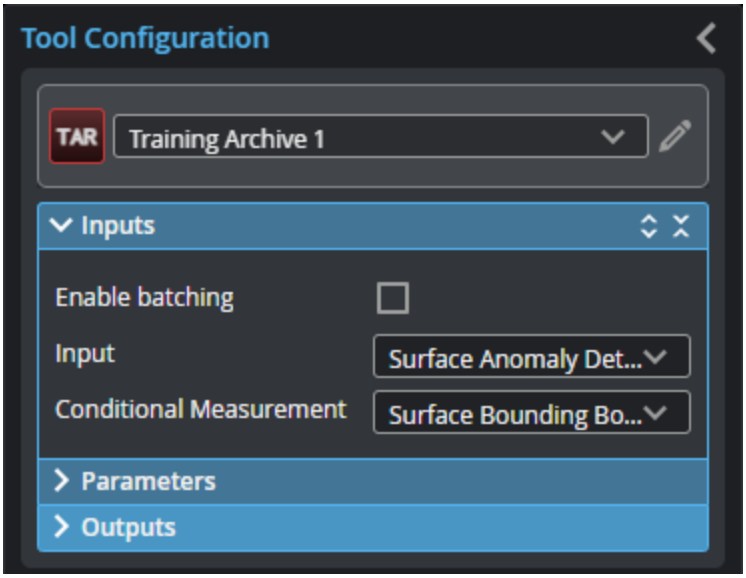
For more information, see *Train Model* on page 307.

Training Archive

The Training Archive tool lets you conditionally add frames to an Anomaly Detector project. (For more information on the Surface Anomaly Detector tool, see *Surface Anomaly Detector* on page 716.)

INPUTS

You configure the tool's inputs in the expandable **Inputs** section.



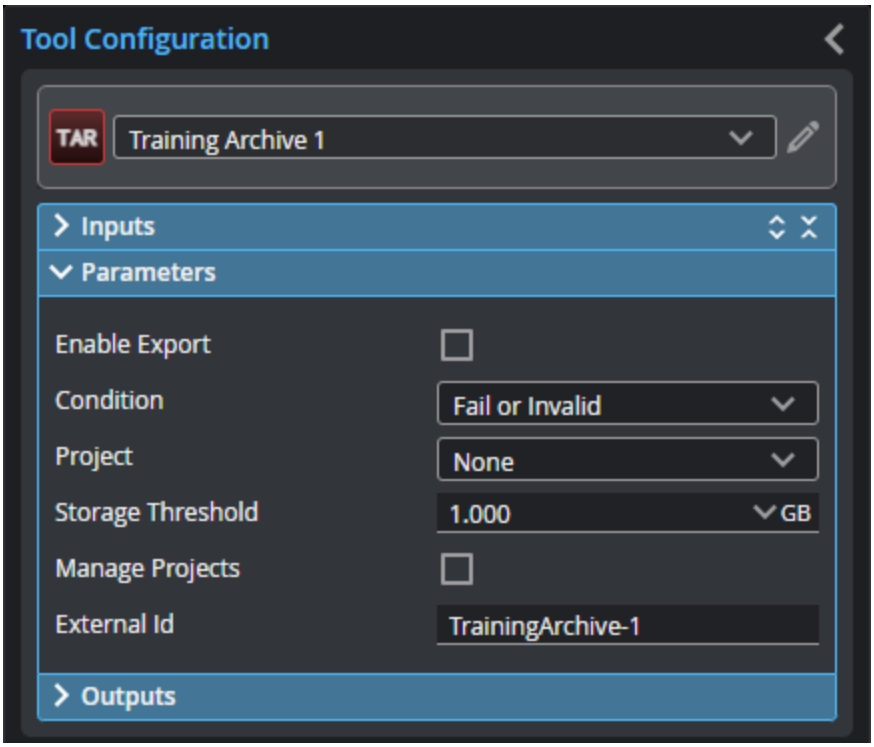
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see Arrays, Batching, and Aggregation.
Conditional Measurement	The measurement the tool uses as a condition to add frames to a project. Some examples of measurements typically used are the "Is Anomaly" output of the Surface Anomaly Detector tool, the Result measurement of the Measurement Formula tool, or an "Output" output from a Script tool.

PARAMETERS

You configure the tool's parameters in the expandable **Parameters** section.



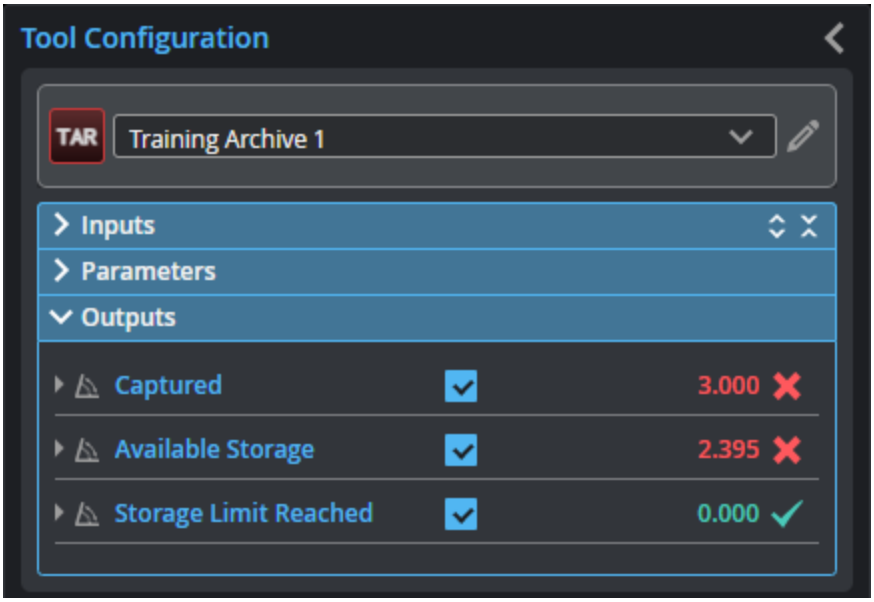
Parameters

Parameter	Description
Enable Export	When this setting is checked, if the conditions are met (**Condition** and **Condition Threshold** parameters) the tool will add the current frame to the selected project.
Condition	Displayed if the **Conditional Measurement** input is set to a measurement. Set this to the type of condition (with respect to the measurement set in **Conditional Measurement**) the tool should use to determine whether to add a frame to the selected project. The frame is added based on the following options: **Fail or Invalid**: If the selected measurement returns a Fail or Invalid result. **Invalid**: If the selected measurement returns an Invalid result. **Pass**: If the selected measurement returns a Pass result. **Less or Equal to**: If the selected measurement less than or equal to the value provided in **Condition Threshold**.

Parameter	Description
	Greater or Equal to: If the selected measurement greater than or equal to the value provided in **Condition Threshold**.
Condition Threshold	The value the tool compares with the measurement set in **Conditional Measurement** to determine whether to add a frame to the selected project. This parameter is only displayed when **Condition** is set to "Less or Equal to" or "Greater or Equal to".
Project	The project the tool will add frames to if the selected conditions are met.
Storage Threshold	The minimum amount of storage allowed on the GoMax unit or the PC. If the storage goes below this value, Training Archive will stop adding frames to the selected project, and the Storage Limit Reached output will be set to 1. Use the Available Storage output to see the current amount of storage on the device.
Manage Projects	When you enable **Manage Projects**, in the **Action** drop-down, you can delete the project currently selected in **Project**, or delete *all* of the projects on the device.
Action	⚠ You are not prompted to confirm whether to delete projects!
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

OUTPUTS

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	
Captured	The number of frames added to the project selected in **Project** so far.
Available Storage	The amount of storage available on the GoMax or PC.
Storage Limit Reached	When the available storage goes below the value in the **Storage Threshold** parameter, this output returns 1.

Image Arithmetic

On color sensors, all Image tools except Image Intensity convert color pixels in images to a monochrome value internally for processing and measurement.

When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value:

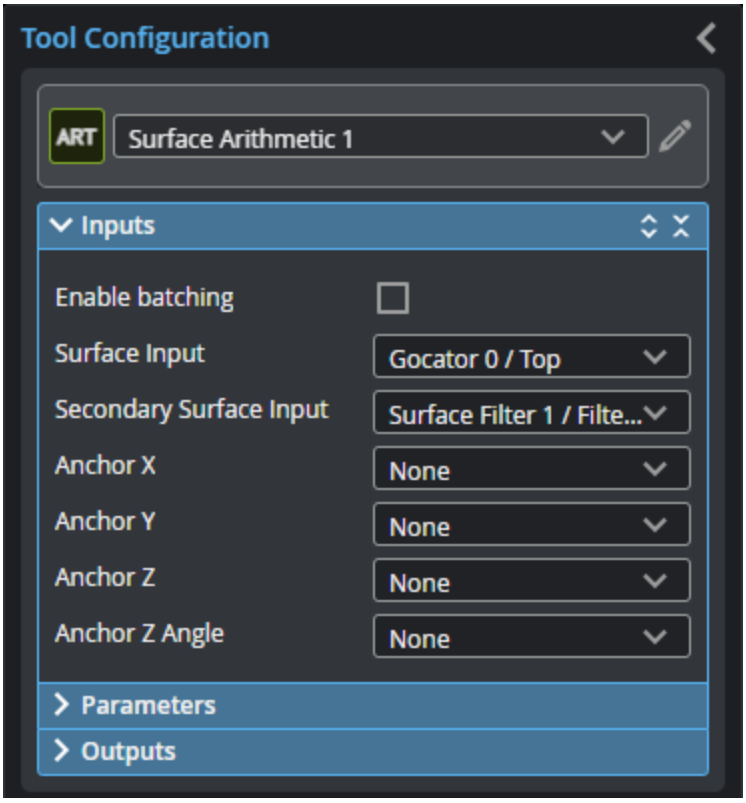
```
pixel = 0.299R + 0.587G + 0.114B
```

When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run.

When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.

The Image Arithmetic tool lets you perform various operations on a pair of images, such as using performing dynamic masking from frame to frame. The tool performs bitwise operations (AND or OR) on the corresponding pixels in the source data.

Inputs



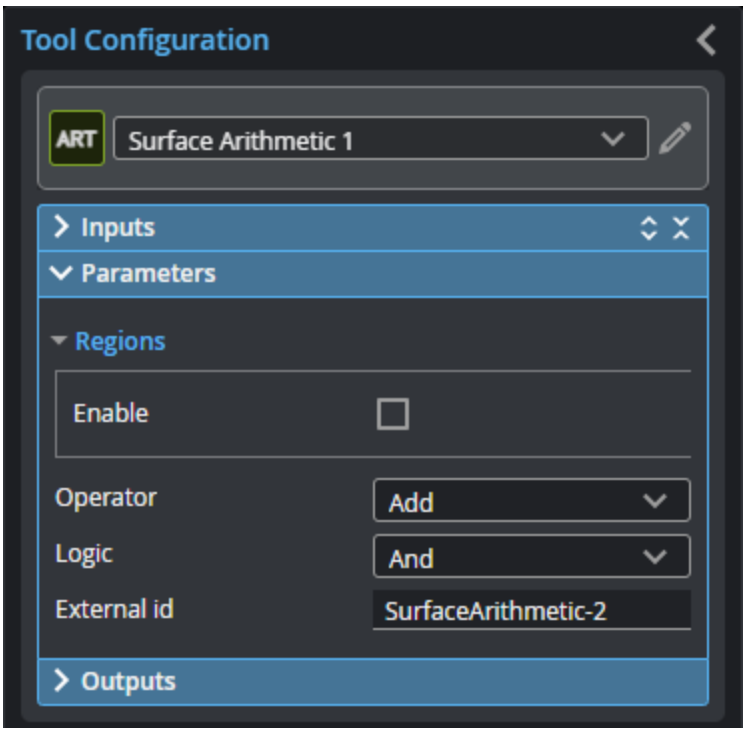
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array,

Name	Description
	other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
First Image Input	The data the tool applies measurements to or processes.
Second Image Input	The secondary input the tool uses to perform arithmetic. Some operations only require a single image.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

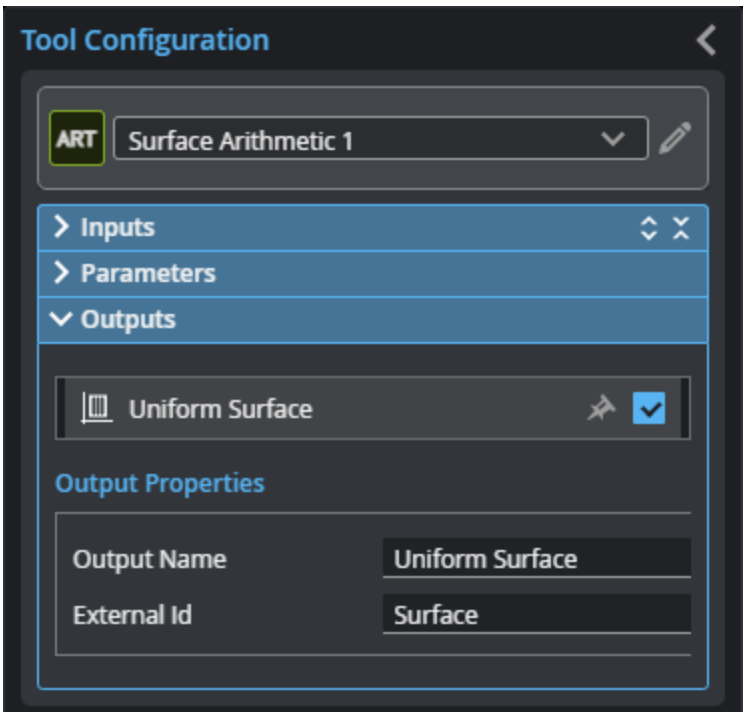


Parameters

Parameter	Description
Operator	One of the following: **Add** - Add the pixel values at the corresponding positions in the images. If the result is an overflow, the maximum value of the image data type is used. **Subtract** - Subtract the pixel values at the corresponding positions in the images. If the result is less than 0, zero is used. **Multiply** - Multiply the pixel values at the corresponding positions in the images. If the result is an overflow, the maximum value of the image data type is used. **Divide** - Divide the pixel values at the corresponding positions in the images.

Parameter	Description
	Average - Calculate the average of the pixel values at the corresponding positions in the images. **Max** - Determine the maximum of the pixel values at the corresponding positions in the images. **Min** - Determine the minimum of the pixel values at the corresponding positions in the images. **Scale and Offset** - Perform a linear transformation for the input image (Ax+B). **>** - Calculate whether the pixel value from the first input image is greater than the one from the second input image for the same position. **>=** - Calculate whether the pixel value from the first input image is not less than the one from the second input image for the same position. **<** - Calculate whether the pixel value from the first input image is less than the one from the second input image for the same position. **<=** - Calculate whether the pixel value from the first input image is not greater than the one from the second input image for the same position. **==** - Calculate whether the pixel value from the first input image is equal to the one from the second input image for the same position. **!=** - Calculate whether the pixel value from the first input image is not equal to the one from the second input image for the same position. **Bitwise And** - Calculate the bitwise AND result of the pixel values at the corresponding position in the two images. **Bitwise Or** - Calculate the bitwise OR result of the pixel values at the corresponding position in the two images **Bitwise Xor** - Calculate the bitwise XOR result of the pixel values at the corresponding position in the two images **Bitwise Not** - Calculate the bitwise NOT result of the every pixel value in the input image. **Logical And** - Calculate the logical AND result of the pixel values at the corresponding position in the two images. **Logical Or** - Calculate the logical OR result of the pixel values at the corresponding position in the two images. **Logical Xor** - Calculate the logical XOR result of the pixel values at the corresponding position in the two images. **Logical Not** - Calculate the logical NOT result of the every pixel value in the input image
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Data

Type	Description
Result	The processed data.

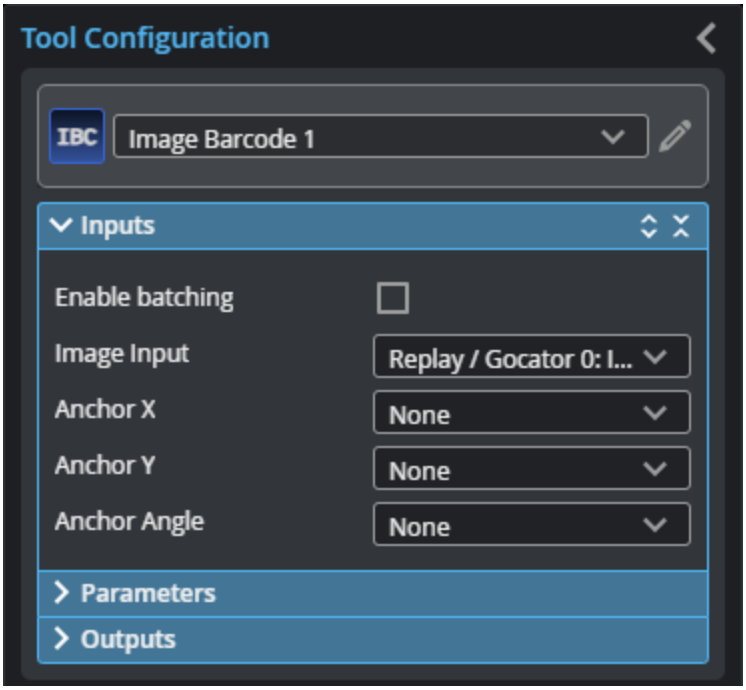
Image Barcode

The Barcode tool lets you decode 1D (linear) barcode data encoded in Image data in grayscale or color.

The tool returns the decoded string, as well as the position (center) and the angle of the code.

The tool also provides measurements that can help you assess the print quality of the code symbols. The tool uses the ISO 15416 grading standard. Only Code 128, Gs1 128, and EAN 13 are supported.

Inputs



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

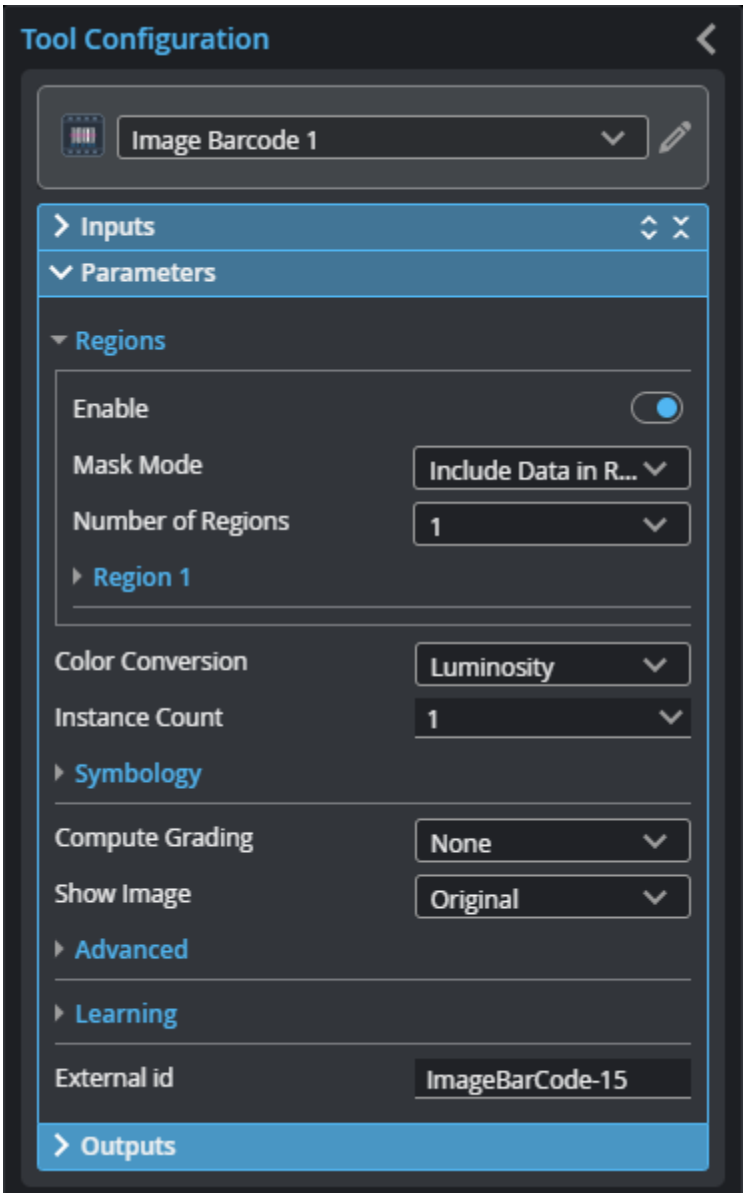
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Name	Description
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.

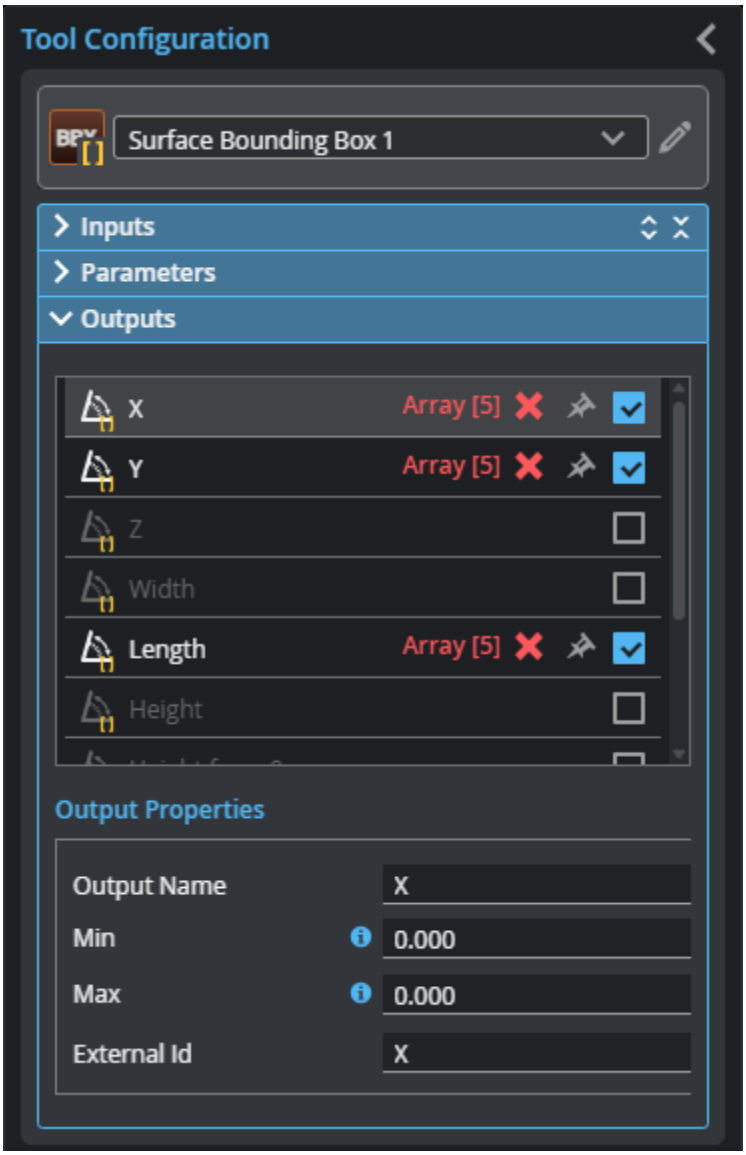
Parameter	Description
Enable	Enables regions and displays the region- and mask-related settings (see below).
Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value: `pixel = 0.299R + 0.587G + 0.114B` When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.
Instance Count	The maximum number of bar codes to decode in a frame.
Symbology	Three categories of bar codes are provided under expandable sections: Default, Optional, and Unsafe. Use the **Operation** drop-down to select or deselect all types in the category. **Default** The following types are enabled by default. Code 128 / EAN 8 / EAN 13 GS1-128 / Code 39 / Code 39 Extended Code 39 Reduced / Code 93 / Code 93 Extended GS1 DataBar Omnidirectional / GS1 DataBar Limited / GS1 DataBar Expanded **Optional** The following types are disabled by default. Codebar / Code 25 Interleaved / MSI UPC A / UPC E / ADS Anker BC 412 / Code 11 / Code 13 Code 2/5 Datalogic / Code 2/5 Matrix / Code 2/5 IATA

Parameter	Description
	Code 2/5 Industry | Code 2/5 Compressed | Code 2/5 Inverted Code 32 | Code BCD Matrix | Code C.I.P. IBM Delta Distance A | Plessey | Telepen RSS14 | RSS14 Limited | RSS14 Expanded **Unsafe** Code STK | Binary Code | Pharma Code One Track
Compute Grading	Whether to assess the quality of the barcode using the ISO 15416 grading standard. Only Code 128, Gs1 128, and EAN 13 are supported. If **Compute Grading** is set to a value other than None, measurement are added in the **Outputs** section.
Show Image	Determines how the image (or the image data in a region) is displayed in the data viewer. One of the following: **Original** Shows the original image data. In color sensors, this will be the color image. **Greyscale** Shows the image data converted to greyscale. The conversion is determined by the option you choose in the tool's **Color Conversion** parameter.
Advanced	Expandable section containing advanced settings. **Display Checksum** Whether to display the checksum of the decoded result. **Human Readable Gs1** Whether to convert the decoded result to human readable result. **Reading Orientation** For most symbologies, a pattern identifies whether the codes must be read from left to right or right to left. For the Code STK, Binary Code and Pharmacode symbologies, select the orientation. **Enable Permissive Decoding** Whether to decode codes containing partially incorrect information. When enabled, the tool assumes characters might be badly printed and uses the

Parameter	Description
	checksum to correct that character. In rare cases, this option may lead to reading false positives codes. Only applies to symbologies for which a checksum is not mandatory. **Timeout** The maximum time the tool is allowed to take. **Validate** Mandatory Checksum: Whether to validate symbologies for which a checksum is mandatory. Optional Checksum: Whether to validate symbologies for which a checksum is optional . Pharmacode: Whether to validate the PharmacodeOneTrack symbology.
Learning	Expandable section containing learning-related settings and operations. Enable or disable settings, place a representative barcode under the camera, and click **Learn image** to determine the best options to use to read bar codes. Click **Reset** to clear. Learning improves robustness and speed on "similar images," meaning repeated production images with comparable optics, distance, and code layout. Avoid it for highly variable scenarios (different cameras, large variation in code size and distance) where one learned configuration will not generalize well. **Keep Default Scales** Whether to keep the scales used by default even if they are useless for reading the barcode. **Add All Scales** Whether to add all of the scales at which a code is detected. Otherwise, the smallest number of scales needed to read the barcode is added.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

When **Compute Grading** is set to "Letter and Number", the tool outputs both "Number Grade" measurements containing a value between 4 and 0 and "Grade" outputs that contain A, B, C, D, or F. If the setting is either "Letter" or "Number", the tool only returns "Grade" outputs, which contain the selected type.

If you set **Instance Count** to a value greater than 1, measurements other than Total Count are returned as arrays.

Measurements

Measurement
Total Count The number of bar codes decoded.
X **Y** The X and Y position of the center of the barcode.
Angle The angle of the barcode.
Decoded String The decoded string.
Decode Grade **Decode Number Grade** Whether the barcode was decoded.
Symbol Contrast Grade **Symbol Contrast Number Grade** The fraction of the total image contrast used by the barcode. For the "Number" grade, between 0 (<= 51 gray values) and 4 (>= 180 gray values).
Minimum Reflectance Grade **Minimum Reflectance Number Grade** The ratio between the smallest and largest gray values in the barcode. For the "Number" grade, the value is 4 if the smallest gray value of the barcode is <= half of its highest value, and 0 otherwise.
Minimum Edge Contrast Grade **Minimum Edge Contrast Number Grade** The smallest contrast between two adjacent bar and space, including quiet zones (blank areas to each side of the barcode that are needed for an accurate reading). For the "Number" grade, the value is 4 if the minimum edge contrast is >= 15%, and 0 otherwise.
Modulation Grade **Modulation Number Grade** The importance of the minimum edge contrast compared to the symbol contrast. For the "Number" grade, the value is between 4 (>= 70%) and 0 (< 40%).
Defects Grade **Defects Number Grade** Degree of irregularities found in elements and quiet zones. For the "Number" grade, the value is between 4 (small irregularities) and 0 (large irregularities).

Measurement
Decodability Grade
Decodability Number Grade
Degree of difference between the measured and expected distances. For the "Number" grade, the value is between 4 (measured distances are close to expected) and 0 (measured distances are far from those expected).
Additional Requirements Grade
Additional Requirements Number Grade
A symbology-specific requirement. Code 128, Gs1 128, and EAN 13 grade the size of the quiet zone of the barcode. For the "Number" grade, the value is 4 (quiet zone is large enough) or 0 (quiet zone is too small).
Global Grade
Global Number Grade
The lowest of all the other grades. If two scan lines yield different decoded strings, Global Grade is set to 0. For the "Number" grade, the value is between 4 (best) and 0 (worst).

Features

Type	Description
Point	The center point of the barcode.



For more information on geometric features, see *Geometric Features* on page 227.

Image Blob

The Image Blob tool lets you detect surface defects, such as uneven or excess material, gouges, or blemishes, on a relatively uniform or flat background. It can also extract targets from the image. The tool optionally lets you set its intensity threshold relative to a user-defined reference region.

The tool first filters image data based on an intensity threshold (above or below it), and then uses configurable morphological operations to better isolate parts of the image. 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).

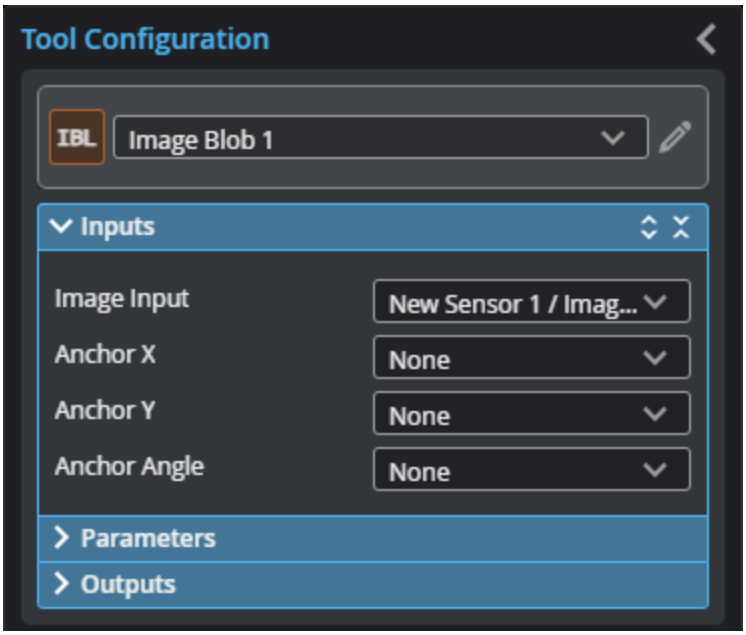
The tool lets you configure the maximum number of blobs to output and returns the total blob count. For each blob, the X and Y center, the width and length, and the area. The center point of each blob is available as a geometric feature.

Note that knowing the rough size and shape of the kinds of defects you expect is important when you are configuring the open and close kernels and the tool's filters.

> When **Enable Array Output** is checked (under the **Parameter** > **Advanced Parameters** section), the tool outputs the blobs in an array called Blob Image. Be *sure* to enable Blob Image in the **Outputs** section after enabling this parameter.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs



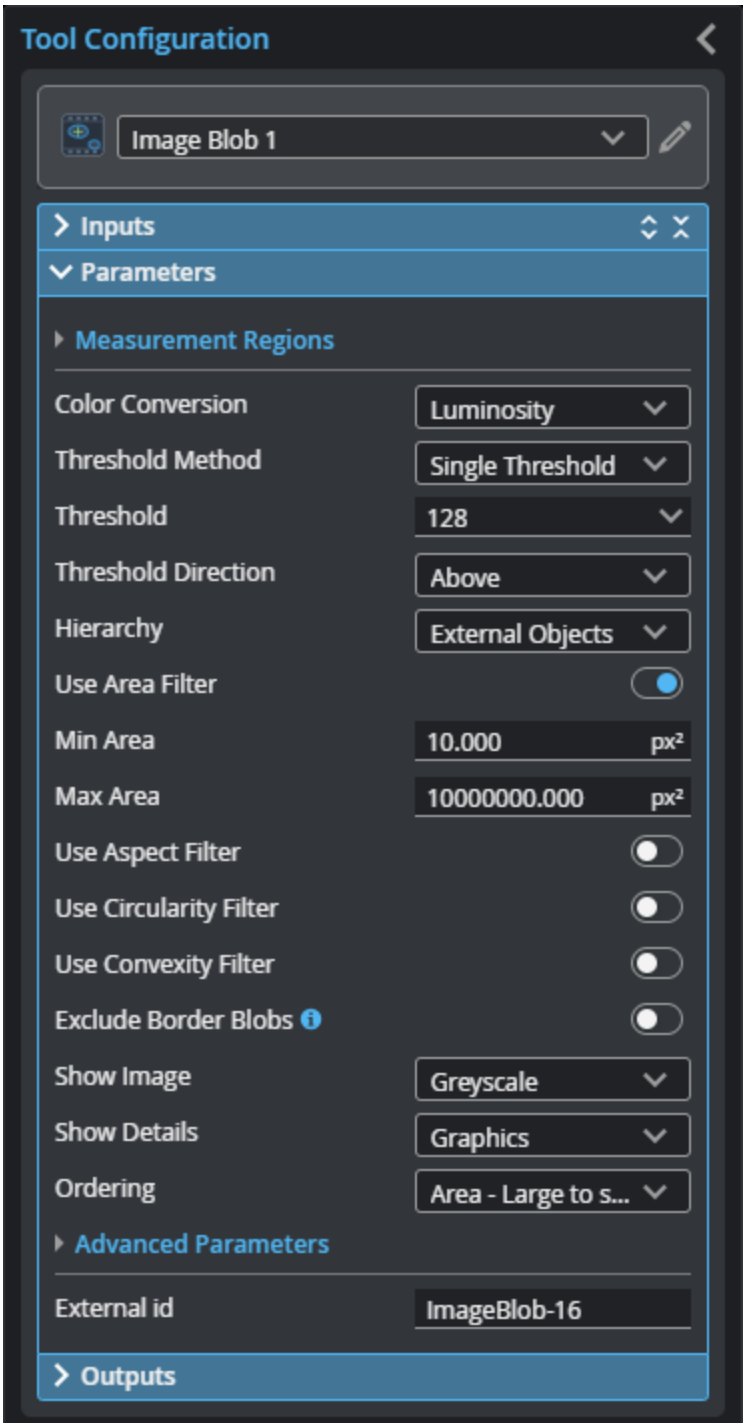
> To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

Parameter	Description
Measurement Regions	When expanded, displays the region- and mask-related settings. Enable and configure regions to limit blob detection to specific areas.

Parameter	Description
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value: `pixel = 0.299R + 0.587G + 0.114B` When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.
Threshold Method	The thresholding method the tool uses to extract blobs. One of the following: **Single Threshold** The tool uses a single threshold, comparing pixel intensities to the value of **Threshold**. Use **Threshold Direction** to choose whether pixels above or below the threshold are included. **Dual Threshold** The tool uses a range of pixel intensity thresholds to extract blobs. Set **Upper Threshold** and **Lower Threshold**. **Otsu** The tool uses the OTSU algorithm to choose the optimal threshold. Use **Threshold Direction** to choose whether pixels above or below the threshold are included. **Local** The tool uses an adaptive threshold in the defined window (**Threshold Window**) using the specified threshold value (**Threshold**) in the input image.
Threshold	The intensity threshold above or below which a pixel is considered for being a

Parameter	Description
	blob. Use the **Threshold Direction** setting to determine whether pixel intensities above or below the threshold are considered.
Threshold Direction	Determines whether pixel intensity above or below the threshold is considered as being part of a blob. **Below**: The **Threshold** value is the maximum intensity that will be considered as part of a blob. **Above**: The **Threshold** value is the minimum intensity that will be considered as part of a blob.
Hierarchy	Provides options to let you find either external objects only or all objects. **External Objects** Use this option to ignore smaller objects in larger objects. Only the outermost objects are returned. **All objects** Use this option to include smaller objects in larger objects. **Holes** Use this option to get holes inside objects. **All Contours** Use this option to get all detected contours.
Use Area Filter	If **Use Area Filter** is enabled, the tool applies an area filter to potential blobs using the values in **Max Area** and **Min Area**.
Use Aspect Filter	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**.
Use Circularity Filter	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	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.
Show Image	Determines how the image (or the image data in a region) is displayed in the data viewer. One of the following: **Original** Shows the original image data. In color sensors, this will be the color image. **Greyscale**

Parameter	Description
	Shows the image data converted to greyscale. The conversion is determined by the option you choose in the tool's **Color Conversion** parameter.
Show Details	Shows graphics and labels, or hides them.
Ordering	Orders the measurements, features, and image data of the individual blobs output by the tool. One of the following: • Position - X increasing • Position - X decreasing • Position - Y increasing • Position - Y decreasing • Area - Large to small • Area - Small to large
Enable Array Output	When **Enable Array Output** is unchecked, the tool outputs each blob as an individual Image output. When **Enable Array Output** is checked, the tool outputs the blobs in an array called Blob Image. Be *sure* to enable Blob Image in the **Outputs** section after enabling this parameter. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Number of Blob Outputs	Determines the number of blobs the tool outputs as measurements, features (center points of blobs), and image data. Currently limited to 200 blobs. If **Enable Array Output** is checked, this parameter is hidden.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

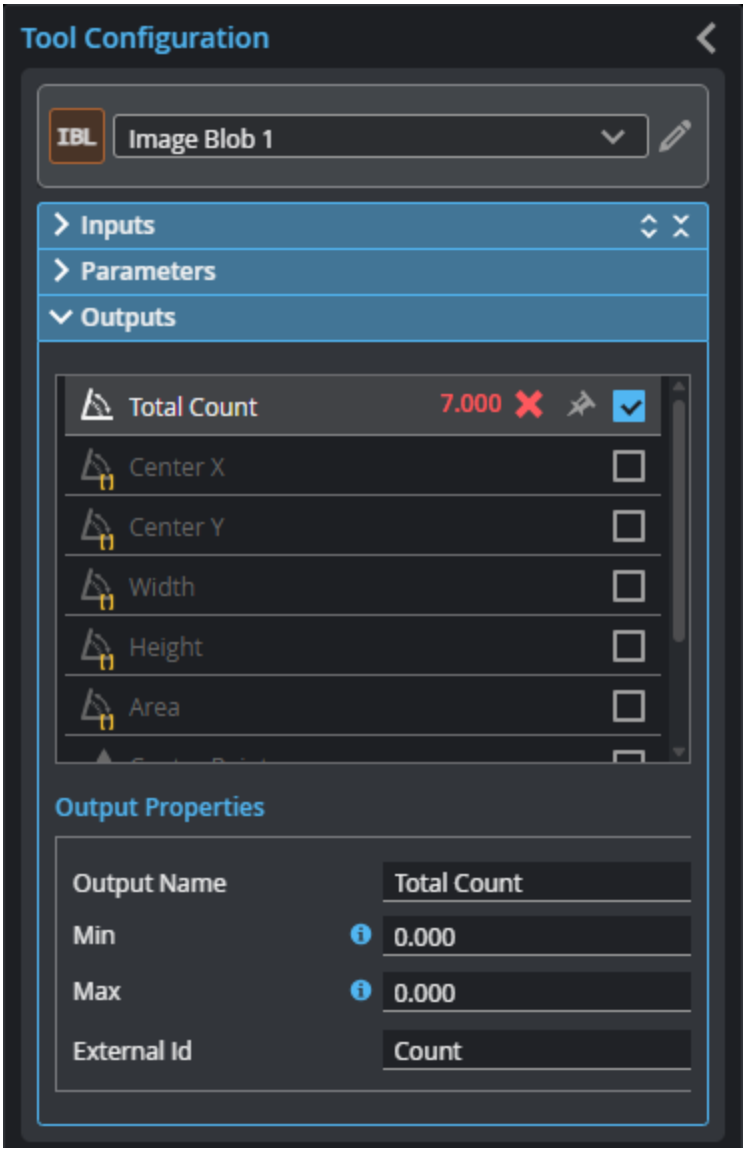
Advanced Parameters

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.
Enable Array Output	When **Enable Array Output** is unchecked, the tool outputs each blob as an individual Image output. When **Enable Array Output** is checked, the tool outputs the blobs in an array called Blob Image. Be *sure* to enable Blob Image in the **Outputs** section after enabling this parameter. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.

Parameter	Description
Number of Blob Outputs	Determines the number of blobs the tool outputs as measurements, features (center points of blobs), and image data. Currently limited to 200 blobs. If **Enable Array Output** is checked, this parameter is hidden.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

If **Enable Array Output** is checked, the tool outputs arrays for the center position, width, height, and area measurements, as well as the Center Point and Blob Image outputs.

If **Enable Array Output** is unchecked, the tool outputs individual, numbered measurements. Use the **Number of Blob Outputs** setting to determine the number of measurements listed in the **Measurements** tab.

Measurements

Measurement
Total Count Returns the total number of blobs identified, based on the tool's parameters.
Center X {n} / Center X **Center Y {n} / Center Y** The X and Y positions of the center of mass of a blob extracted from the image.
Width {n} / Width **Length {n} / Length** The length and width of the rotated bounding box that encapsulates the blob extracted from the image. These are always the major and minor axis of a blob, respectively.
Area {n} / Area The area of a blob.

Features

Type	Description
Center Point {n} / Center Point	The point representing the center of a blob.

For more information on geometric features, see *Geometric Features* on page 227.

Data

Type	Description
Diagnostic Image	Image data you can use to evaluate the impact of the tool's parameters, after filters are applied, to properly separate the areas corresponding to the defects or targets you need to detect.

Type	Description
Image {n} / Blob Image	Image data corresponding to each blob. When the tool is configured to output arrays, the blob images are in Blob Image.

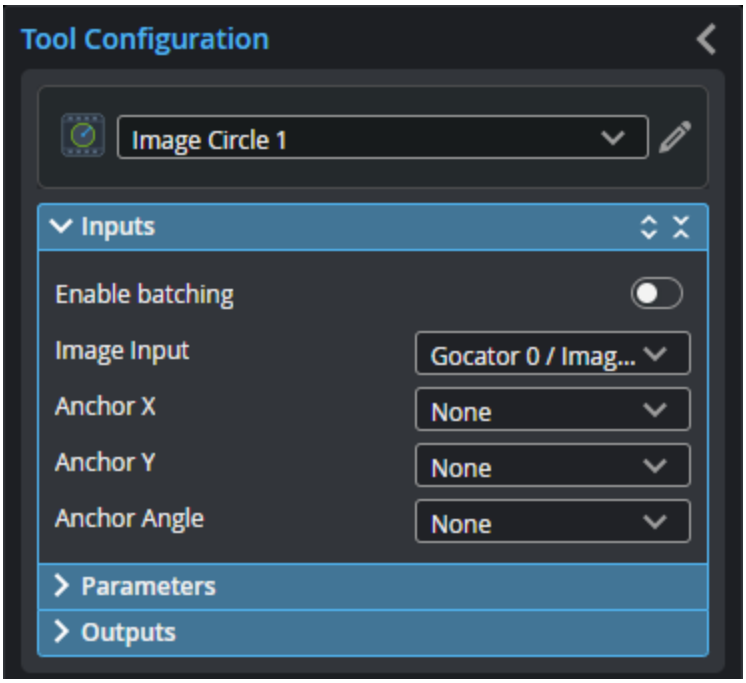
Image Caliper

Coming soon!

Image Circle

The Image Circle tool calculates the best-fit circle for a circular edge in the image and returns measurements of various characteristics of the circle, such as its radius and position. The tool searches along radial paths, whose spacing you define, in the image for changes in intensity values to find the circle.

Inputs

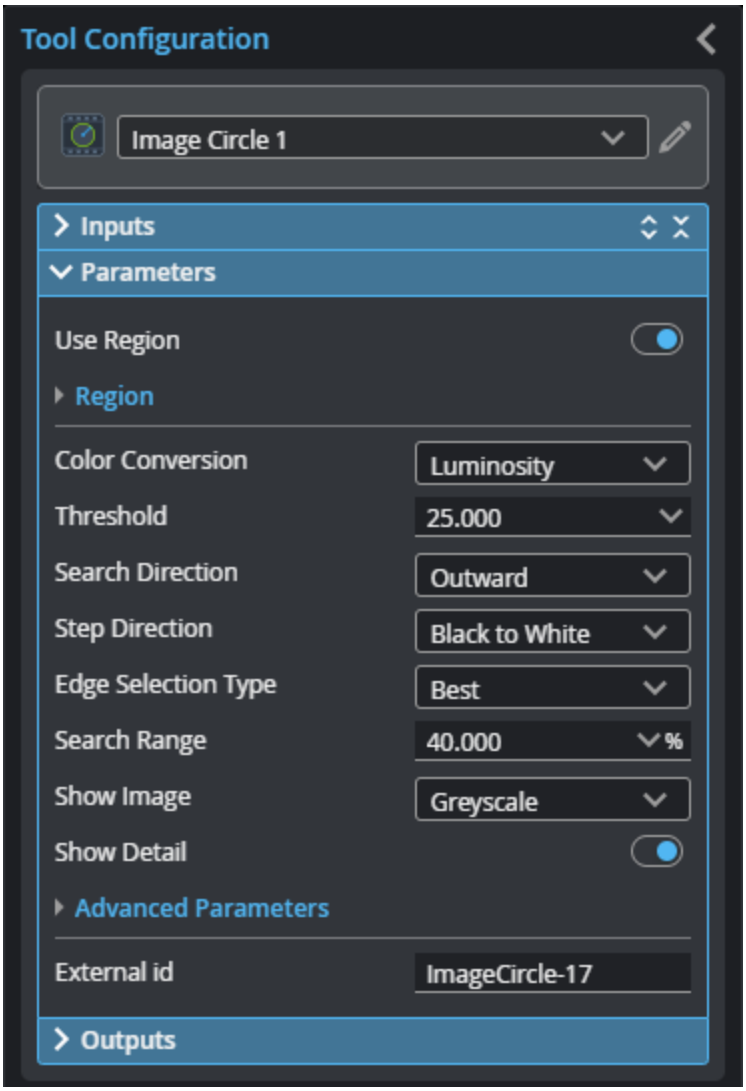


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.



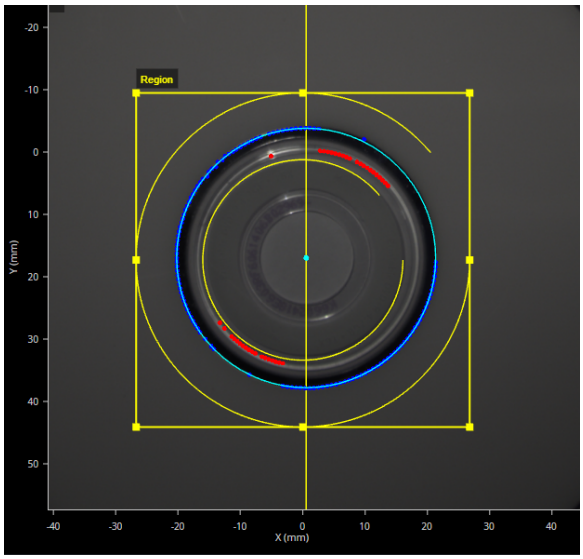
Parameters

Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.

Parameter	Description
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value: `pixel = 0.299R + 0.587G + 0.114B` When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.
Threshold	The threshold intensity value that will be used to determine an edge.
Search Direction	The search direction of the edge search. Either Inward (toward the center of the region of interest) or Outward.
Step Direction	Determines whether intensity transitions should rise or fall. Note that this setting depends on the **Search Direction** setting for its interpretation of intensity transitions. One of the following: **Black to White or White to Black**: Searches for rising or falling intensity values. **Black to White**: Searches for rising intensity values. **White to Black**: Searches for falling intensity values.
Edge Selection Type	Determines which step in intensity the tool uses when there are multiple steps. **Best**: Selects the greatest change in intensity in the search direction. **First**: Selects the first change in intensity in the search direction. **Last**: Selects the last change in intensity in the search direction.
Search Range	TBD
Outlier Fraction	The percentage of outlier pixels to exclude. Setting this to a small value can help the tool fit the circle better to the edge of the circle.
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.

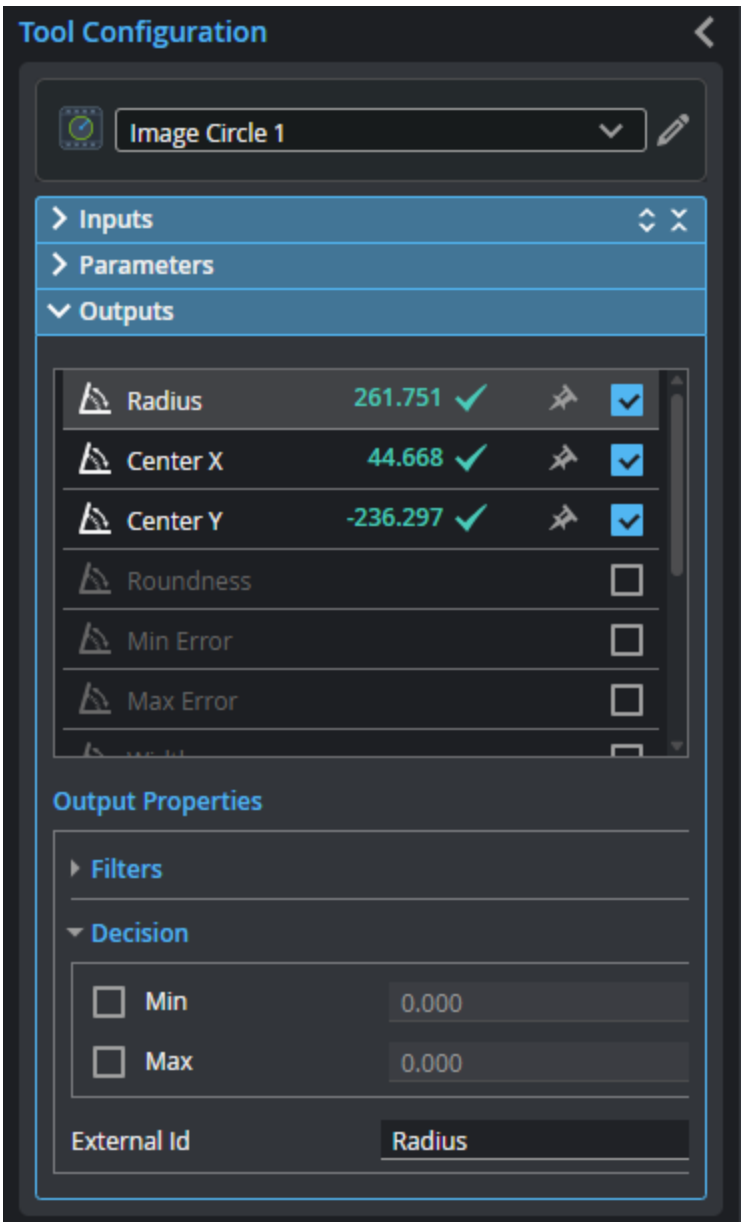
Parameter	Description
Show Image	Determines how the image (or the image data in a region) is displayed in the data viewer. One of the following: **Original** Shows the original image data. In color sensors, this will be the color image. **Greyscale** Shows the image data converted to greyscale. The conversion is determined by the option you choose in the tool's **Color Conversion** parameter.
Show Detail	When disabled, hides the calipers and caliper path, as well as the edge points.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Advanced Parameters

Parameter	Description
Angle Start Angle Span	These settings work together to let you set a partial path and exclude part of the image.  **Angle Start** is the starting angle for a circular path along which the tool searches for the edge of the circle. 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.
Minimum relative change	The minimum change with respect to the value in **Threshold**.

Parameter	Description
Path Spacing	Sets the spacing between paths along which the tool searches for intensity changes. A higher number of paths results in a higher number of edge points, which makes the fitting of the circle more accurate. However, a higher number of edge points results in a greater tool execution time.
Step Smoothing	The size of the window along the radial path used to calculate an average for each pixel along the path. 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).
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 **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*.
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.
Which Circle	Tells the tool which circle (Inner or Outer) to use when Minimum Zone Circle is the fit method in **Fit Type**.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Measurements

Measurement
Radius Measures the radius of the circle.
Center X

Measurement
Center Y The circle center X and Y positions.
Roundness Returns the roundness of the fitted circle.
Width **Height** The width and height of the circle's bounding box. (The values will always be the same; the outputs are included for consistency with other Image tools.)
Min Error **Max Error** The minimum and maximum error among the pixels with respect to the fitted circle.

Features

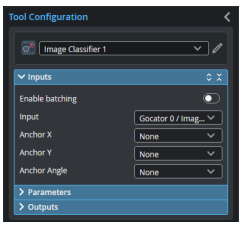
Type	Description
Center Point	The center of the fitted circle.
Circle	The fitted circle.
Min Error Point	The point of minimum error which is furthest inside of the circle.
Max Error Point	The point of maximum error which is furthest outside of the circle.

Image Classifier

The Image Classifier tool lets you train a model using classes that you define. For example, for a food processing application, you could define classes such as drumstick, leg, and so on. When scanning targets and using the tool's inference engine, it can determine what type of cut it's seeing.

The tool supports uploading YOLO26 models.

Inputs



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	Batching (when a tool takes an array as input and processes each element in the array individually) is currently not fully supported by this tool. If you choose to use batching, keep the following limitations in mind: • The Label page only shows the first element of the input array. • For anomaly detection, blobs representing the anomalies are not displayed. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Development Workflow

The high level workflow is as follows:

1. Collect representative scans of the targets you need to inspect in order to train a model. In addition to "good" parts, you need to scan "bad" parts, that is, ones with typical defects.
2. Save the scans to a .gprec file.
3. Create a pre-processing toolchain and train a project's model.

4. Deploy the model and connect post-processing tools.
5. Optionally, collect new production data for model retraining.

Collect dataset

Unless you are using a previously created dataset, you must create the dataset that the tool will train a model on before it can process new production data. To do this, perform the following steps:

1. Acquire 20 or more images that represent each of the classes you intend to add to the project.

   You can acquire the data in a single .gprec file or to multiple files. If you record scans to a single file, grouping each type together will make it easier to label them later in the wizard. Make sure there are no empty frames.

   If you have previously acquired images, you can also upload them from the Label Project page; for more information, see *Create New or Open Project* below.

2. (Optional) Save a separate .gpsup file with the device configuration settings such as exposure time and related imaging parameters.

   Ensure the same parameters are used on the production system. Otherwise, you will need to collect new training data.

Implement a Preprocessing Toolchain

You typically need to create a preprocessing tool chain for your application. The preprocessing tool chain limits variation in the scan data to the region or surface of interest. With less variation in the scan data, the total number of frames needed for training is reduced.

The preprocessing tool chain must not change between training and inference. Otherwise, the system will not function properly. You must save your preprocessing toolchain in the .gpsup file created previously or using a separate .gpjob file.



You *must* use the same preprocessing toolchain for training and inference to ensure proper detection of anomalies.

Train Model

ADD THE TOOL

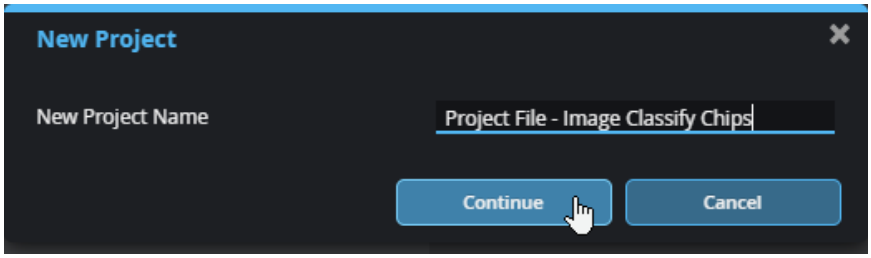
1. Add and connect the tool to the output of the preprocessing tool chain.
2. In tool's **Parameters** section, click the **Train Model** button.

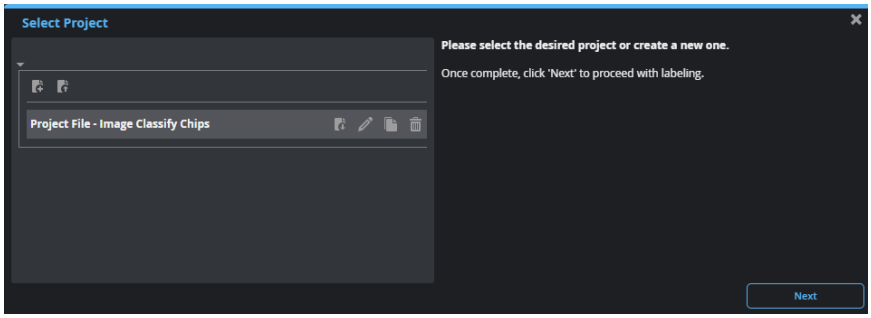
   The routine starts and the **Select Project** page opens. You will create a new project in this page, or open a previously created project.

CREATE NEW OR OPEN PROJECT

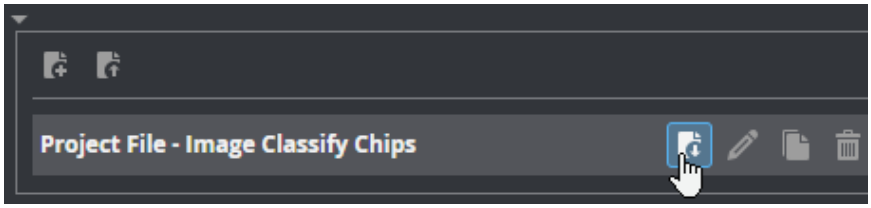
A project is a collection of frames used for training, classes and labels you apply to those frames, and the models trained on those frames. You should choose a meaningful project name, such as the part name, especially when working with multiple projects.

1. On the **Select Project** page, click the New Project icon, provide a project name, click **Continue**, and then click **Next**.

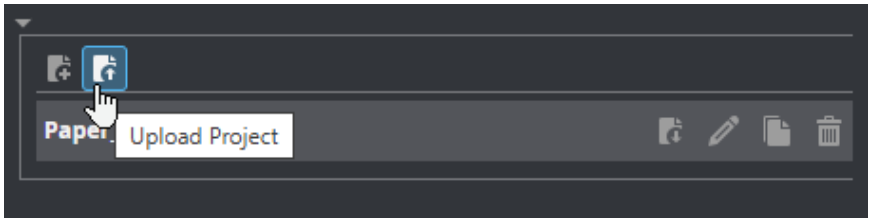




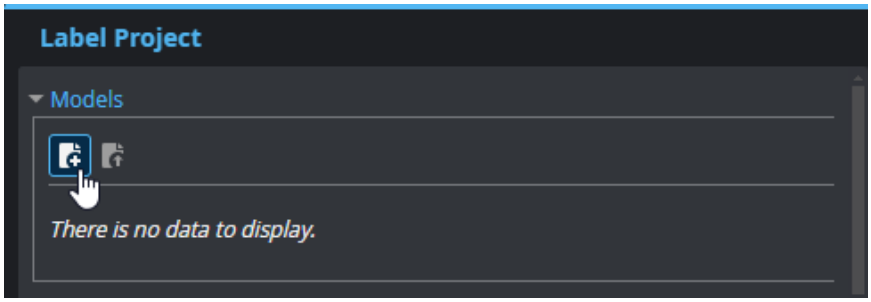
Use the download icon on a project's row to download this to your PC and upload to another device.

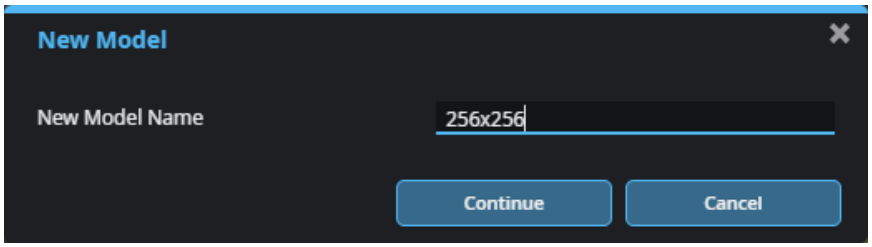


Use the upload icon to import a previously downloaded project.

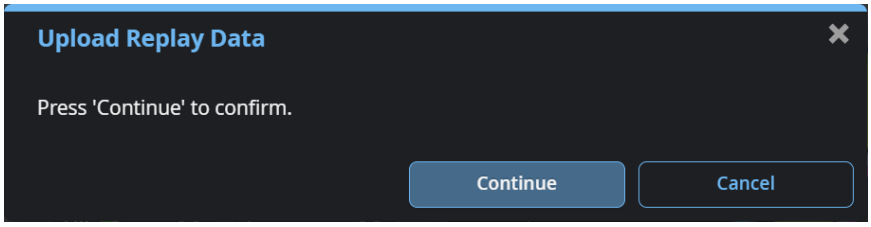


2. In the **Label Project** page, in the **Models** section, create a new model and click **Continue**.





3. In the **Frames** section, do one of the following
   - To upload currently loaded replay data, click the Upload Replay Data button ( ) , and then click **Continue**.



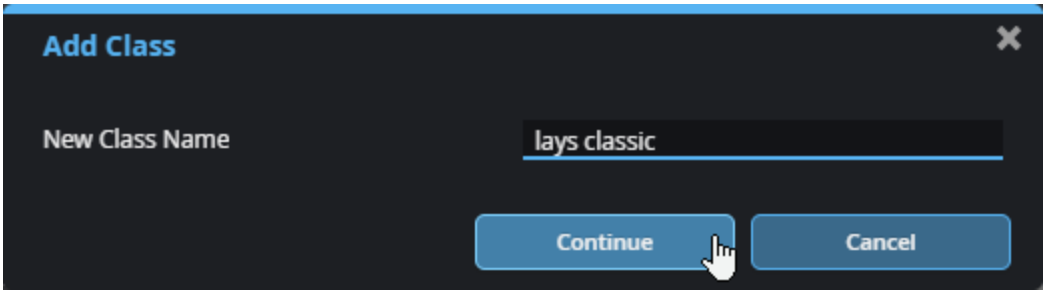
   - To upload the data from a .gprec file on the local PC, click the Upload From Disk button ( ), choose a file in the Open dialog a, click Open.

   The tool loads the frames into the model.

⚠	Loading replay data into the project can take several minutes. If a "Delayed Response" message displays, click "Wait." In this case, add a Decimation Filter to the pipeline.

ADD CLASSES

You must now add classes to the project. For each class, click the Add Class button ( ), provide the name of the class, and click **Continue**.



LABEL FRAMES

You must now assign a class label to each frame.

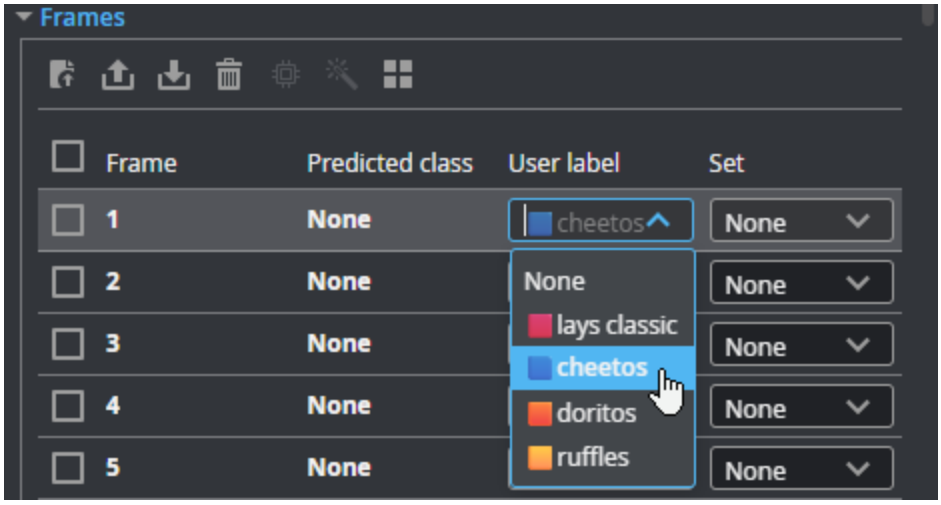
By default, the tool chooses the set each frame belongs to (Train or Test, under the **Set** column), based on the following proportions:

- 70% of a class in Train
- 30% of a class in Test

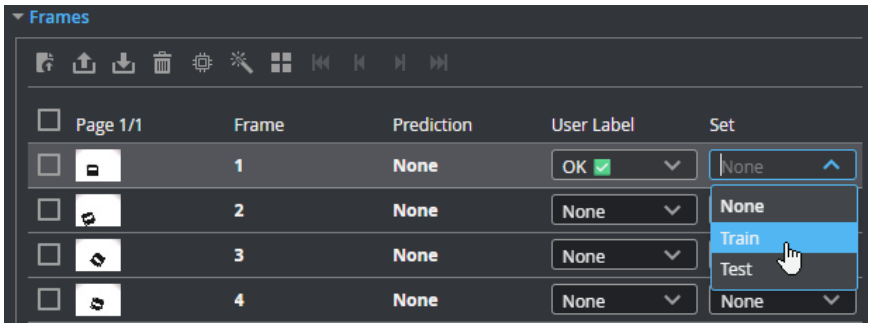
If you choose to disable Auto-split and assign frames to sets manually, make sure to use the recommended proportions.

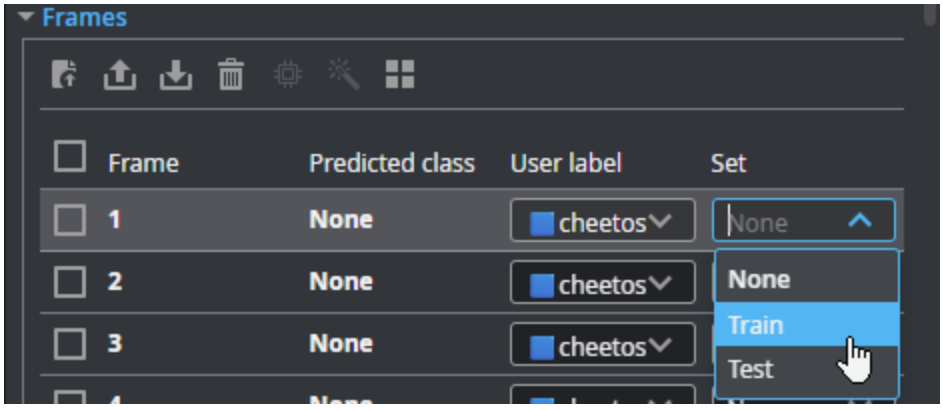
1. Select each row under **Frames** and choose correct label for the frame under **User Label**.

1. Select each row under **Frames** and choose the correct label for the frame (OK or NG) under **User Label**.



If you have disabled auto-split, also choose the frame's set under **Set**. Make sure you use the proportions described in step .





To quickly choose the label and set for multiple frames, do one of the following to first select the frames:

- To select a range of frames, click the checkbox of the first frame, press Shift, and click the checkbox of the last frame of the range.
- To select multiple frames, click the checkbox of the first frame, press Ctrl, and click the checkbox for each frame you want to select.
- To select non-contiguous ranges, alternate between clicking Ctrl and Shift.

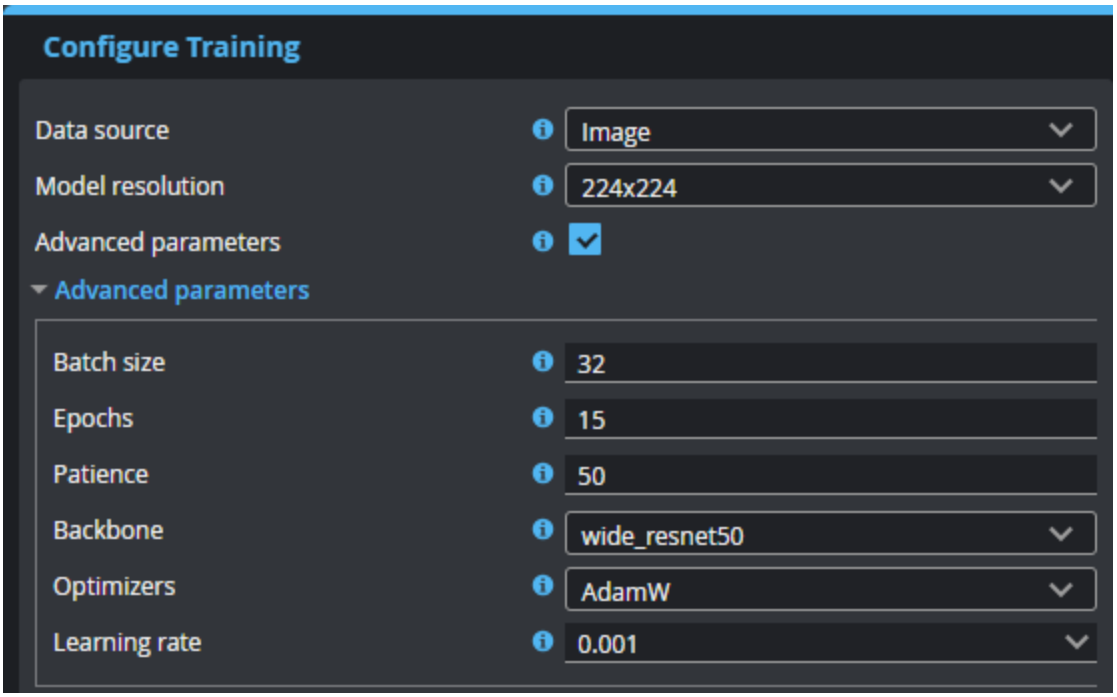
After selecting the frames, choose the label as described above. To deselect all frames, click anywhere between the checkbox and the User Label and Set drop-down menus.

2. Click **Train Model** to configure the training parameters.

   For information on configuring training, see *Configure Training* below.

CONFIGURE TRAINING

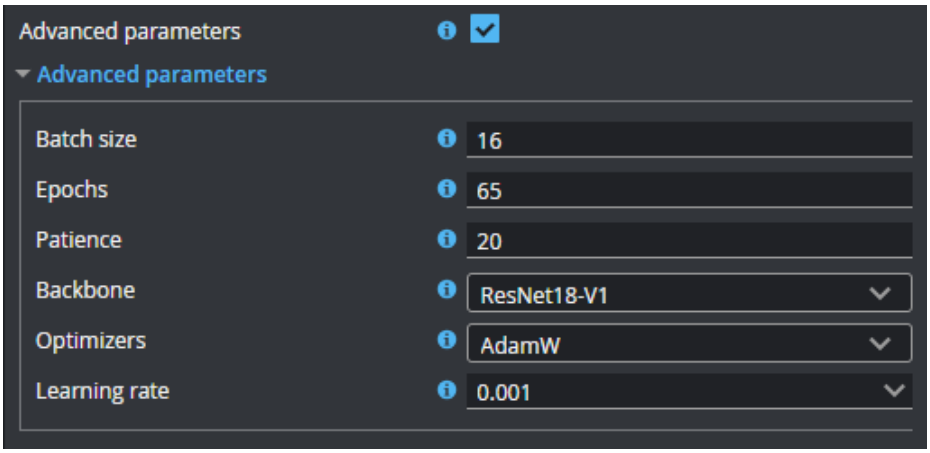
After labeling the project, you must configure training. For descriptions of the settings, see below.



Training Parameters

Parameter	Description
Data source	The source the tool uses during training.
Model resolution	Decimates the input data to the specified resolution for processing. A higher resolution improves anomaly location accuracy but increases processing time for training.For most applications, leave this at 224x224.
Advanced parameters	If the training results are generally good but the tool is reporting a small number of false positives or false negatives, you can modify these parameters to further improve results.

Parameter	Description



Batch size

The number of frames in a batch (a subset of the entire dataset passed to the training algorithm for an iteration).

Epochs

The number of times the entire dataset passes through the training algorithm. To reduce incorrect classifications, increase this value. Note that training will take longer.

Patience

The number of epochs to wait without improvement in validation metrics before stopping the training early to prevent overfitting. As a guideline, start with a value around 10-20% of total epochs.

Backbone

The feature extractor backbone used by the tool.

Select pre-trained model weights for training. One of the following:

ResNet18-V1: A lighter model that is fast and a good starting point.

ResNet18-V2: A medium model with more capacity than V1.

ResNet50: The best balance for most tasks.

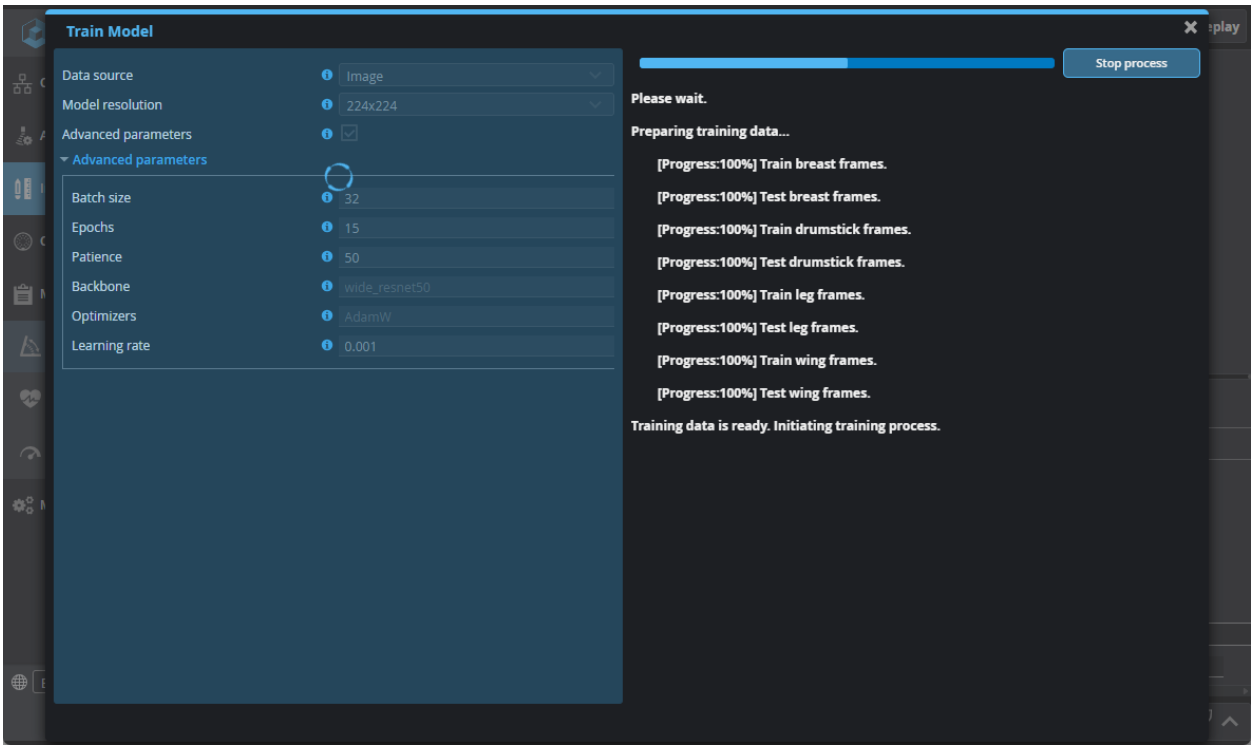
Wide ResNet50-2: Slow and larger memory, with better for complex textures.

Optimizers

Algorithms that adjust the model's weights and biases to minimize the loss (error) and improve its accuracy. One of the following:

Parameter	Description
	Adam: Best for most cases. AdamW: Slightly slower than Adam but better at preventing overfitting. Use this when the model performs well during training but fails on new images. SGD: Maximum accuracy, requires large datasets; use this when you need that extra 1%-2% accuracy and have time to tune.
	Learning rate Set this to a higher value to reduce training time. Note that a value that is too high can cause the model to miss the point where the difference between predicted and actual class is at its smallest (that is, overshooting it). A lower value causes the model to learn more slowly, but potentially more accurately; if the value is too low, the model can get stuck.

After you configure training, click **Start Training** to train the model. The tool will start by preparing the training data, providing progress feedback:



After the data is prepared, the tool will start training and generating graphs of metrics to chart the progress.

Loss: This metric represents the cumulative error for each frame in the Test set, determined after each training run iteration or epoch. To put it another way, it's the difference between the model's

current prediction and the actual label you provided (sometimes called the "ground truth"). A decreasing number (the graph's line is moving down) means the model is improving.

Accuracy: This metric displays the portion of total correct predictions (value between 0 and 1).

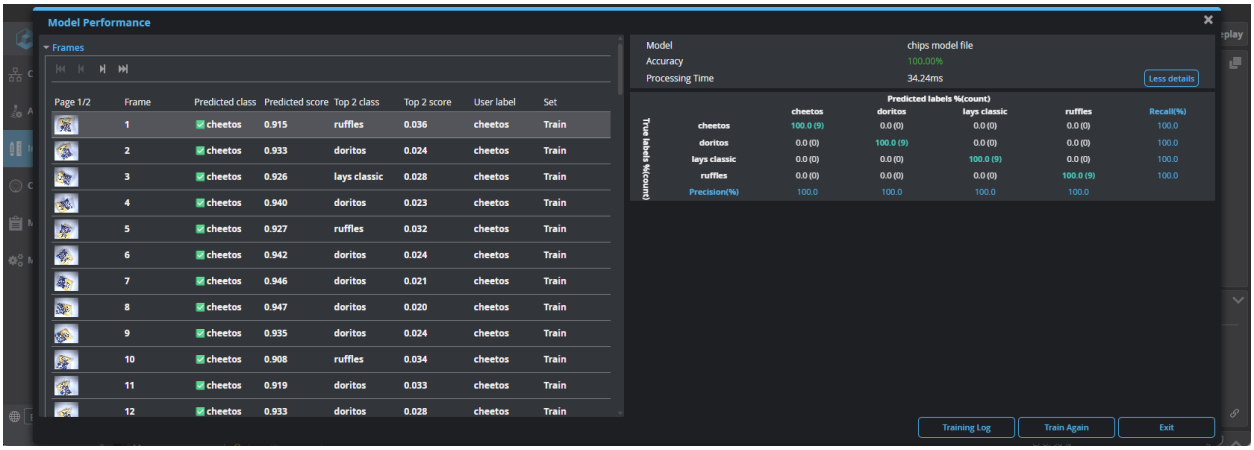
After the training has completed, click **Next** to test the model. Note that training can take several minutes.

MODEL PERFORMANCE TESTING

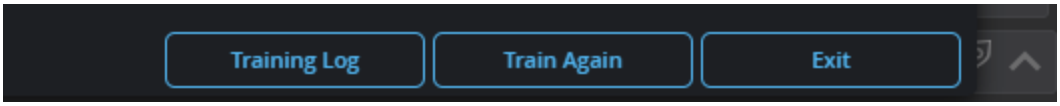
After the tool has finished training the model, it will use the model to make predictions on the frames to determine a score for each frame. The Model Performance page displays the following results:

Performance Results

Name	Description
Model	The trained model the tool used during the model performance phase.
Processing Time	The average time required for inference. Includes the pre-processing tool chain.
Accuracy	A percentage representing the overall accuracy of the model's class predictions.



The Model Performance page also lets you download an HTML file containing reports and other diagnostics. LMI recommends checking the visual results in this file.



You can also retrain the model with different training configurations if you wish.

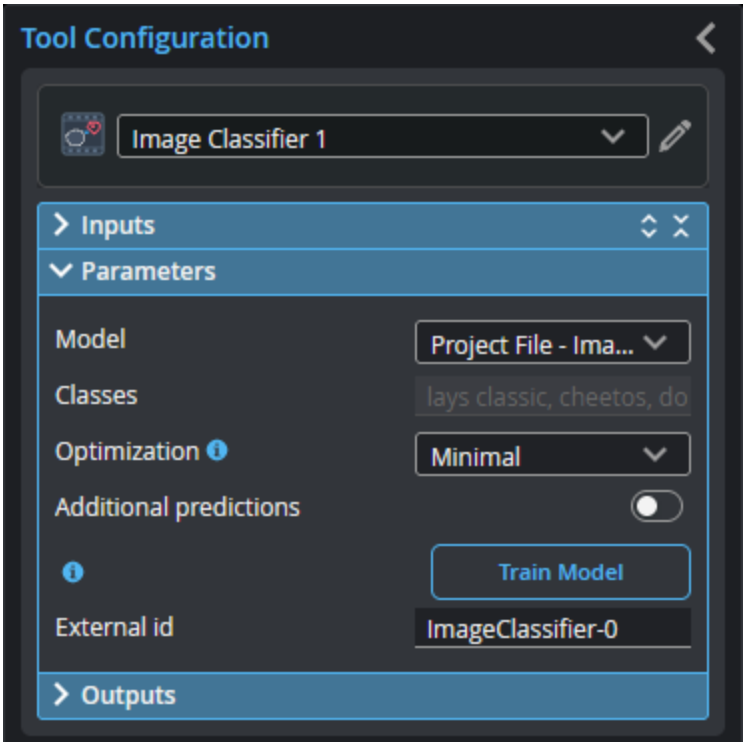
You are now ready to use the trained model on new scan data in production.

Parameters

In the **Parameters** section of the tool, you can launch the Train Model wizard, in which you can create a project, as well as the models, classes, and frames that the project includes. After you have

created the project, the Train Model wizard lets you train the model on the frames. Once training is complete, you can use the project as an inference engine to classify new frames of date, that is, on the production line. For information on creating a project and training its model, see XYZ.

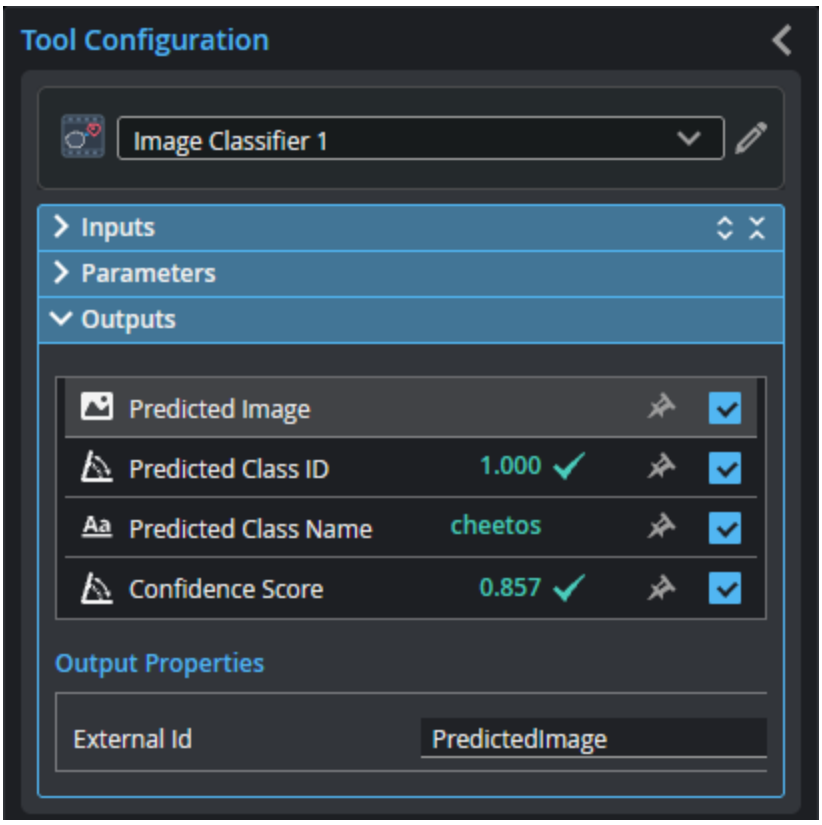
The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

Parameter	Description
Model	The trained model the tool uses to classify images during production.
Classes	A read-only list of the classes defined in the selected model.
Optimizations	The level of optimization applied to the model. One of the following: Minimal, Balanced, High. A High setting requires re-optimization of the model.
Additional predictions	When this parameter is enabled, the tool also outputs arrays containing lists of the project's classes and IDs, as well as the scores associated with the classes / IDs, for the current frame. The array items are ordered from highest to lowest score. For more information, see *Measurements* on the next page.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Note that some measurements return IDs, whereas others return class names. The IDs correspond to the class names added in the project. So for example, given the classes "lays classic", "cheetos", "doritos", and "ruffles", the corresponding IDs are 0, 1, 2, and 3. The ID values are ordered based on the order in which the classes were added in the project.

Measurements

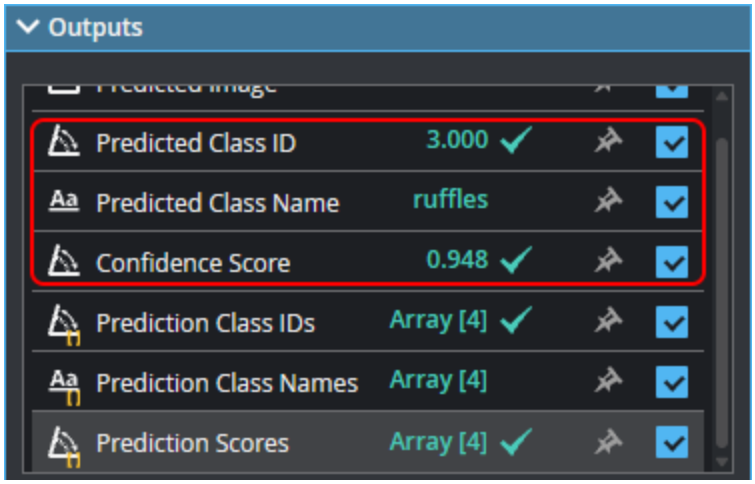
Measurement
Predicted Class Id The ID of the highest scoring predicted class of the image.
Confidence Score The score associated with the predicted class.
Prediction Class IDs
Prediction Class Names
Prediction Scores

These measurements and outputs are only displayed if **Additional predictions** is enabled.

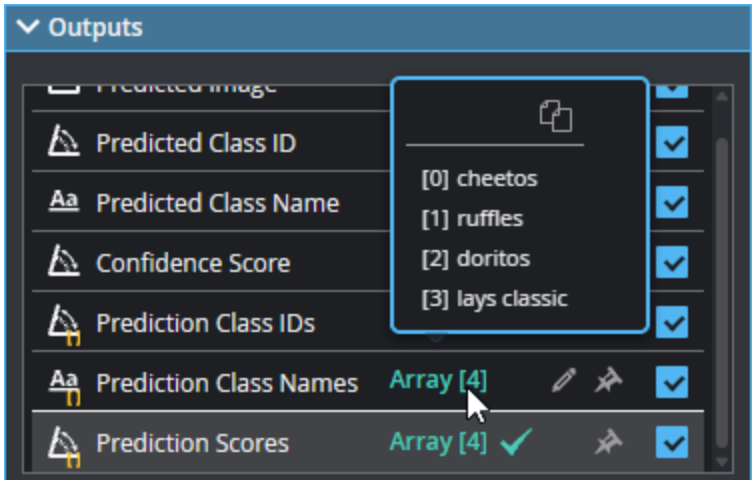
Measurement

The Prediction Class IDs and Prediction Class Names outputs return arrays containing the project's class IDs and class names, ordered based on their score in decreasing order. The Prediction Scores array returns an array containing the scores of the image, ordered from highest to lowest.

For example, let's say the tool has classified an image of a bag of snack food as "ruffles" (ID = 3), rather than any of the other classes ("lays classic", "cheetos", or "doritos"), with a score of 0.894.

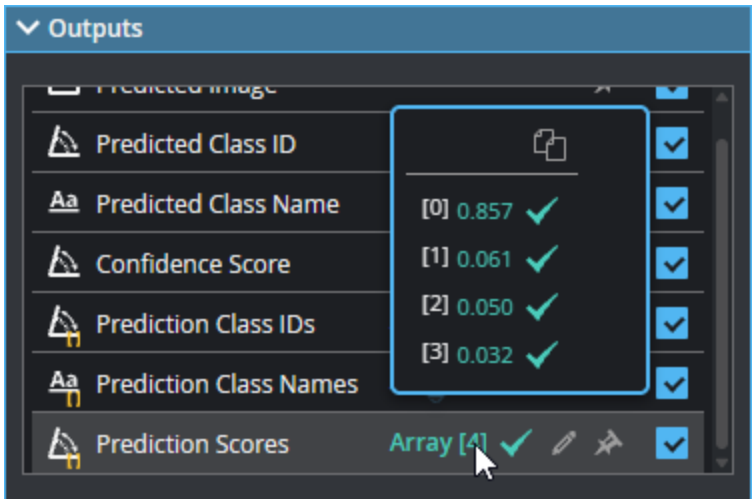


The Prediction Class Names output contains the names of classes in the project ordered from the highest to lowest score the tool assigned to the current frame. In the following, "wing", the predicted class name for the frame's image, is the first item in the array.



The scores corresponding to those ordered classes are available in the Prediction Scores output. The following shows the corresponding scores. The score for "ruffles" (0.926) is the first item. The next value, 0.032, indicates prediction score for "doritos", and so on.

Measurement



Use these alternate candidate predictions to determine if the model needs specific additional training to better classify images.

Data

Type	Description
Predicted Image	The frame's image with an overlay displaying the class names and their corresponding prediction scores.

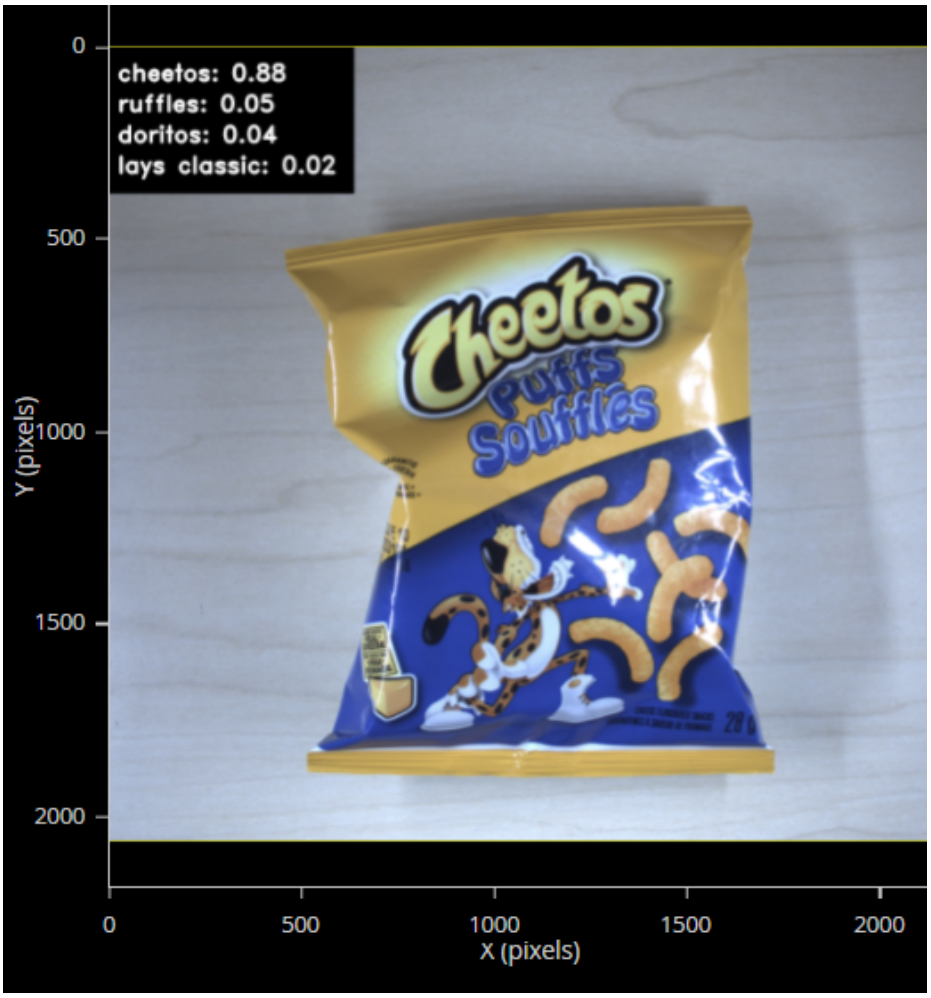
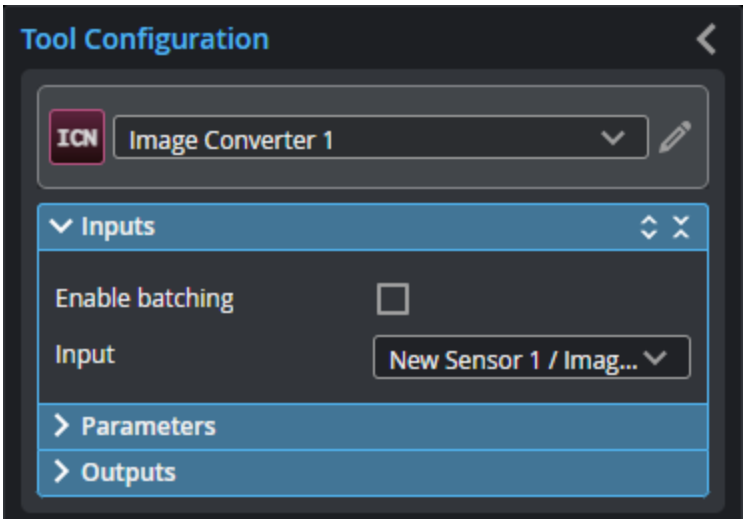
Type	Description
	
Predicted Class Name Prediction Class Names	The Predicted Class Name output contains the text of the frame's predicted class name. You can output this to a Script tool or to a PLC to control other devices based on the class name. The Prediction Class Names output contains an array of the all of the project's class names, ordered by their score for the current frame.

Image Converter

The Image Converter tool lets you convert 8-bit RGB images to grayscale images, and from grayscale to RGB. Note that 16-bit images are not supported.

Inputs

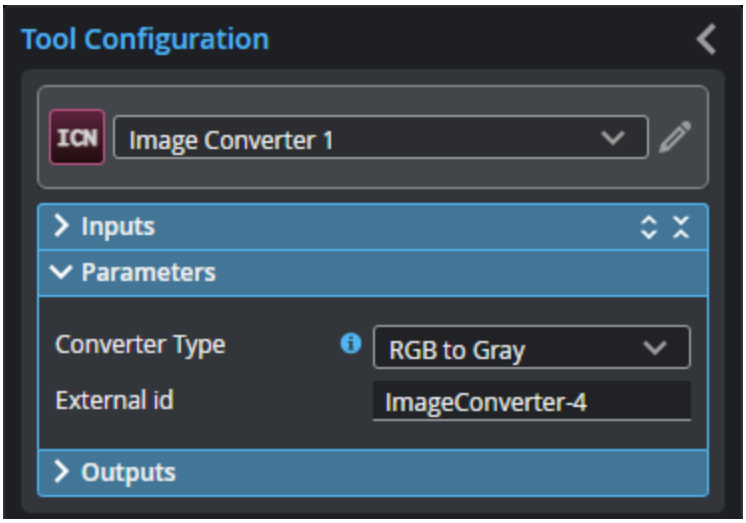


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.

Parameters

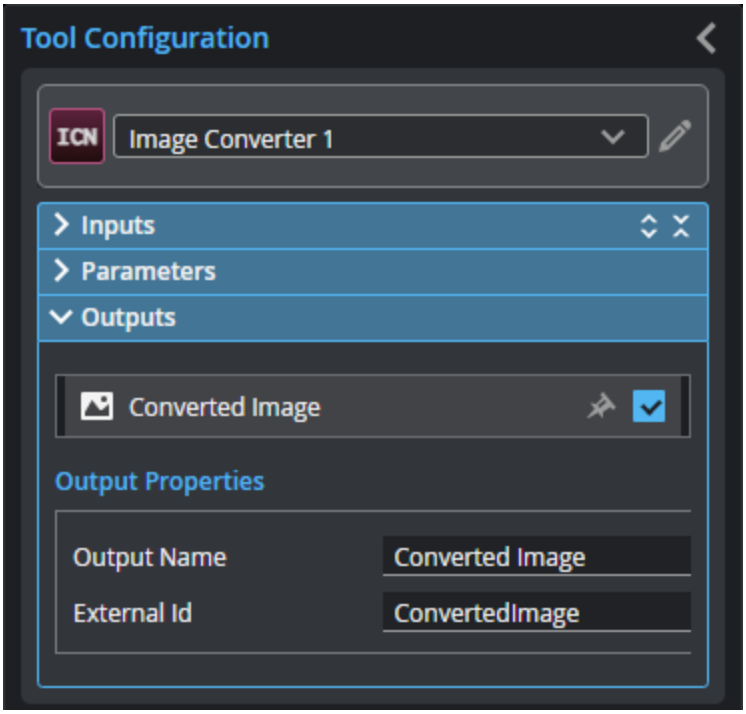
The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

Parameter	Description
Converter Type	The type of conversion. One of the following: RGB to Gray, Gray to RGB. The tool only supports 8-bit images.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Data

Type	Description
Converted Image	The converted image.

Image Feature Mask

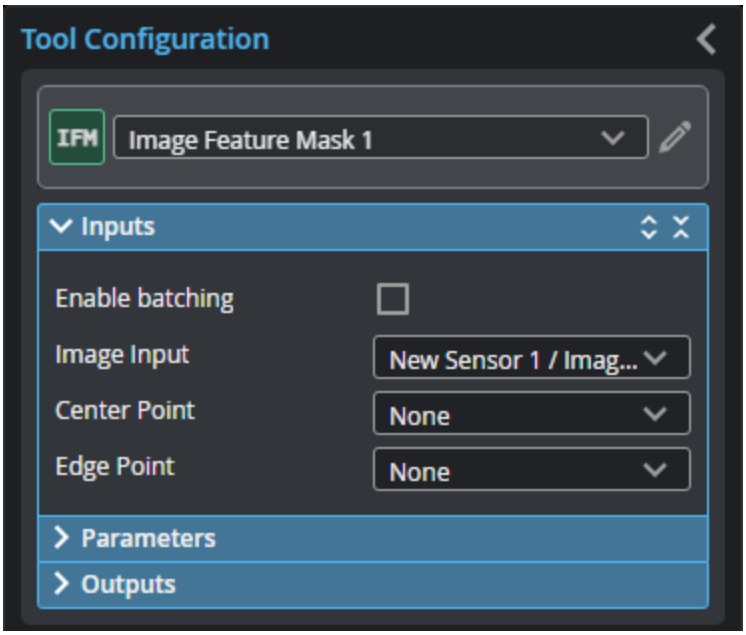
The Image Feature Mask tool dynamically defines a polygon or circle mask using Point geometric features produced by other tools to crop an image and output it.

When only two Point geometric features are passed as input to the tool, it creates a circular mask, where one point is the circle's center point, and the other is a point on the circle.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



Inputs

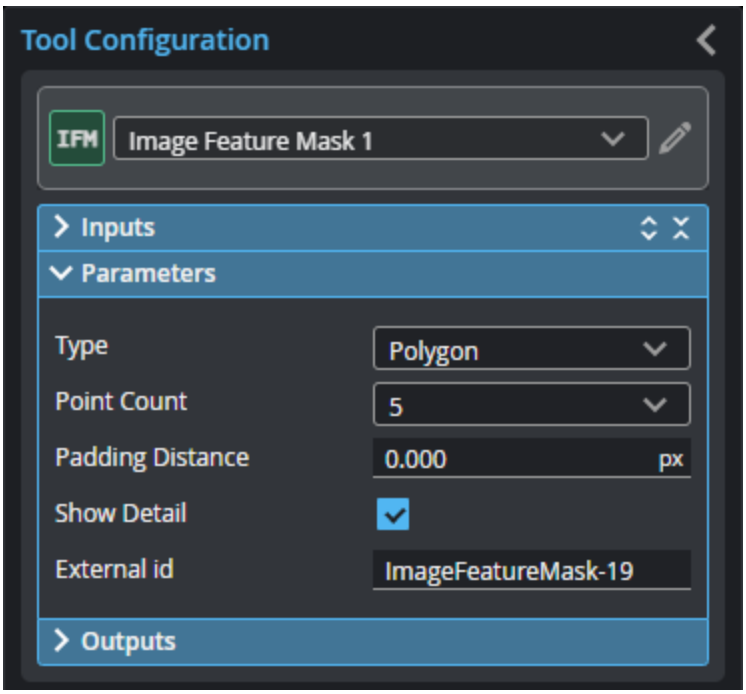
Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Point {n}	The nth Point geometric feature. Only displayed if you set **Point Count** to 3 or higher.
Center Point	The center Point geometric feature of the circle. Only displayed if you set **Type** to Circle and **Point Count** to 2.
Edge Point	The Point geometric feature on the circle.

Name	Description
	Only displayed if you set **Type** to Circle and **Point Count** to 2.
Image Input	The data the tool applies measurements to or processes.

For more information on geometric features, see *Geometric Features* on page 227.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



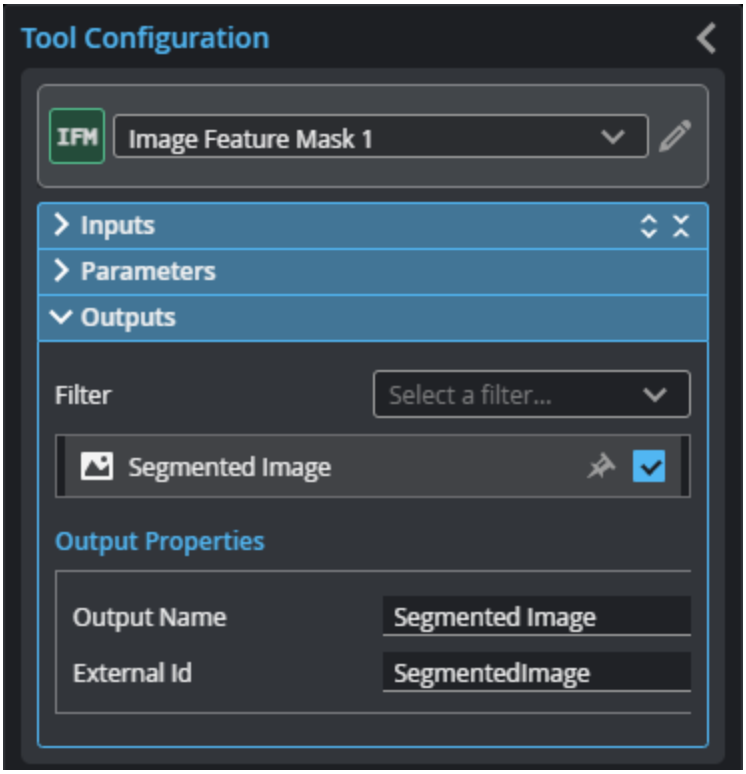
Parameters

Parameter	Description
Type	The shape type of the mask. It supports two options: **Polygon** - The tool needs at least 3 Point geometric features. **Circle** - The tool needs 2 or 3 point geometric features.
Point Count	The count of the input point features. For the Polygon **Type** option, the tool supports integers in the range of 3 to 10. For the Circle **Type** option, the tool supports integers of 2 or 3. After setting this parameter, set the point inputs in the Inputs section to the appropriate Point geometric features.
Padding Distance	The offset from the mask. When the value is greater than 0, the offset is outward. When the value is less than 0, the offset is inward.

Parameter	Description
Show Detail	Toggles whether to display the graphics of the mask region.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Data

Type	Description
Segmented Image	The cropped Image data.

Image Filter

The Image Filter tool provides several common vision processing filters that you can apply to the camera images, as well as two "cropping" filters that output a part of the image, letting you pre-process images to get more repeatable measurements. Any Image tool can use the resulting filtered image as input.

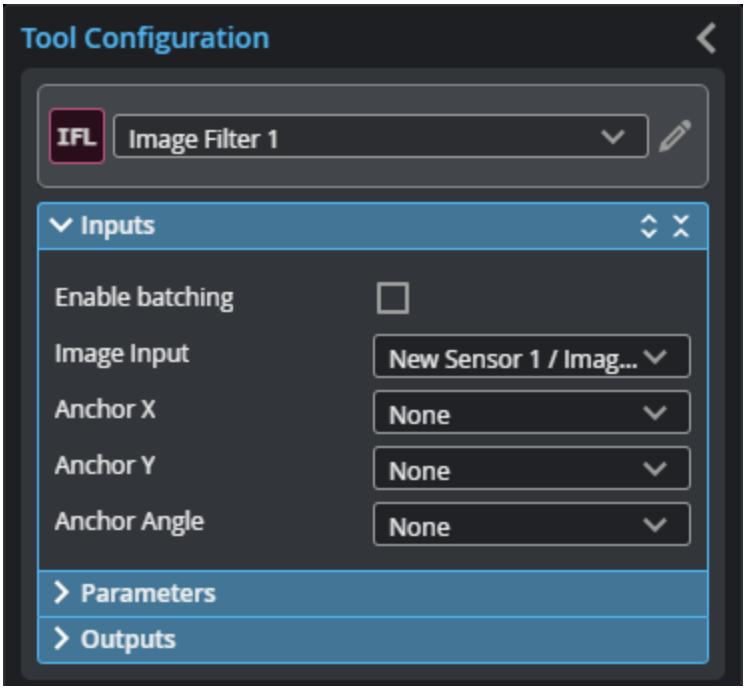
For a list of the filters, see *Filters* on page 380.

The Filter tool provides no measurements or decisions: it only outputs processed image data.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

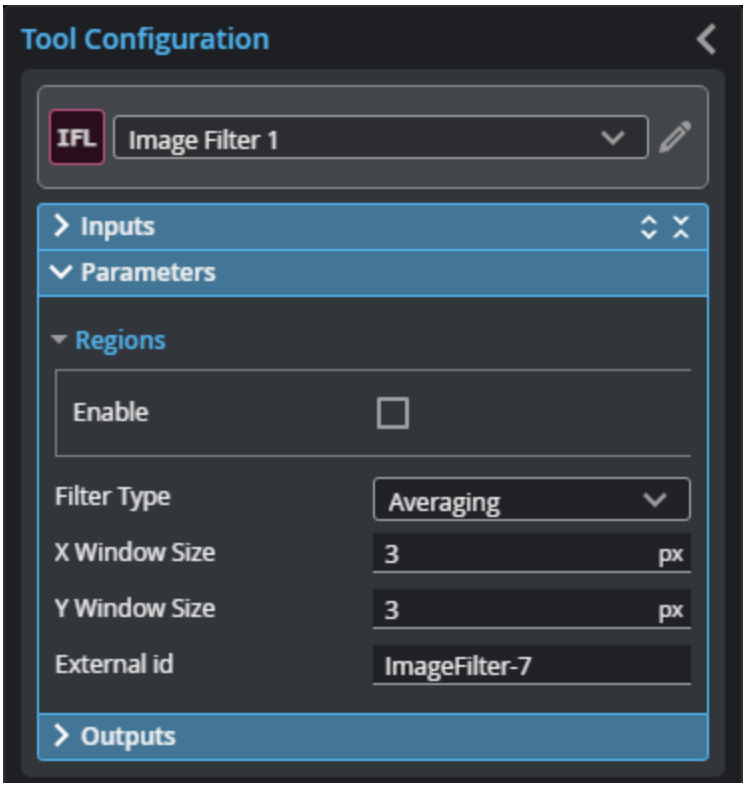
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor.

Name	Description
	For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings. Check Enable to configure regions.
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and

Parameter	Description
	"flexible" regions, see *Regions* on page 214.
Filter Type	The type of the filter. For information on the available filters, see *Filters* below.
Threshold	The threshold that the filter uses. (Not available on all filters.)
X Window Size Y Window Size	SThese parameters enable filtering along the X and the Y axis, respectively. Set the corresponding window sizes in **X Window Size** and **Y Window Size**.
Preserve Data Outside Region	Enable this setting when **Mask Mode** is set to **Include Data in Region** to include data outside the region in the Filtered Surface data output. Otherwise, the Filtered Surface data output only includes the data in the region. This setting is only displayed if regions are enabled.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

The following filters are available in the Filter tool.

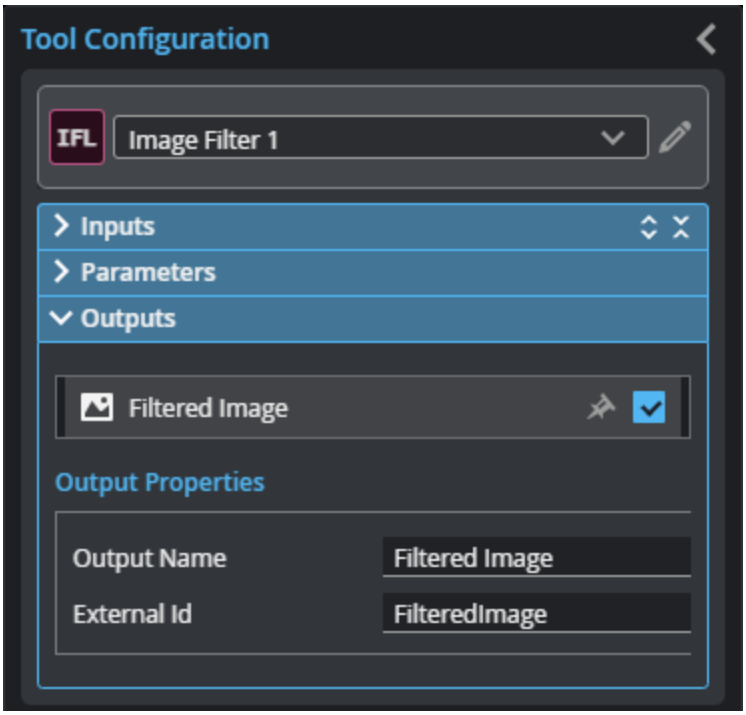
Filters

Name	Description
Averaging	Substitutes a data point value with the mean value of that data point and its nearest neighbors within the window or windows set in **X Window Size** or **Y Window Size**. 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 neighborhood.
Gaussian	A Gaussian filter. Set the **Sigma X** and **Sigma Y** parameters as required. Sigma X: Set to 0 to let the filter automatically calculate sigma based on the kernel size, using sigma = 0.3 * ((ksize-1) * 0.5 - 1) + 0.8 Sigma Y: Set to 0 to let Sigma Y be equal to Sigma X .
Median	Substitutes the value of a data point with the median calculated within the window or windows set in **X Window Size** or **Y Window Size** around the data point. If the number of valid (non-null) data points in the window is even, the average of the two values in the center is used. If the number of valid points is odd, the value in the center of the sorted list of values is used. Missing data points will not be filled with the median value calculated from data points in the neighborhood. With an odd-numbered window size, the output is at the center of the window. With an even window size, the output is 0.5 pixels to the right of the center (that is, using window / 2-1 values from the left, and window / 2 from the right.

Name	Description
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 Window Size** to 0.2, only points every 0.2 millimeters will be used. The filter generates points starting from the leftmost edge of the scan data, stepping in equal steps away from that side.
Erosion	Applies an erosion filter.
Dilation	Applies a dilation filter.
Opening	Erosion followed by dilation.
Closing	Dilation followed by erosion.
Morph Gradient	Applies a morphological gradient. The difference between dilation and erosion.
Sobel Magnitude	Applies a Sobel magnitude filter.
Laplacian	Applies a Laplacian filter. Useful for detecting areas of distinct edges. Set Kernel Size to an odd number less than 31.
Negative	Inverts the pixel values in the image.
Equalize	Improves contrast of the image.
Binarize	Converts the image to a binary image by comparing pixels with the threshold.
Percentile	Limits the scan data to pixels between the values you set in **High Percentile** and **Low Percentile**.
Relative Threshold	Crops scan data relative to user-specified intensity thresholds (High Threshold and Low Threshold). Pixels between the thresholds are included in Filtered Image.
Crop only	Crops the scan data to the user-defined region.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Data

Type	Description
Filtered Image	The filtered image, available for use as input in the **Input** drop-down in other tools.

Image Inference

The Inference tool lets you deploy a model you have trained outside of GoPxL using your preferred vision software and your own training workflow ("Bring Your Own Model").

Once you've uploaded your model to the tool and selected it in the **Model** drop-down, the tool runs your model on each frame it receives from a sensor or smart camera, and returns detections or classifications, measurement values that you can configure as anchors for other, and images that can be used by other tools.

The tool does not require a Pro Tools license.

> LMI has a commercial agreement with Ultralytics that lets you deploy models trained using Ultralytics open-source code in GoPxL for commercial use without needing to purchase an Ultralytics license.

The tool supports the following model formats:

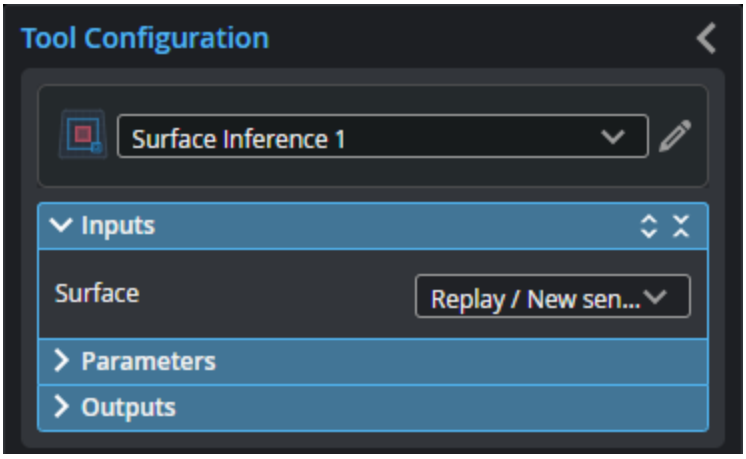
- PyTorch (.pt, .pth)
- ONNX (.onnx)
- Compiled TensorRT engines (.plan, .engine)

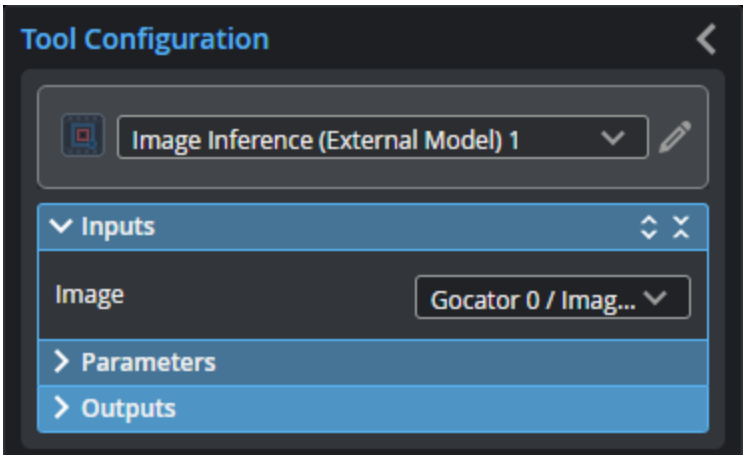
The tool only supports models trained for object detection and instance segmentation tasks. The model's task is automatically detected when the tool loads it.



For examples of scripts using the Inference tool outputs, see *Script Tool* on page 670.

Inputs



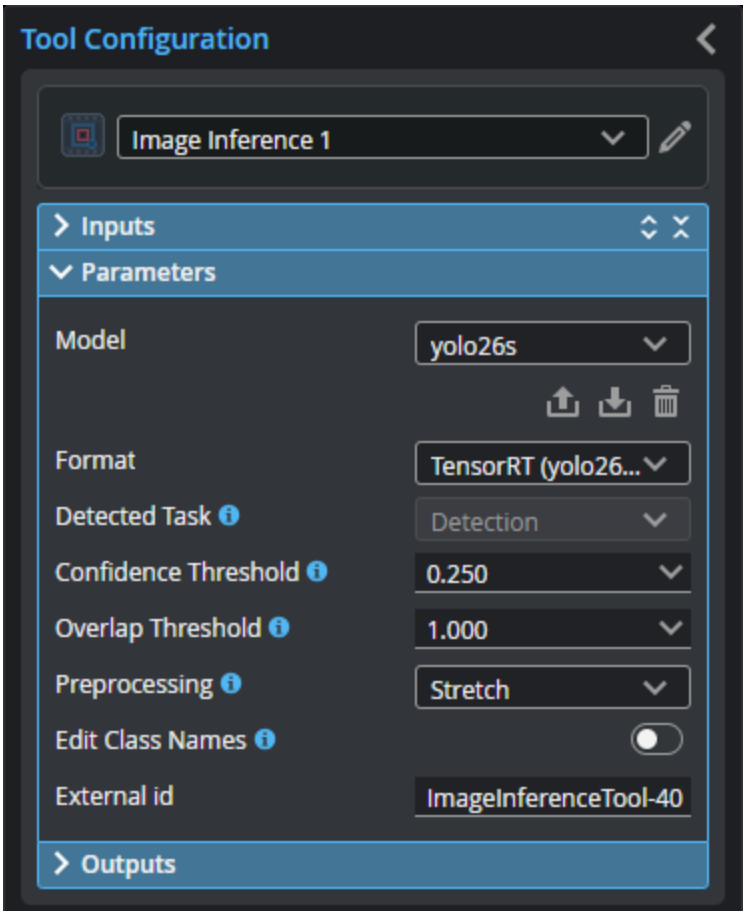


Inputs

Name	Description
Image Input	The data the tool applies measurements to or processes.

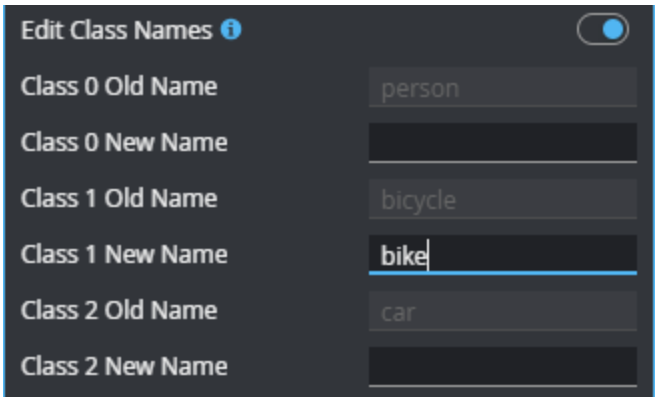
Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

Parameter	Description
Model	Selects an uploaded model for inference. After you select a model, the tool loads it and displays the recognized model type in the **Detected Task** field. Click Import () to upload a model from your PC. Click Download () to download a model to your PC. Click Delete () to delete the selected model.
Format	Drop-down list of available model formats in the dedicated storage folder for the selected model. If other model formats (.pt, .pth, .onnx, .plan, or .engine) exist in the dedicated storage folder for your model, this drop-down lets you switch between the imported format. Note that the tool performance is affected by which format you choose. If there are no optimized models available in the dedicated storage folder, an **Optimize** button will be displayed.
Detected Task	A read-only field displaying the task the loaded model was trained to perform. One of the following: Detection or Segmentation. If the type can't be determined, this field displays "Auto / Unknown".
Confidence Threshold	The minimum confidence score for a detection or segmentation to be reported.
Overlap Threshold	Helps reduce duplicate detections, due to overlapping bounding boxes. (Only displayed if the model supports duplicate suppression.) The value is between 0 and 1. With a lower value, the tool will remove duplicates more aggressively. A higher value is more permissive of overlap. Two bounding boxes that overlap perfectly would have an overlap value of 1.0. Two bounding boxes with almost no overlap would have an overlap value close to zero.
Optimize button	Click this button to optimize the model selected in **Model**. For more information, see *Optimizing Models* on page 389.
Preprocessing	How the tool fits the input frame to the model's input size. Match this to the preprocessing strategy the model was trained with. **Stretch (default)** The frame is resized directly to the model's input size, ignoring aspect ratio. Nothing is cropped or padded, so all content survives, but objects change shape (squashed or stretched) whenever the frame's proportions differ from the model's. Use this with models trained on stretched images. **Fill (center crop)** The frame is scaled up until it completely covers the model's input, then the overflow is cropped away symmetrically around the center. The frame is not distorted, no padding is applied, and data in the center is at full resolution. However, anything near the frame's edges is discarded before inference, so

Parameter	Description
	objects there can be missed entirely. **Fit** The frame is scaled down, preserving aspect ratio until it fits inside the model's expected input size. The padding needed to reach the model's exact expected input size depends on the model family (for example, Ultralytics YOLO typically pads with gray). The frame is therefore visible to the model at correct proportions and the padding color is whatever the backend chooses, not yours. **Fit and Pad** Scales the frame down and preserves the aspect ratio as with the Fit method, but the tool pads the result to the model's exact input size using the color from the **Padding** parameter (see below). Use this when you know what padding color the training pipeline used and want it reproduced exactly. **Tiling** The frame is never resized as a whole. Instead it's cut into tiles that the model runs on one by one (or in batches when **Batched Tiles** is enabled), and the per-tile detections are merged back into full-frame coordinates. For information on the related parameters, see *Tiling Parameters* on the next page.
Padding (with "Fit and Pad" preprocessing)	The color of the padding that the tool adds around the scaled frame. One of the following: Black, White, or Gray. The default is Gray, which matches the letterbox padding used during typical model training.
Edit Class Names	When this parameter is enabled, you can customize the names of the classes in the loaded model.  Class names are changed when you press Enter or move the cursor out of the field. The new class names are used in outputs, but the model is not modified.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Tiling Parameters

Parameter	Description
Tiling Mode	When **Preprocessing** is set to Tiling, set **Tiling Mode** to "Sliding Window" or "Grid" (see below), and optionally set an overlap (**Tile Overlap**). **Sliding Window** The tool moves a tile you define in pixels in **Tile Width** and **Tile Height** across both axes. **Grid** Divides the frame into a grid you define using the **Rows** and **Columns** parameters. Use this mode for small defects in large, high-resolution frames, where downscaling the whole frame to model size would remove the features you want to detect.
Tile Width Tile Height or Rows / Columns	When **Tiling Mode** is set to Sliding Window, the width and height of each tile in pixels (default 640). If possible, you should use the width and height of the images the model was trained on. When **Tiling Mode** is set to Grid, the number of tiles the frame is divided into along each axis, using the **Rows** and **Columns** parameters.
Tile Overlap	How much each tile overlaps its neighbors, as a percentage of the tile size (0-95%, default 0%). Increase the value for more seamless detection and segmentation around tile boundaries, at the cost of more tiles per frame.
Tile Geometry	Read-only field that describes the current tile layout the tool uses on the input. In Grid tiling mode, it shows the resulting tile size in pixels. In Sliding Window tiling mode, it shows the resulting number of rows and columns.
Batched Tiles	When this parameter is enabled, the tool passes tiles to the model in groups instead of one at a time, which can greatly reduce per-frame inference time. The parameter is only available if the loaded model supports batched input. Set **Max Batch Size** to the maximum number of tiles the tool passes to the model. The upper limit is set by the loaded model.
Max Batch Size	Maximum number of tiles processed together when **Batched Tiles** is enabled. The upper limit is set by the loaded model. This option is available for YOLO-family models only. To find the best batch size for your setup, first pick the tile size based on what the model was trained on, then raise this parameter gradually until you hit memory limits or throughput stops improving.

Optimizing Models

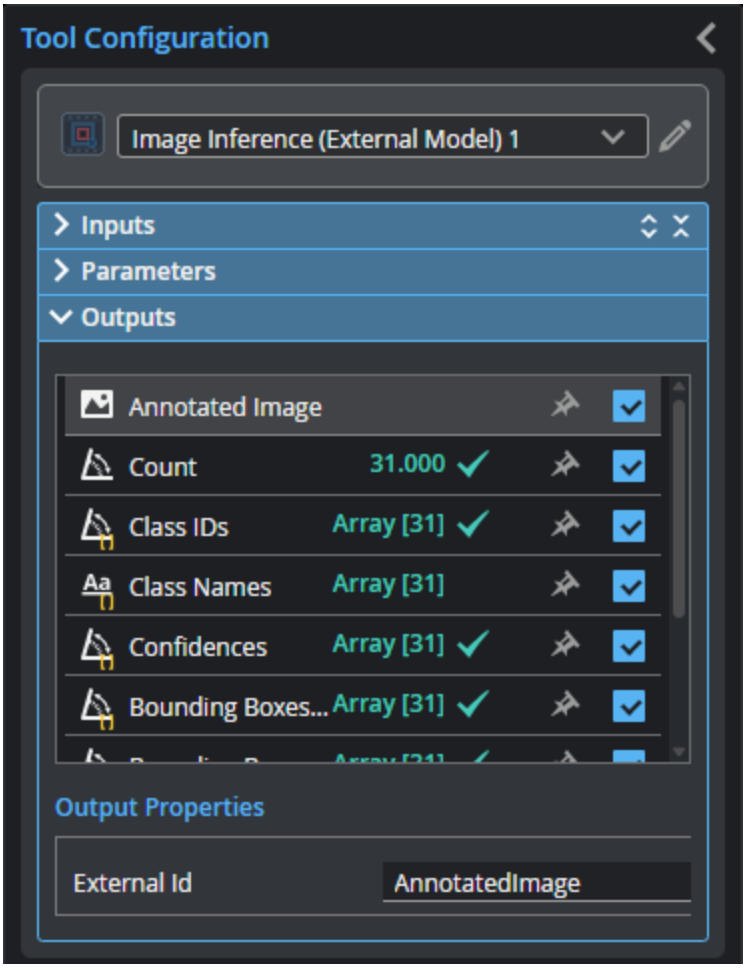
When a .pt, .pth, or .onnx model is selected in **Model**, an **Optimize** button appears. You should always optimize a model after you have uploaded it.

Some models (such as PyTorch or ONNX models) are uncompiled. Optimizing these models compiles them for the hardware of your GoPxL device (a sensor, GoMax device, or PC), and ensures they run faster.

Other models (.plan and .engine) are compiled but for a different device and may not load or run properly on a GoPxL device. Optimizing these models recompiles them for your hardware.

Optimization runs in the background and may take 5 to 15 minutes. While the tool optimizes the model, the **Optimize** button is greyed out and displays "Optimizing...". When optimization finishes, the tool automatically switches to the compiled model, and the button disappears. You should wait until the optimization is complete before continuing to configure your tool chain.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Note that some measurements return IDs, whereas others return class names. The IDs correspond to the class names added in the project. So for example, given the classes "lays classic", "cheetos", "doritos", and "ruffles", the corresponding IDs are 0, 1, 2, and 3. The ID values are ordered based on the order in which the classes were added in the project.

Measurements

Measurement
Count The number of detected or segmented objects.
Class IDs An array of the class indexes the model predicts for each object.
Class Names An array of the class names the model predicts for the objects.
Confidence Scores An array of the confidence scores for each detection or segmentation.
Bounding Boxes X (center) **Bounding Boxes Y (center)** Arrays of the X and Y positions of the center of the bounding boxes fitted to each object.
Bounding Boxes Width **Bounding Boxes Height** Arrays of widths and heights of the bounding boxes fitted to each object.
Major Axis **Minor Axis** An array containing the lengths of each detection along their major and minor axis. For segmentation models, the measurements are fitted to the segmented shape. For detection models, they are the bounding box's longer and shorter sides.

Features

Type	Description
Bounding Box Center Points	An array of Point geometric features representing the center the fitted bounding boxes.
Major Axis Line	An array of Line geometric features representing the angle of the major axis of the segmented objects. Only available with segmentation models.

Data

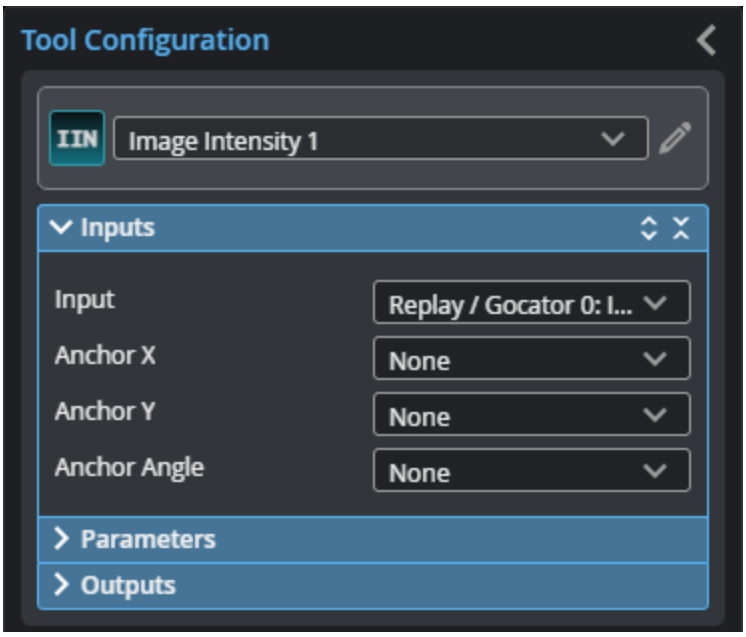
Type	Description
Annotated Image	The frame's image with an overlay displaying the class names, bounding boxes, and confidence scores of the detected or segmented objects.
Mask Image	The detected regions are drawn as colored areas on a black background. (Segmentation models only.)
Detection Region Images	An array of cropped images of the detections, taken around the detection's bounding box. With a segmentation model, the pixels outside the segmented target in each region image are zeroed according to the segmentation mask. Useful for feeding images of individual detections to other tools in the tool chain.
Inference Input	The actual image (or array of images when Tiling is used in **Preprocessing**) on which the tool runs the model. The image includes any preprocessing in the tool such as padding. Use this to verify what the model receives as input.

Image Intensity

The Image Intensity tool returns average, minimum, maximum, and median intensity values, as well as the contrast. The tool lets you choose whether to combine pixel colors using the formula described above, to output a single red, green, or blue channel, or an array. Optionally, you can output an array containing a histogram.

Note that when the tool returns array results, the order of the color channels is BGR.

Inputs



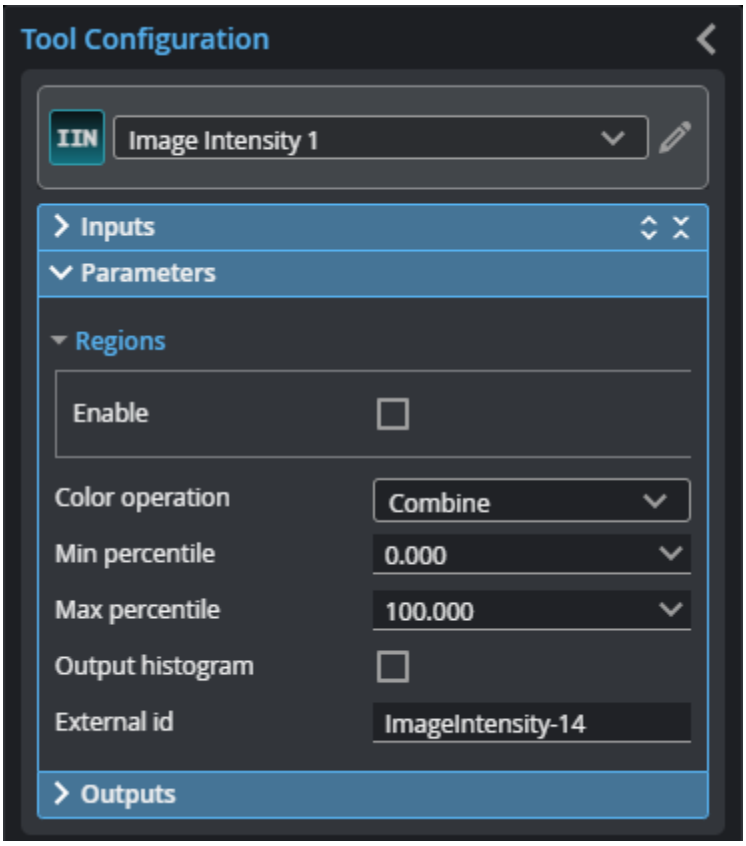
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

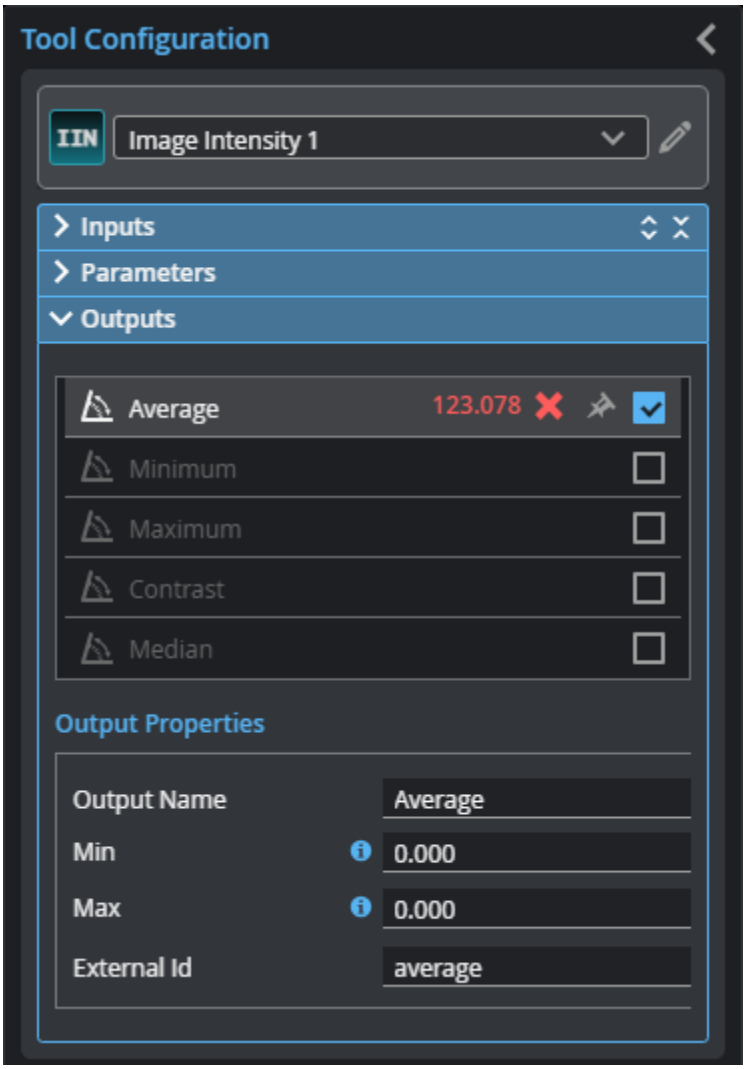


Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Color operation	One of the following: Combine, Red, Green, Blue, Array. **Combine** Converts RGB pixel colors to grayscale using the formula described above. **Red** **Green** **Blue** Returns the selected color channel only.

Parameter	Description
	Array Returns an array containing the values of the color channels in BGR order.
Min percentile Max percentile	Limits calculations to the pixels whose intensities fall between Min percentile and Max percentile.
Output histogram	When enabled, outputs a histogram of the image. Set the number of bins in
Histogram bin count	The number of bins into which the pixel intensities are combined in the histogram.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

The **Color operation** parameter determines which pixels the tool uses for the outputs.

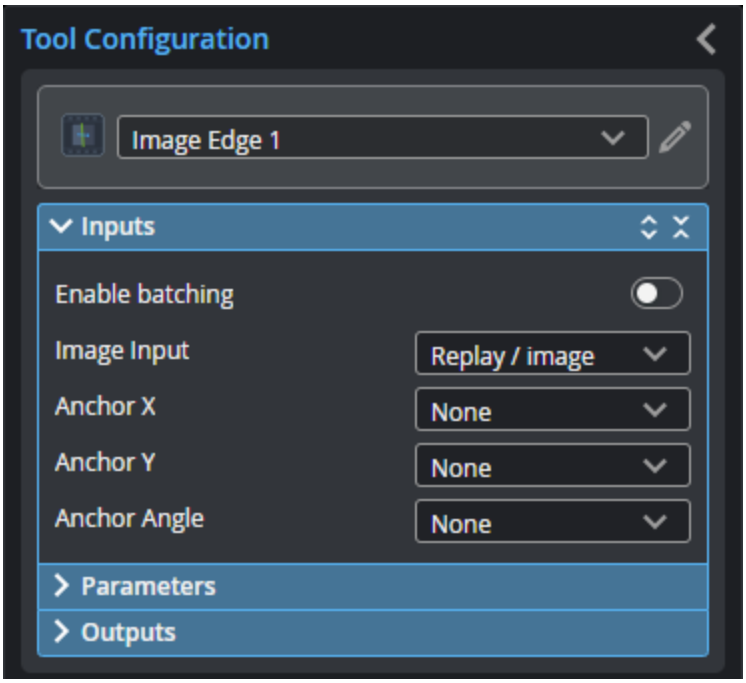
Measurements

Measurement
Average The average pixel intensity.
Minimum **Maximum** The minimum and maximum pixel intensity values.
Contrast The contrast of the pixels, based on the mean and standard deviation of the pixels.
Median The median of the pixel intensities.
Histogram When the **Output histogram** parameter is enabled, the tool outputs an array representing the histogram of the image, using the number of bins set in **Histogram bin count**.

Image Edge

The Image Edge tool fits a line to a step in an image, that is, rising or falling intensity transitions greater than the provided threshold.

Inputs



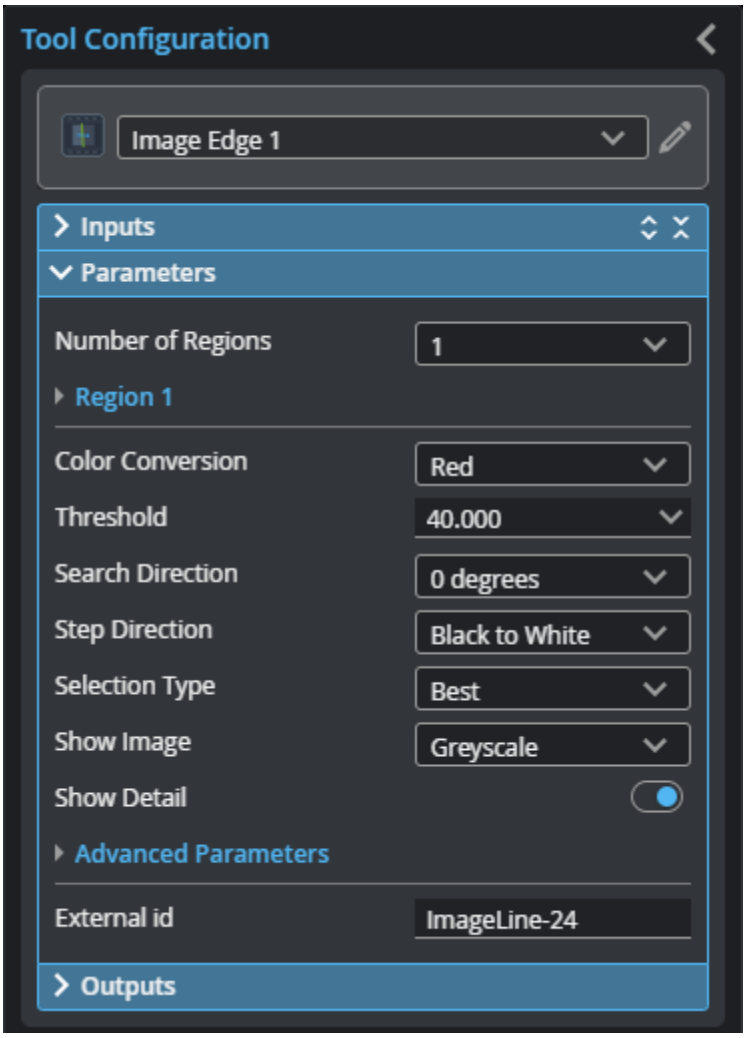
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.



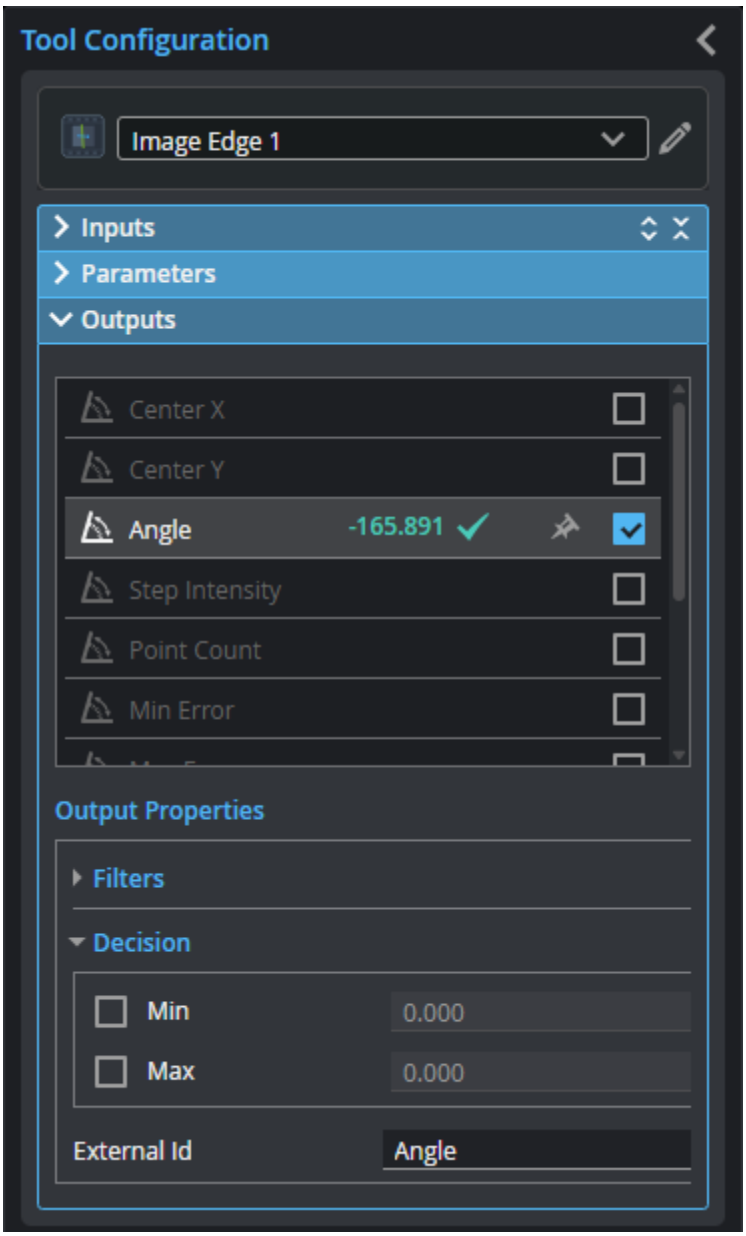
Parameters

Parameter	Description
Number of Regions	The number of regions the tool uses. You can use multiple regions to fit a line to non-contiguous features.
Region {n}	When expanded, displays the region-related settings.
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value: `pixel = 0.299R + 0.587G + 0.114B`

Parameter	Description
	When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.
Threshold	The intensity threshold above or below which a pixel will be considered a step point. On color devices, the RGB values are converted to an 8-bit monochrome image.
Search Direction	The search direction for steps. 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.
Step Direction	Determines whether intensity transitions should rise or fall. Note that this setting depends on the **Search Direction** setting for its interpretation of intensity transitions. One of the following: **Black to White or White to Black**: Searches for rising or falling intensity values. **Black to White**: Searches for rising intensity values. **White to Black**: Searches for falling intensity values.
Selection Type	One of the following: Best, First, or Last. Determines whether the best transition (greatest intensity difference), or the first or last transition exceeding the value of **Threshold** is used as a step point.
Show Image	Determines how the image (or the image data in a region) is displayed in the data viewer. One of the following: **Original** Shows the original image data. In color sensors, this will be the color image. **Greyscale** Shows the image data converted to greyscale. The conversion is determined by the option you choose in the tool's **Color Conversion** parameter.
Show Detail	Displays or hides step points and the direction arrow.
Advanced Parameters	When expanded, displays advanced parameters. **Outlier Fraction** The percentage of outlier points to exclude. Setting this to a small value can help the tool fit the line better. **Minimum relative change** The minimum change with respect to the value in **Threshold**.

Parameter	Description
	Path Spacing The spacing between sample paths perpendicular to the region's center line, along which the tool searches for steps. Smaller spacing increases processing time. **Step Smoothing** The size of the 1D window in which the tool applies an average to smooth pixel intensity values. The setting is useful for averaging out noise.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Measurements

Measurement
Center X **Center Y** The X and Y position of the intersection between the fitted line and the center line of the region.

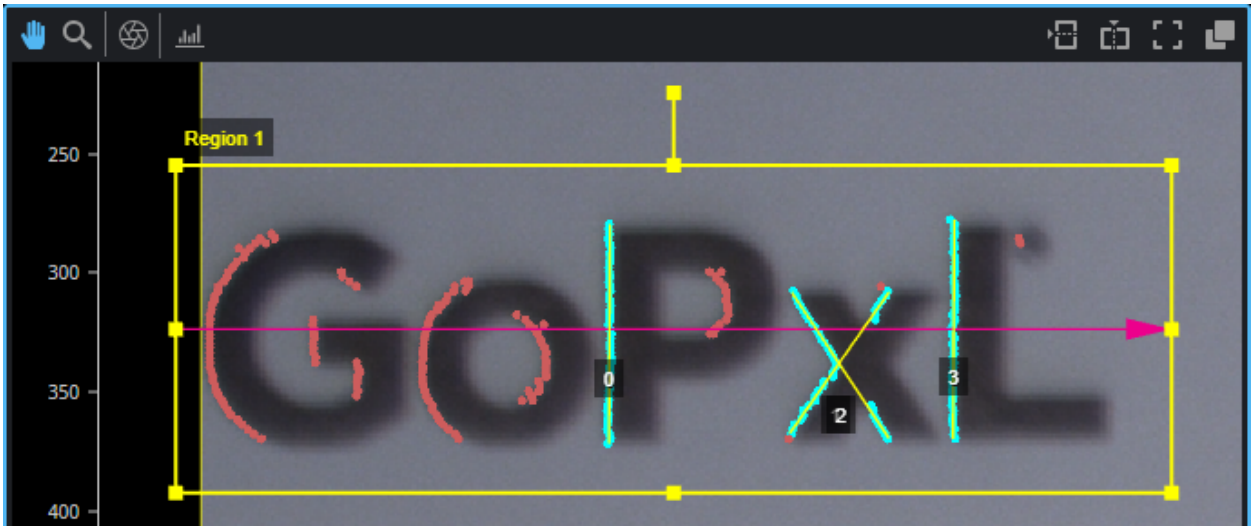
Measurement
Angle The angle of the line.
Step Intensity The average of the differences in intensity in the steps to which the tool fits a line.
Point Count The number of valid points used to fit the line.
Min Error **Max Error** The minimum and maximum error among the valid data points with respect to the fitted line (the minimum and maximum distances).
Width **Height** The width and height of the non-rotated bounding box fitted to the edge segment.

Data

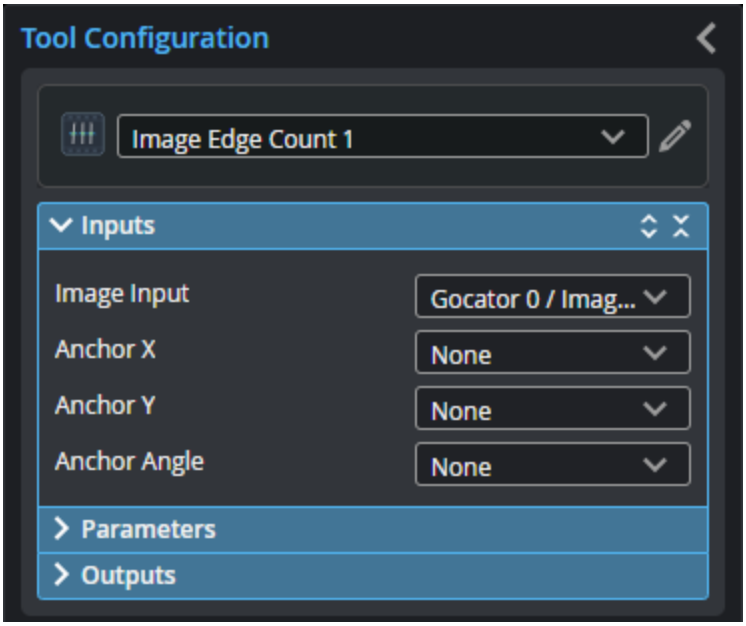
Type	Description
Edge	A Line geometric feature representing the fitted line.
Point	A Point geometric feature representing the center of the fitted line (Center X and Center Y).

Image Edge Count

The Image Edge Count tool detects and counts lines in one or more defined regions. To detect lines, the tool searches along paths perpendicular to a region's center line. Steps in intensity (increasing, decreasing, or both) greater than a user-defined threshold are identified as potential edge points; edge points can be excluded by settings in the tool's Advanced Parameters section. The tool fits a line to the accepted ("inlier") edge points, and outputs measurements related to the detected lines.



Inputs



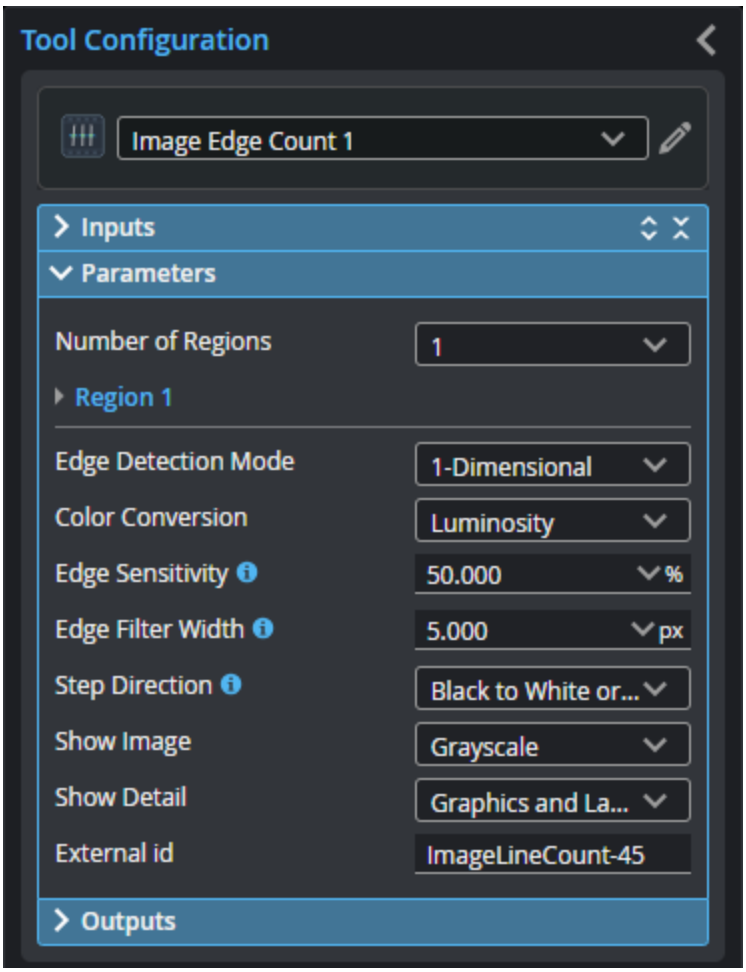
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.



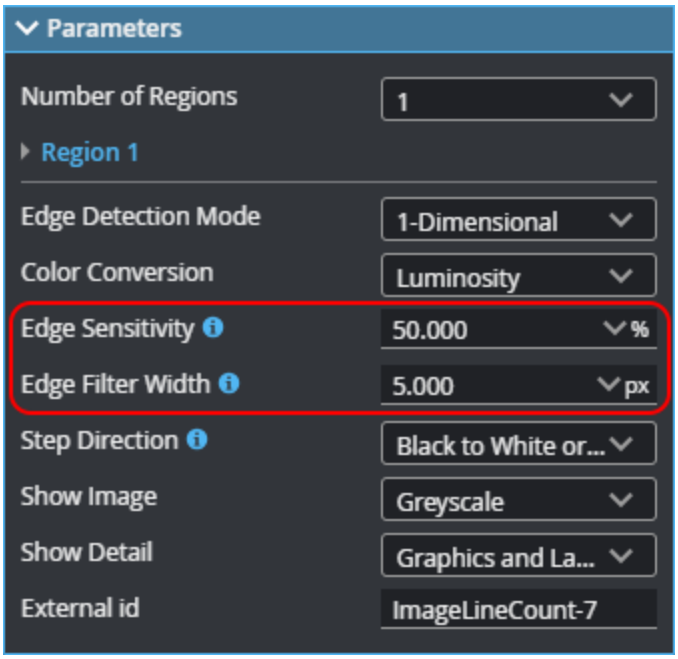
Parameters

Parameter	Description
Number of Regions	The number of regions the tool uses. You can use multiple regions to search for lines in non-contiguous features.

Parameter	Description
Region {n}	When expanded, displays the region-related settings.
Edge Detection Mode	The mode the tool uses. One of the following: **1-Dimensional** The tool evaluates a row of pixels (along the path defined by the center of the region of interest), and returns a point. This method lets you identify an edge more quickly. For descriptions of the mode-specific parameters, see *1-Dimensional Mode* below. **2-Dimensional** For descriptions of the mode-specific parameters, see *2-Dimensional Mode* on the next page.
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value: `pixel = 0.299R + 0.587G + 0.114B` When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.
Step Direction	Whether the tool looks for increasing intensity values (Black to White option), decreasing values (White to Black option) intensity values, or both, when placing edge points.
Show Detail	Displays or hides edge graphics (points and the direction arrow) and instance labels. Cyan edge points are used to fit a line. Red edge points are outliers and are not used for line fitting. Can be one of the following: Graphics, Graphics and Labels, Off.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

1-Dimensional Mode

When you set **Edge Detection Mode** to 1-Dimensional, set the following parameters.

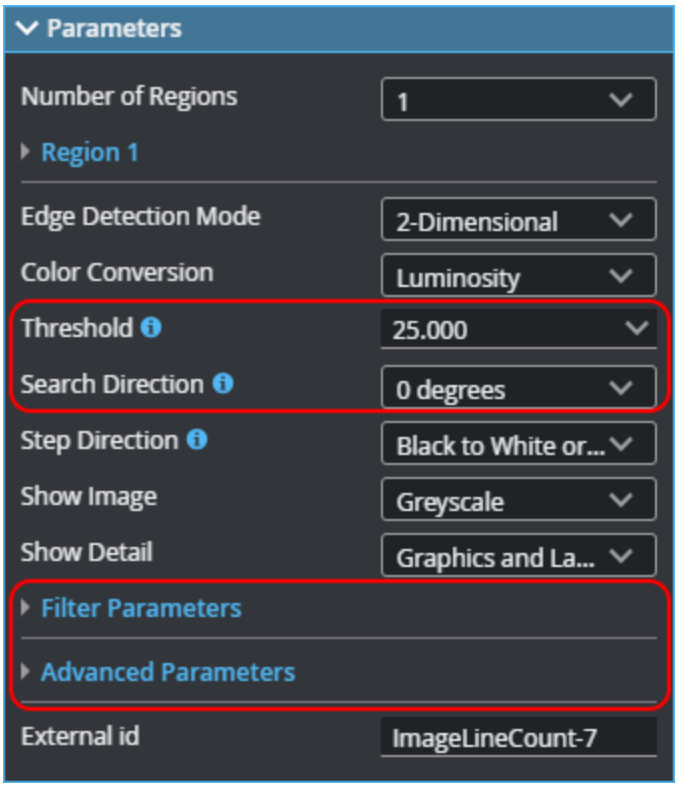


1-Dimensional Parameters

Parameter	Description
Edge Sensitivity	The edge detection threshold as a percentage of the strongest response. A lower value detects weaker edges.
Edge Filter Width	How heavily the brightness profile is smoothed, defined as the width of a window in pixels. Larger values smooth out more noise. Smaller values preserver fine details.

2-Dimensional Mode

When you set **Edge Detection Mode** to 2-Dimensional, set the following parameters.



2-Dimensional Parameters

Parameter	Description
Threshold	The intensity threshold above or below which a pixel will be considered a step point. On color devices, the RGB values are converted to an 8-bit monochrome image.
Search Direction	The search direction for steps. Can be 0, 90, 180, or 270 degrees. Choose a value that is roughly perpendicular to the edges on the target. The direction is indicated by an arrow in the data viewer.

In 2-Dimensional mode, you can set additional parameters in expandable **Advanced** and **Filter** sections.

Advanced Parameters

Parameter	Description
Minimum relative change	The minimum intensity change as a percentage of the strongest step in the all of the regions. Use this to filter weak edges relative to the strongest.
Path Spacing	The spacing between sample paths perpendicular to the region's center line, along which the tool searches for steps. Smaller spacing increases processing time.
Step Smoothing	The size of the window in which the tool applies an average to smooth pixel intensity values, before edge detection. The setting is useful for averaging out noise.

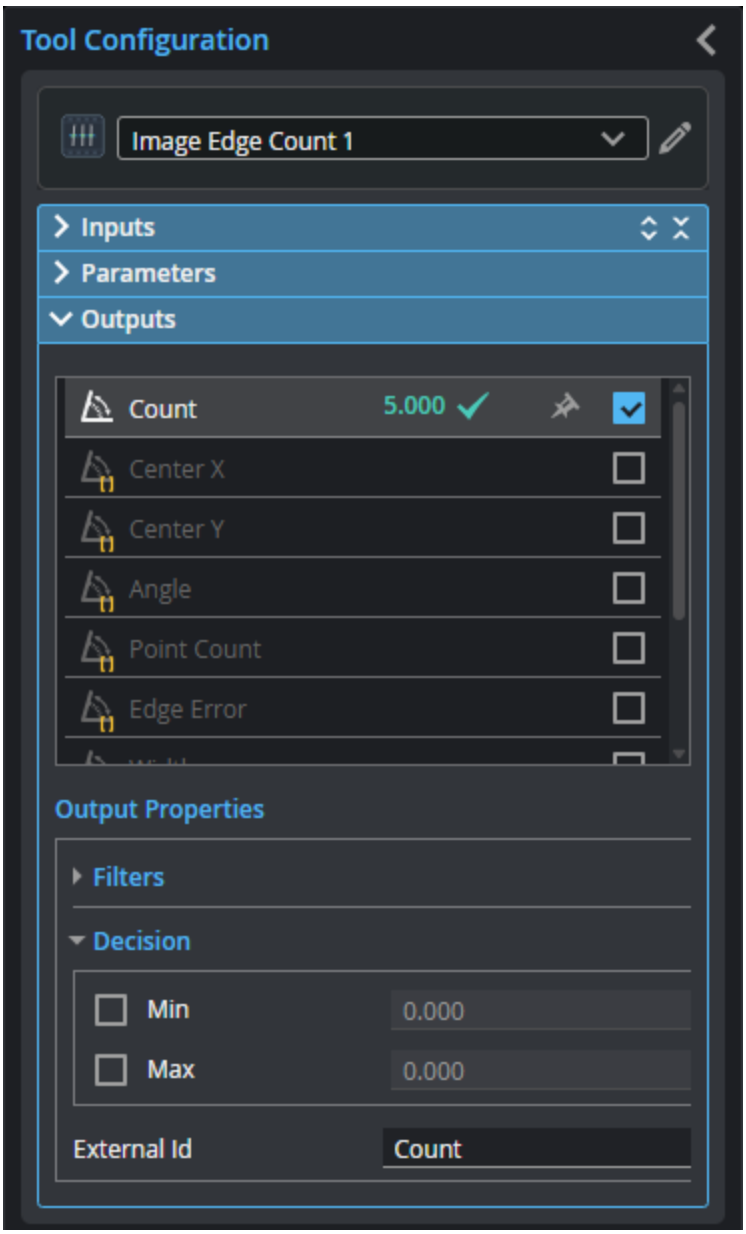
Parameter	Description
Line Thickness	The maximum distance an edge point can be from a fitted line to be considered part of it. Increasing this value merges nearby parallel lines.
Angle Tolerance	The amount of angular variation allowed when fitting a line. A higher value helps fitting to slightly curved or imperfect edges.
Show Outliers	Whether to display points not used for line fitting.

Minimum points: The minimum number of valid edge points for a line to be detected. Set this to a value that will exclude short, unwanted lines.

Filter Parameters

Parameter	Description
Minimum points	The minimum number of valid edge points for a line to be detected. Set this to a value that will exclude short, unwanted lines.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Measurements

Measurement
Count The number of lines the tool detected.
Center X

Measurement
Center Y The X and Y position of the center of a line.
Angle The angle of a line.
Point Count The number of edge points used to fit the line.
Edge Error The maximum deviation among the edge points from the fitted line.
Width **Height** The width and height of the non-rotated bounding box fitted to each edge segment.

Features

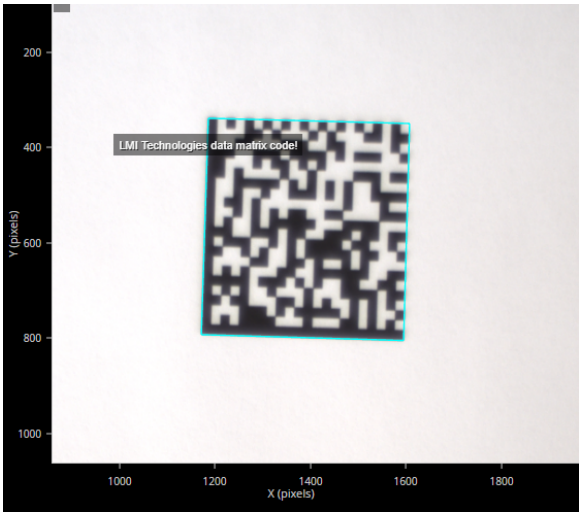
Type	Description
Edge	A Line geometric feature representing the fitted line.
Point	A Point geometric feature representing the center of the fitted line (Center X and Center Y).

Image Matrix Code

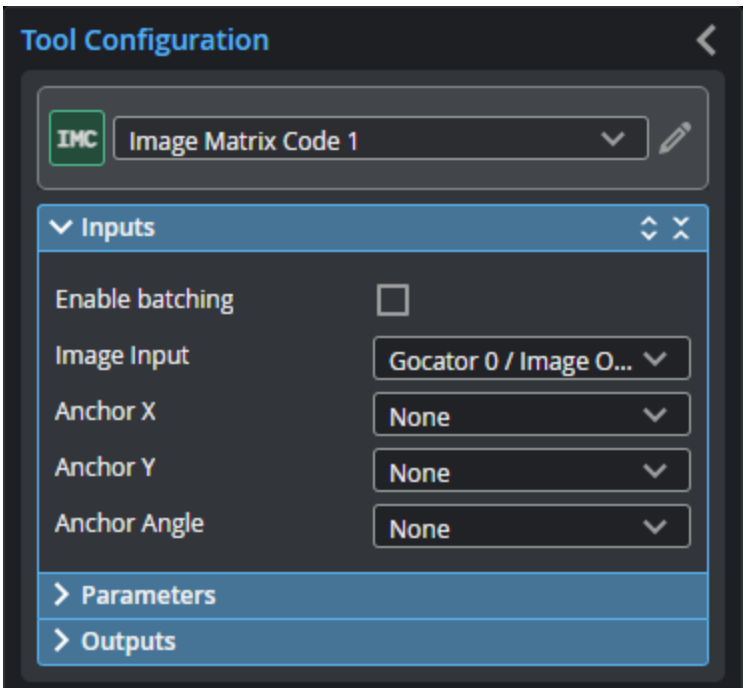
The Matrix Code tool lets you decode data matrix codes in Image data. The tool only supports grayscale images.

The tool returns the decoded string, as well as other measures such as its dimensions and the ECC family to which it belongs. The tool also provides measurements that can help you assess the print quality of the code symbols. Indicators from the following standards are available:

- ISO 15415
- ISO 29158
- SEMI T10



Inputs



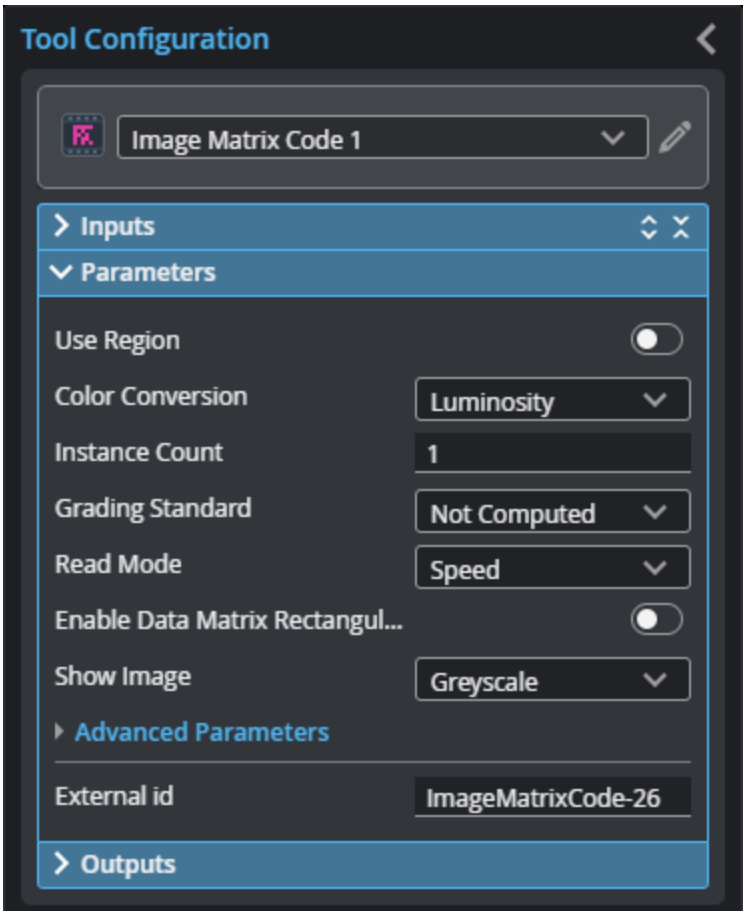
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

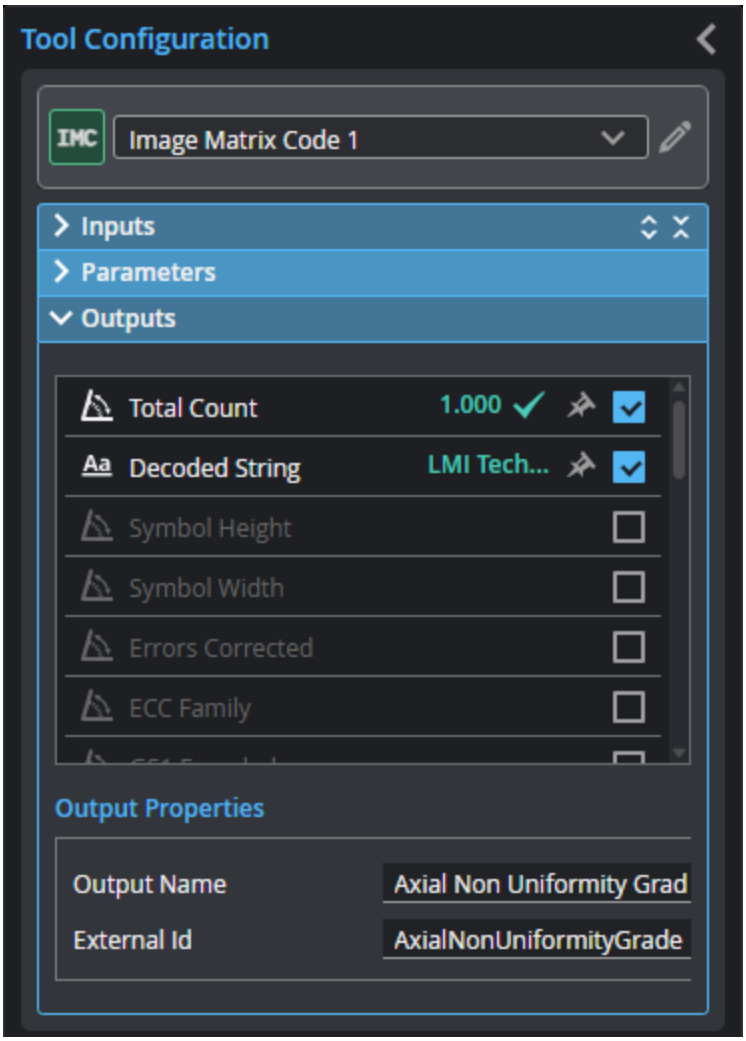
Parameter	Description
Use Region Region	Enable **Use Region** and set the values for the regions.
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value: `pixel = 0.299R + 0.587G + 0.114B` When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.

Parameter	Description
Instance Count	The maximum number of matrix codes to decode in a frame.
Grading Standard	The standard the tool will use to assess the quality of the matrix code. Grading measurements are added in the tool's **Output** section. One of the following: • ISO 15415 • ISO 29158 • SEMI T10
Compute Grading	The type of grading the tool returns in the grading measurements. One of the following: **Number** - Grade measurements return a numerical value between 0 and 4. **Letter** - Grade measurements return A, B, C, D, or F. **Letter and Number** - Both types of outputs are returned. Numerical measurements are distinguished from letter-based ones by "Number" in the name.
Read Mode	Whether the tool prioritizes speed or detection quality. **Speed** The tool stops searching for matrix codes as soon as one of the following happens: • It finds the number of codes selected in **Instance Count**. • It has finished processing the image. This mode results in shorter processing times. **Quality** The tool tries to improve its detection until it has finished processing the image. This mode results in better grading results.
Enable Datamatrix Rectangular Extension	Enables decoding of Datamatrix Rectangular Extension (DMRE).
Show Image	Determines how the image (or the image data in a region) is displayed in the data viewer. One of the following: **Original** Shows the original image data. In color sensors, this will be the color image. **Greyscale** Shows the image data converted to greyscale. The conversion is determined by the

Parameter	Description
	option you choose in the tool's **Color Conversion** parameter.
Advanced	Expandable section containing advanced settings. **Decode Damaged Code** Whether to allow permissive decoding (decode codes containing partially incorrect information) and return unreliable codes (objects that are likely data matrix codes, but can't be decoded). **Set Code Dimensions** Sets the minimum and maximum number of pixels allowed for the sides of a data matrix code. Matrix codes with dimensions outside the range will not be detected. Set **Min Dimension** and **Max Dimension** to the required number of pixels. These dimensions override the dimensions resulting from using the optimization engine and reset the engine. **Optimization Engine** Place a representative matrix code under the camera, and click **Learn image** to determine the best options to use to read matrix codes. Click **Reset** to clear.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Some measurements are only displayed when you select a grading standard in **Compute Grading** (see *Parameters* on page 411). "Modules" refers to the cells or "dots" in the grid of the matrix code.

Measurements (all grading standards)

Measurement
Total Count The number of matrix codes decoded.

Measurement
Decoded String The decoded string, packed as an array of float-point values.
Symbol Height **Symbol Width** The number of vertical and horizontal cells in the matrix code, respectively.
Errors Corrected The number of corrected modules.
ECC Family The family the matrix belongs to (for example, 200 means the code belongs to the ECC200 family).
GS1 Encoded Returns 1 if the decoded string uses the GS1 standard.
Reliability A confidence score related to whether the code is actually a data matrix.

When **Compute Grading** is set to "Letter and Number", the tool outputs both "Number Grade" measurements containing a value between 4 and 0 and "Grade" outputs that contain A, B, C, D, or F. If the setting is either "Letter" or "Number", the tool only returns "Grade" outputs, which contain the selected type.

Measurements (ISO 15415)

Measurement
Decode Grade **Decoding Number Grade** Whether the code was decoded.
Axial Non Uniformity Grade **Axial Non Uniformity Number Grade** Grade for uneven scaling along the X and Y axis.
Grid Non Uniformity Grade **Grid Non Uniformity Number Grade** How much the measured position of the grid intersection point deviates from the expected position. Codes sheered along the X and/or Y axis can result in a low grade. The grade can be unstable and its usefulness is limited.
Symbol Contrast Grade **Symbol Contrast Number Grade** The difference between the dark and light areas of a code, including the quiet zones.
Modulation Grade **Modulation Number Grade** Uniformity of the reflectance of dark and light areas throughout the code. Irregular or mottled regions can

Measurement
result in a low grade.
Reflectance Margin Grade **Reflectance Margin Number Grade** How distinguishable the modules are relative to the global threshold (the mean reflectance of the brightest and the darkest module).
Fixed Pattern Damage Grade **Fixed Pattern Damage Number Grade** A measure of any damage (missing data or data where it is not expected) to the finder pattern (the L-shaped along the left and bottom of the code), the quiet zone (margins outside the matrix), or the clock track (the alternating "dots" along the top). The value returned is an average of the grades of these areas of concern. A higher value indicates the matrix code is more correctly located and identified.
Unused Error Correction Grade **Unused Error Correction Number Grade** Grade representing how little error correction to reconstruct data in damaged or poorly printed codes was used. A higher grade means less error correction was applied.
Horizontal Print Growth **Vertical Print Growth** **Horizontal Print Growth Number** **Vertical Print Growth Number** How much the matrix code may have grown or shrunk compared to the expected size horizontally and vertically, respecitively.

Measurements (ISO 29158)

Measurement
Cell Contrast Grade **Cell Contrast Number Grade** Relative difference between the mean reflectance of the darkest and brightest cells in the code.
Cell Modulation Grade **Cell Modulation Number Grade** The variability and reliability of the black and white colors. A low grade means the tool will have difficulty distinguishing between black and white cells.
Minimum Reflectance Grade **Minimum Reflectance Number Grade** The ratio between the smallest and largest gray values in the barcode. For the "Number" grade, the value is 4 if the smallest gray value of the barcode is <= half of its highest value, and 0 otherwise.
Fixed Pattern Damage Grade **Fixed Pattern Damage Number Grade**

Measurement
How likely the symbol is correctly located and identified in the code.
Mean Light In Required Bounds **Mean Light In Required Number Bounds** Whether mean light is 70%~86% of the maximum gray level.
Mean Light **Mean Number Light** The mean value of the pixels in the center of the light cells.

Measurements (SEMI T10)

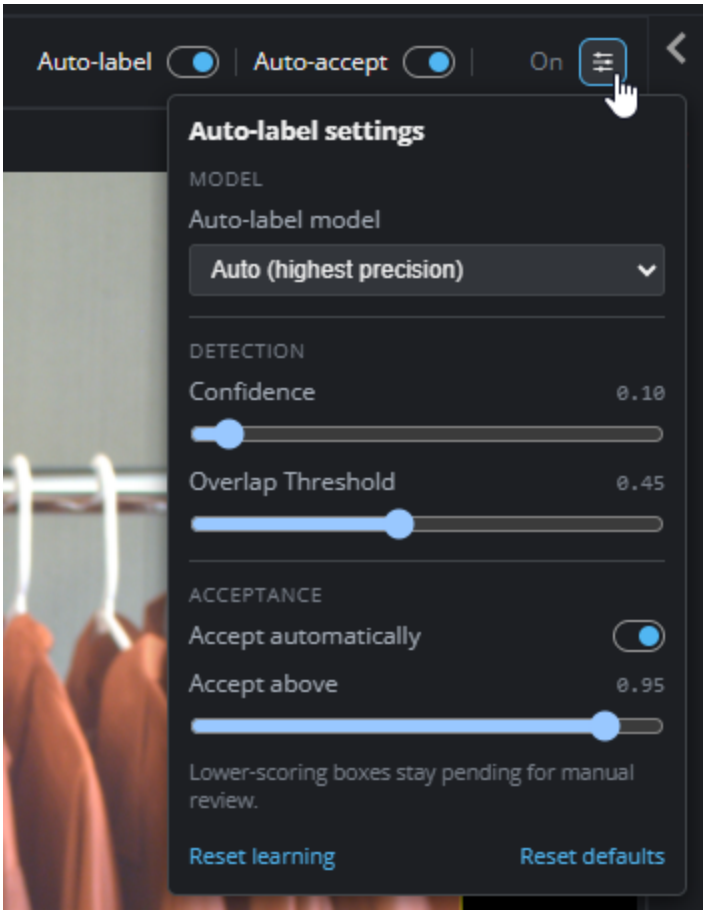
Measurement
Symbol Contrast SNR **Symbol Contrast SNR** The relative strength of the noise in the code. A higher grade means less noise is present.
Vertical Mark Growth **Vertical Mark Growth** **Horizontal Mark Growth** **Horizontal Mark Growth** The relative horizontal and vertical size of a mark (dark cell) compared to a space (white cell), respectively. The highest grade means cells are the same size.
Vertical Mark Misplacement **Vertical Mark Misplacement** **Horizontal Mark Misplacement** **Horizontal Mark Misplacement** The relative distance between the expected positions cell centers and their measured positions.
Data Matrix Cell Height **Data Matrix Cell Height** **Data Matrix Cell Width** **Data Matrix Cell Width** The height and width of the cells.
Finder Pattern Defects **Finder Pattern Defects** Similar to Cell Defects, but limited to L-shamped finder pattern. The tool may not recognize a code as valid if this is too high.
Cell Defects **Cell Defects** Ratio of pixels that have the wrong color to the total number of pixels in the code. Only returned if the code can be decoded.

Image Object Detector

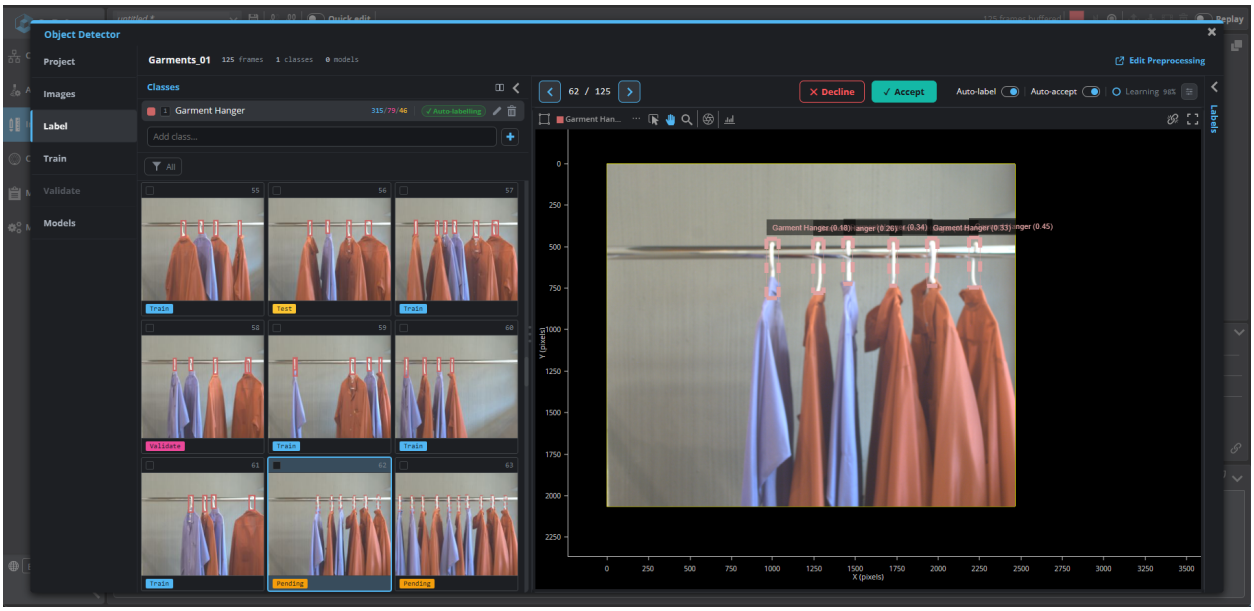
The Image Object Detector tool locates and counts objects (parts, defects, or features) in images, returning a probability score of each object, in addition to determining presence or absence of an object. The tool also returns a bounding box for each located object, which lets tools later in the tool chain act on the position of the bounding box. You should train each class on 75 to 150 images.

The tool also provides an auto-label function that helps make labeling a dataset much faster. After you have labeled two frames and then start labeling a third frame, the tool learns how to make label predictions using the first two frames.

You can also enable Auto-accept and set the threshold above which predictions will be automatically accepted.

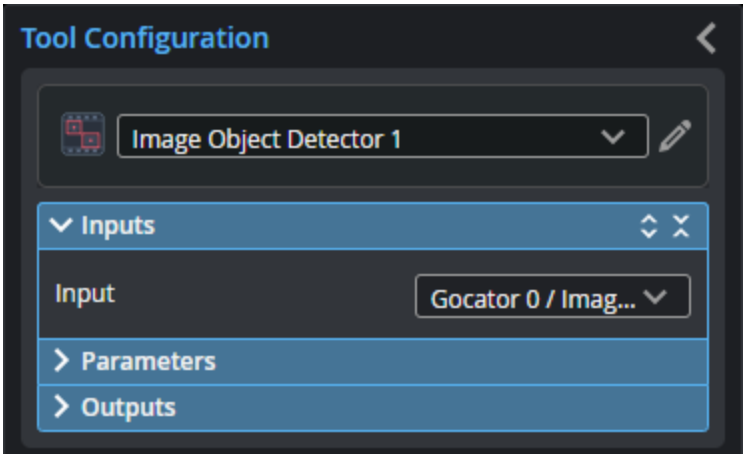


After the tool has predicted labels in each frame, you accept or decline the labels the tool proposes in the frame by clicking **Decline** or **Accept** above the data viewer. Accepting a frame's predicted labels, moves to the next frame.



You can delete, adjust, and reposition the label bounding boxes, or change the class of the label, before accepting the predictions in the frame. If you decline the Auto-label predictions in a frame, the tool removes the labels and moves to the next frame. As you progress through the frames, the tool continually learns from the changes you make, or the frames you accept.

Inputs



Inputs

Name	Description
Image Input	The data the tool applies measurements to or processes.

Development Workflow

The high level workflow is as follows:

1. Collect representative frames of the targets you need to inspect in order to train a model.

2. Save the frames to a .gprec file.
3. Create a pre-processing toolchain and train a project's model. (You will need to re-use the pre-processing toolchain for use in production.)
4. Deploy the model and connect post-processing tools.
5. Optionally, collect new production data for model retraining.

Collect dataset

Unless you are using a previously created dataset, you must create the dataset that the tool will train a model on before it can process new production data. To do this, perform the following steps:

1. Acquire 75 to 150 images for each of the classes you intend to add to the project.

   You can acquire the data in a single .gprec file or to multiple files. If you record scans to a single file, grouping each type together will make it easier to label them later in the wizard. Make sure there are no empty frames.

   If you have previously acquired images, you can also upload them from the Label Project page; for more information, see *Create New or Open Project* on page 424.
2. (Optional) Save a separate .gpsup file with the device configuration settings such as exposure time and related imaging parameters.

   Ensure the same parameters are used on the production system. Otherwise, you will need to collect new training data.

Implement a Preprocessing Toolchain

You typically need to create a preprocessing tool chain for your application. The preprocessing tool chain limits variation in the scan data to the region or surface of interest. With less variation in the scan data, the total number of frames needed for training is reduced.

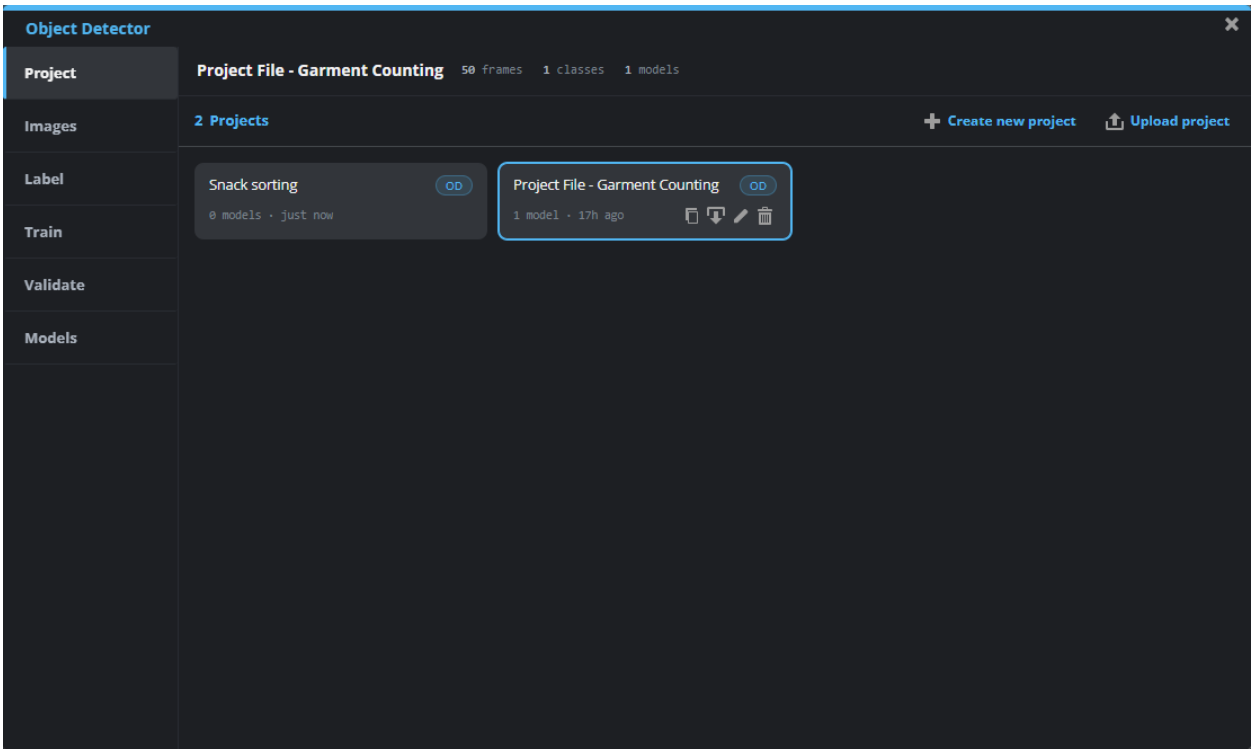
The preprocessing tool chain must not change between training and inference. Otherwise, the system will not function properly. You must save your preprocessing toolchain in the .gpsup file created previously or using a separate .gpjob file.

> You *must* use the same preprocessing toolchain for training and inference to ensure proper object detection.

Train Model

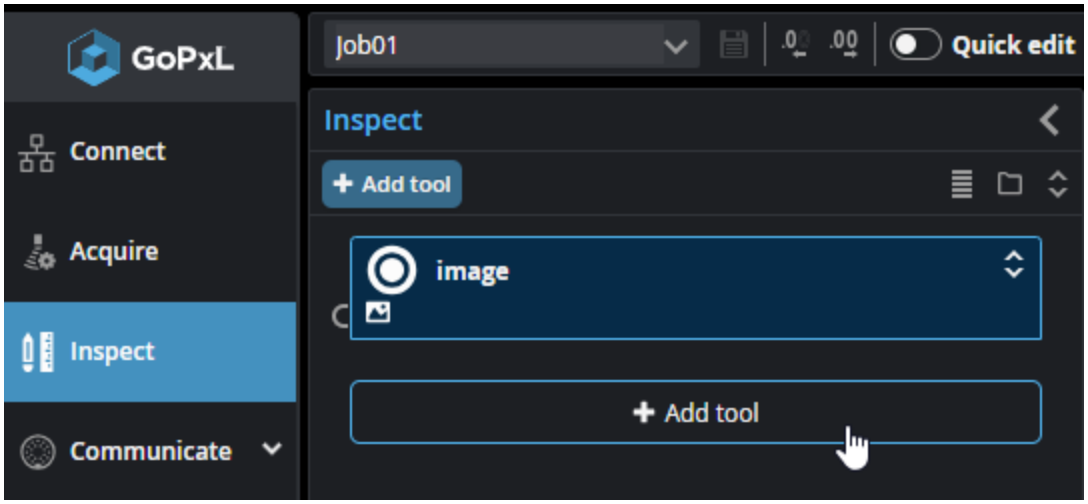
The AI-based Pro Tools provide a step-by-step training wizard to help you create and manage your AI projects, prepare images for training, and train and validate models. The steps are provided on separate tabs in the wizard.

> You may wish to maximize your browser the first few times you use the wizard to see the labels of the commands, rather than just their icons.

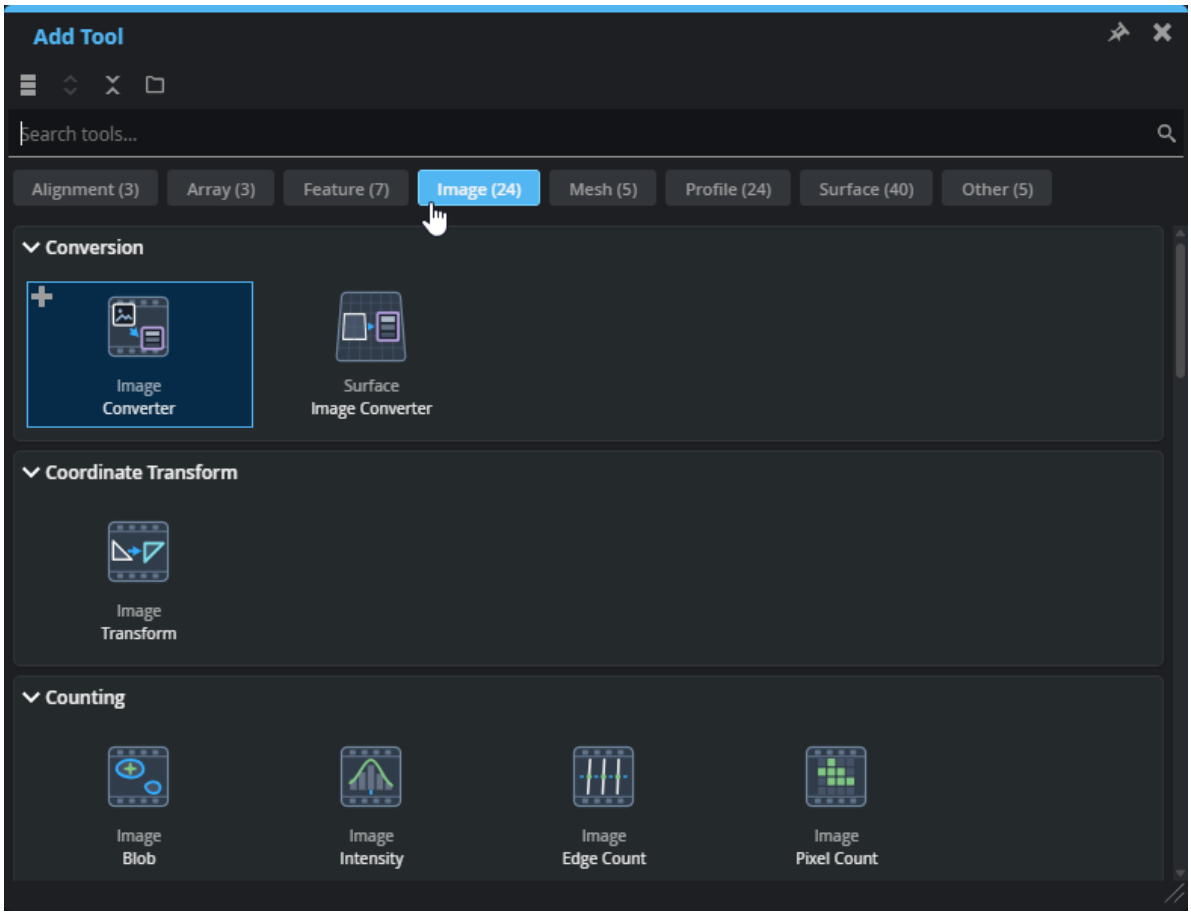


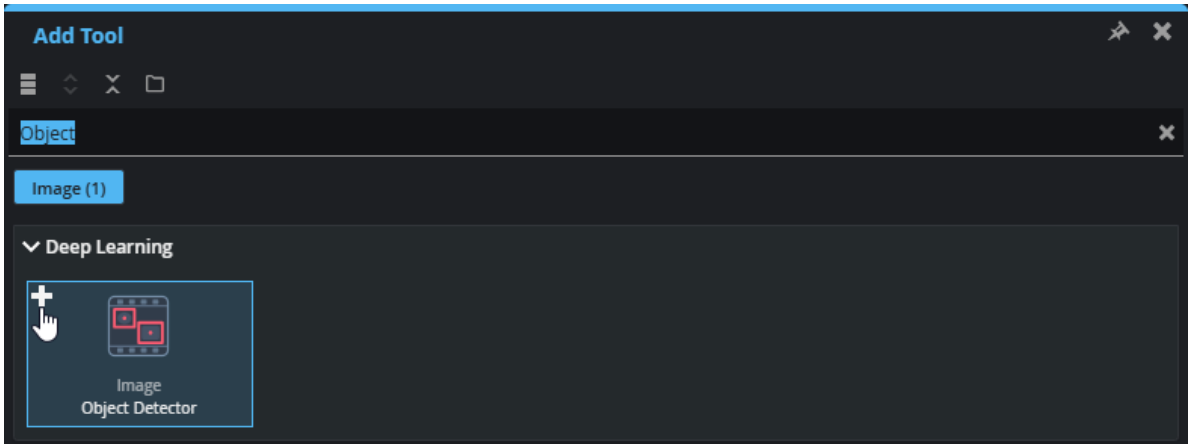
ADD THE TOOL

1. Add and connect the tool to the output of the preprocessing tool chain (if necessary).

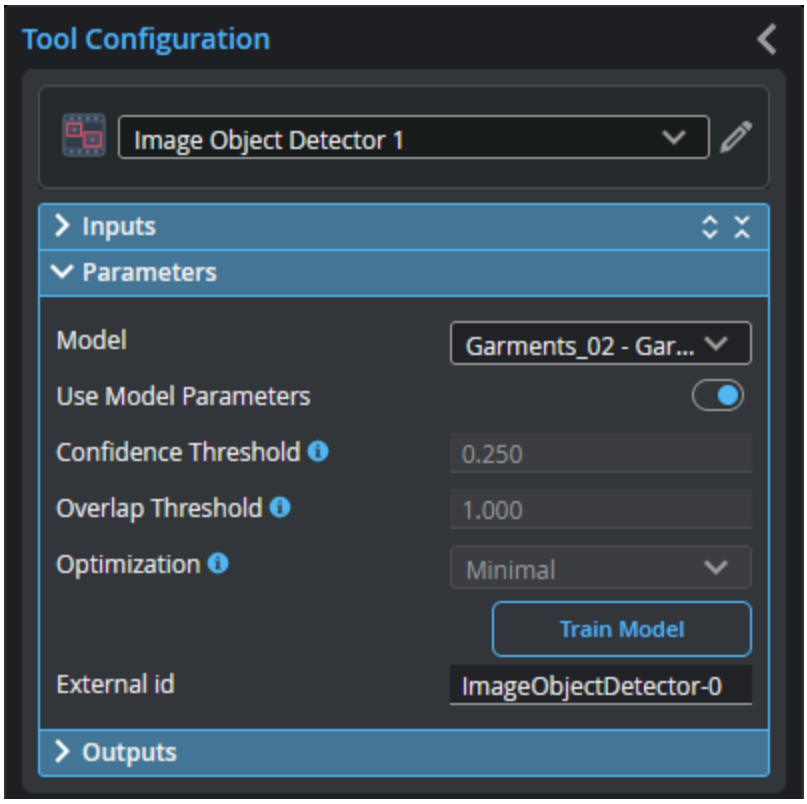


   In the **Add Tool** dialog, find the tool. To make this easier, you can filter tools by type and tool name text (in **Search tools...**).

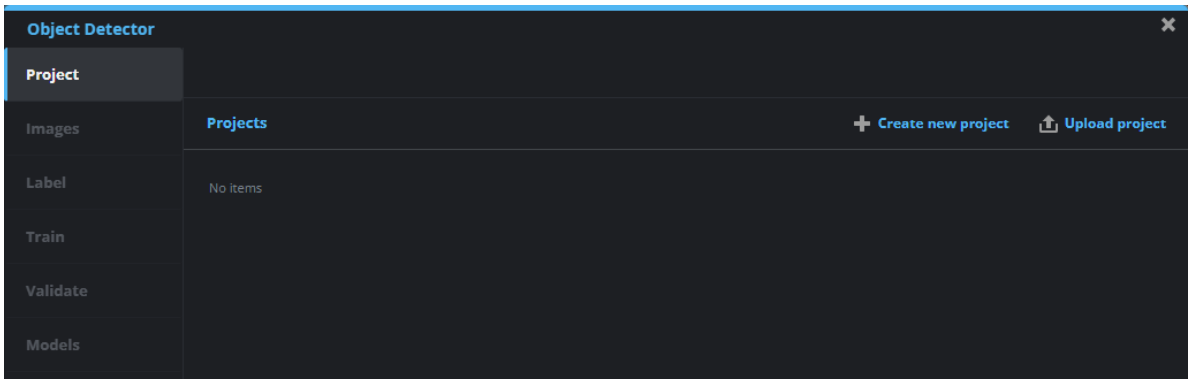




2. In tool's **Parameters** section, click the **Train Model** button.

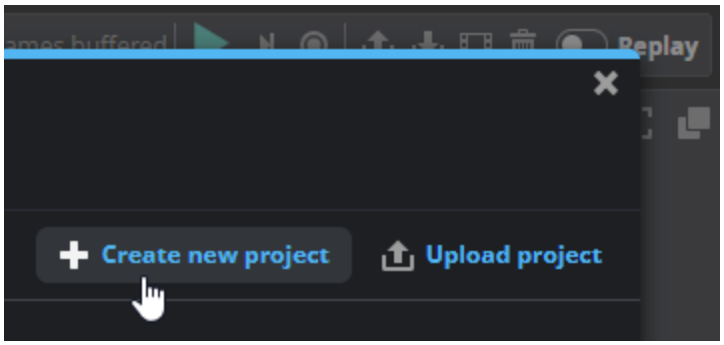


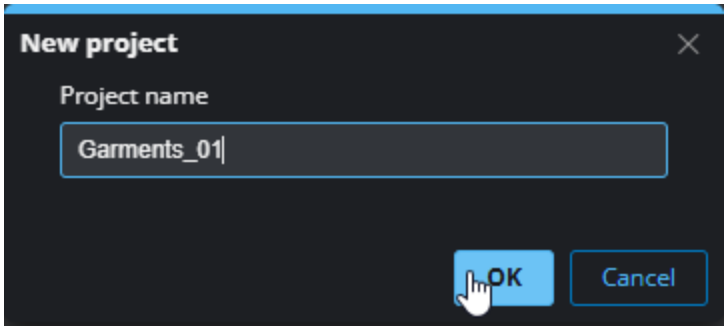
The routine starts. You will create a new project in this page, or open a previously created project.



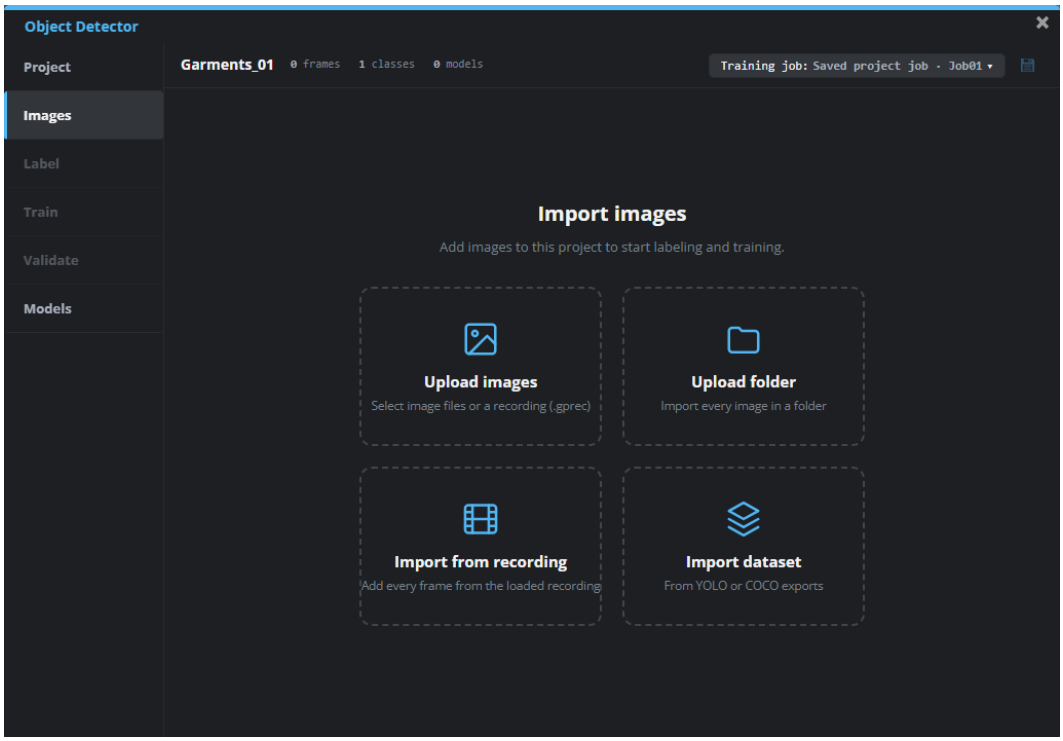
CREATE NEW OR OPEN PROJECT

1. On the **Project** tab, click **Create new project**, provide a project name and click **Continue**.





GoPxL creates a new project and switches to the **Images** tab. On this tab, you can upload images from various sources.



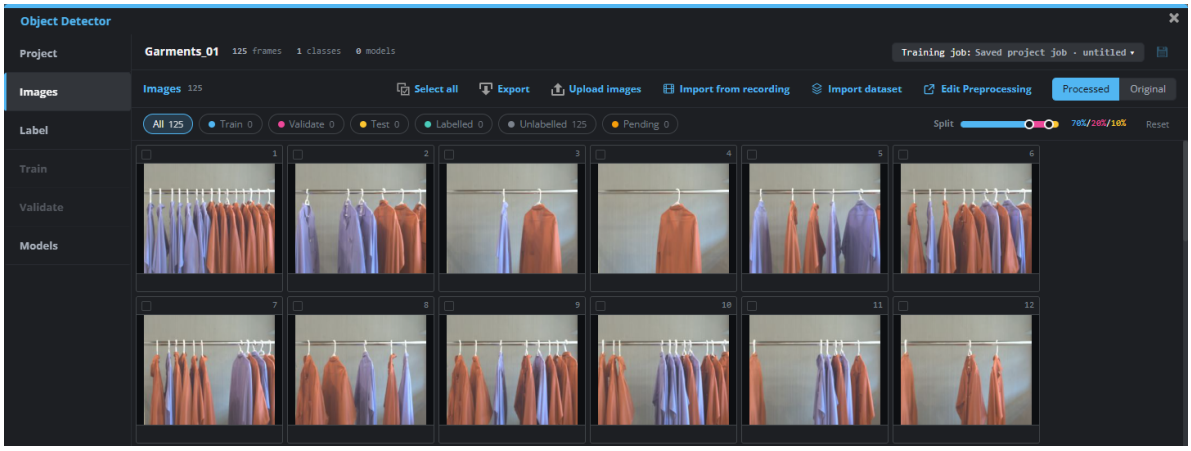
Upload images: Uploads individual image files or a recording (.gprec) from your computer.

Upload folder: Uploads an entire folder. (This feature is not yet available.)

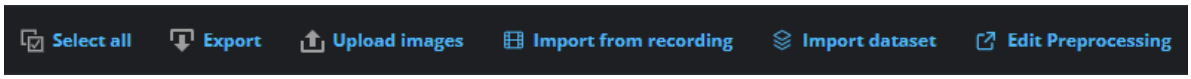
Import from recording: Imports all frames from the loaded recording.

Import dataset: Imports a labeled dataset (in a .zip file) that you exported from Label Studio, Roboflow, or Ultralytics YOLO or COCO.

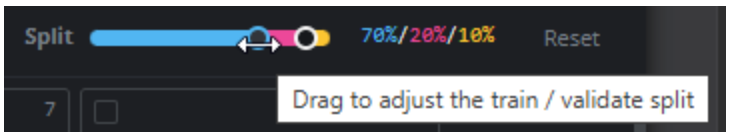
GoPxL imports the frames from the chosen source and then prepares the frames. This may take a few minutes.



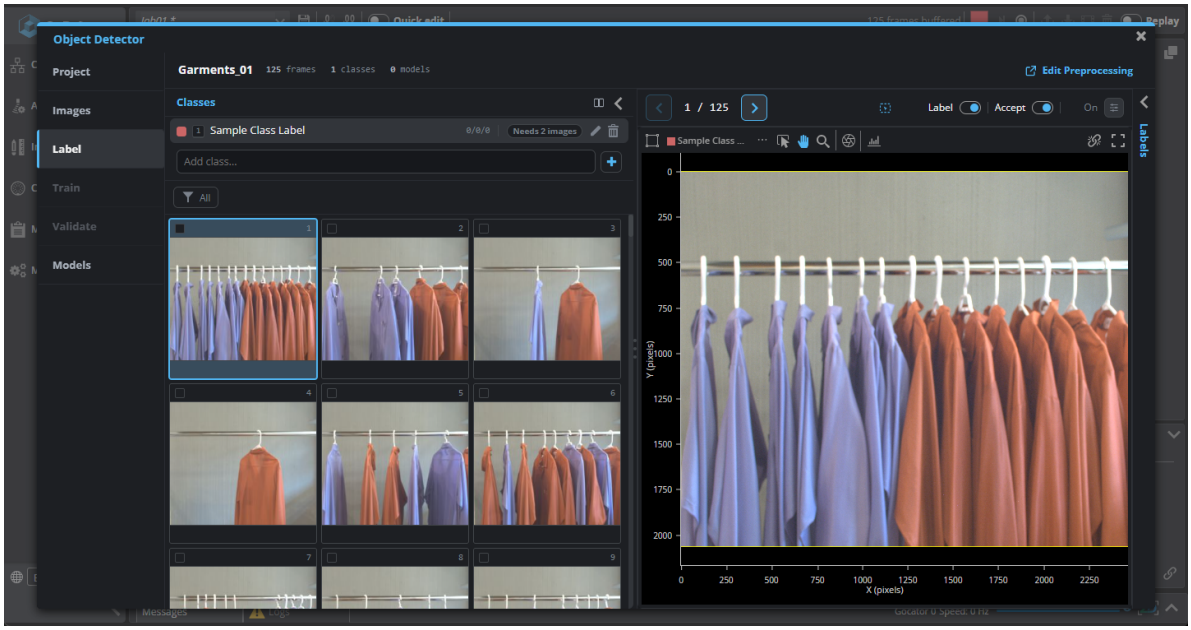
The **Images** tab lets you perform several operations on your dataset, such as selecting some or all of the frames to delete them, or uploading additional frames from the source of your choosing.



You can also adjust the proportions of images used for training, validation, and testing.

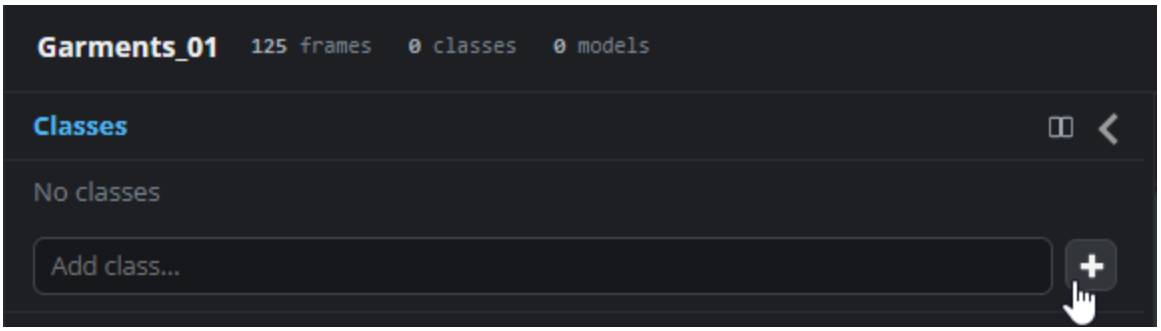


2. Switch to the **Label** tab.

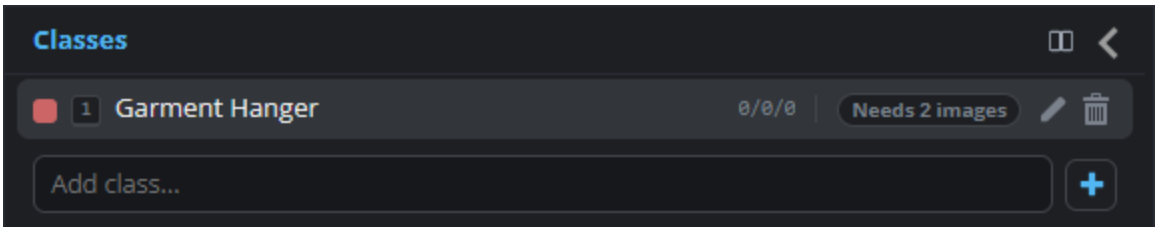


ADD CLASSES

You must now add classes to the project. For each class, type a name in the **Add class...** field and click the Add class button (+).



GoPxL adds the class under **Classes** and, to the left of the class name, displays the color GoPxL will use for the bounding box of the class, as well as the keyboard shortcut you can use to quickly add an instance in a frame. For example, in the following, the "Garment Hanger" class has a red bounding box, and its keyboard shortcut is "1". You can rename and delete classes using the icons to the right of the class.

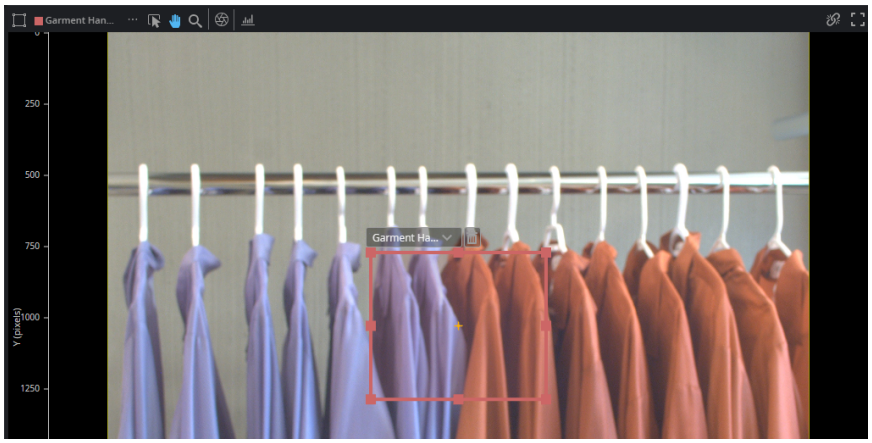


LABEL OBJECTS IN FRAMES, SET BOUNDING BOXES

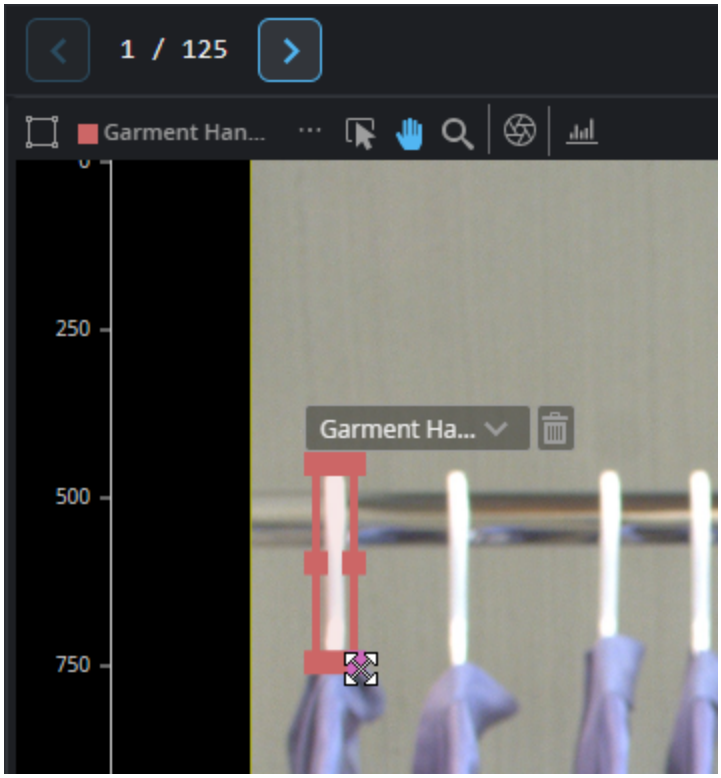
You must now assign class labels to each object in the frames and set a bounding box for each object.

1. Click a frame in the list, and do one of the following:
   - Press the shortcut key of the class.
   - Click the Create label button at the upper left of the data viewer. If you have defined more than one class, click the "..." button to switch to that class.



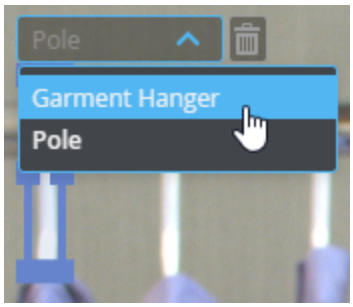


2. Position and resize the bounding box using the mouse as required.



If you make a mistake, you can delete the label by pressing the Delete key or by clicking the trashcan icon next to it.

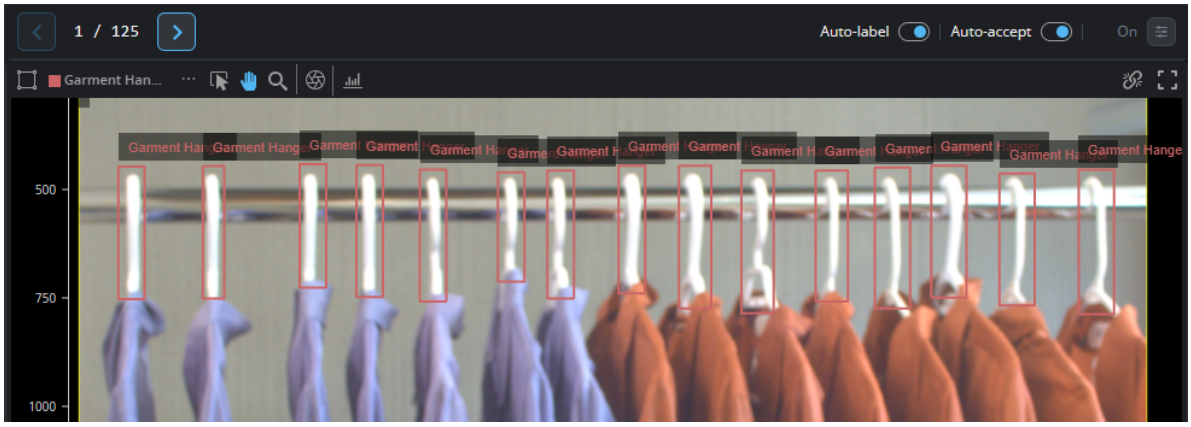
You can also change the selected label's class by clicking its class name and choosing another class from the drop-down, or by pressing the shortcut key of the other class.



3. To add another label, press the Esc key or click anywhere in the data viewer to deselect the label you just added, and use the class shortcut button or click the Create label button to add it.

   GoPxL creates a new label, using the size of previously created label's bounding box, and positioned near the location of the previously created bounding box.

4. Continue adding labels in the frame until all objects of interest are labeled.

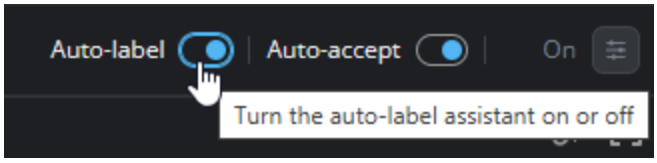


5. Click on the next frame in the frame list and label the objects in it.

   You can also move through frames using the "Q" (previous) and "W" (next) keys.

6. If you decide not to use the Auto-label function, make sure it is disabled, and continue labeling the objects in the remaining frames, and then go to *Configure Training* on page 432.

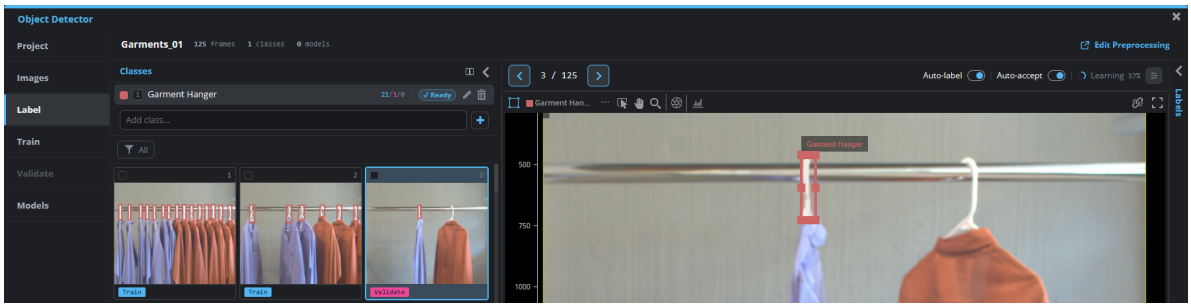
7. If you want use the tool's auto-labeling function, enable the toggle above the data viewer.



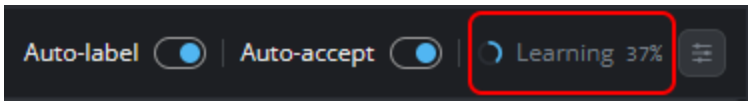
   For information on the auto-lableling and auto-accept settings, see *Auto-label and Auto-accept Settings* on page 432.

8. Click the third frame in your dataset, and create a label in it.

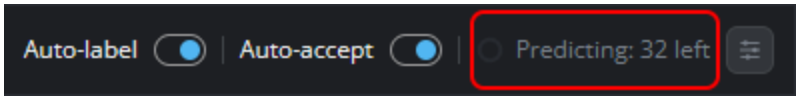
   The tool starts learning based on the labels you have created so far in the frames.



Progressing is indicated at the top right of the wizard.

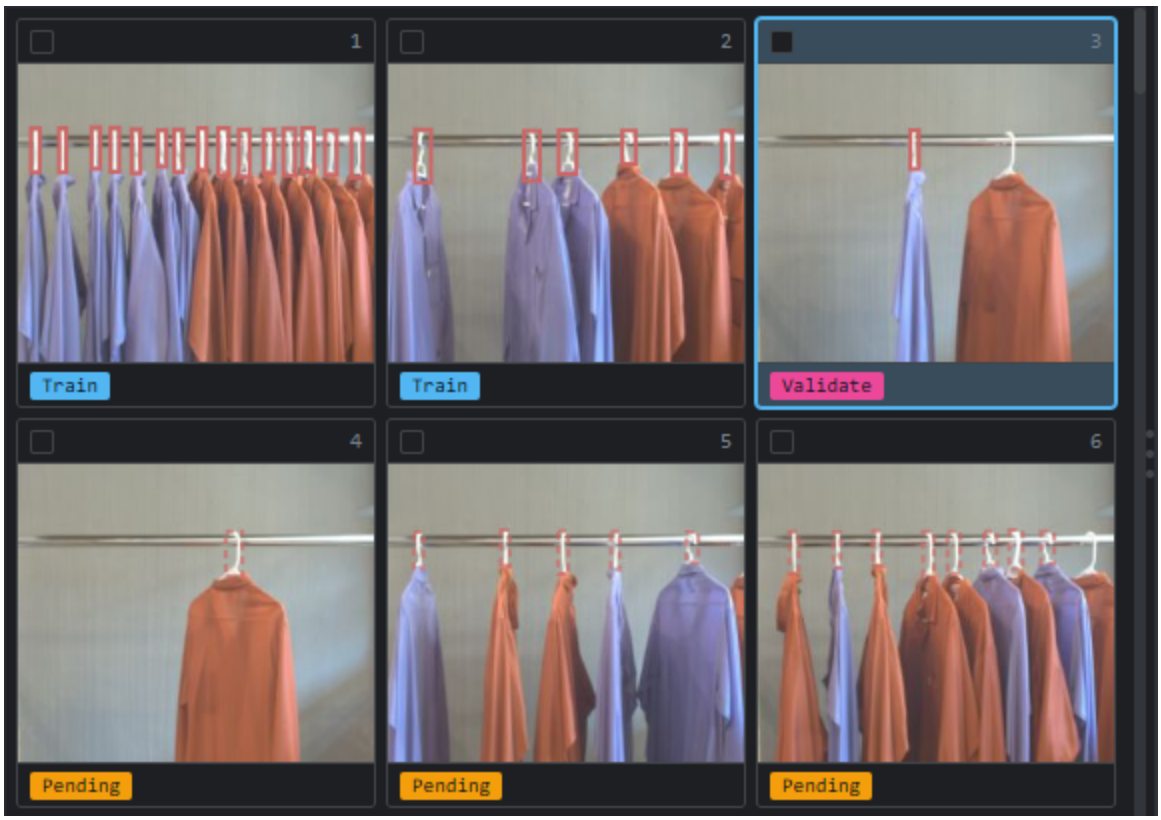


After the tool has learned from the previous labeling, it starts to predict labels for the remaining frames.

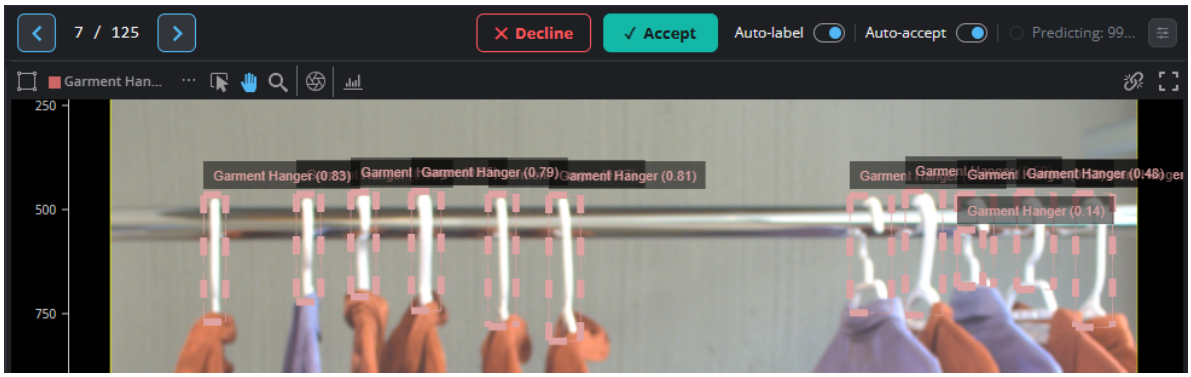


When the tool has finished predicting labels, it displays a notification at the top right saying it is waiting for more labels.

Frames in which the tool has predicted labeled are indicated with "Pending" in the frame list.



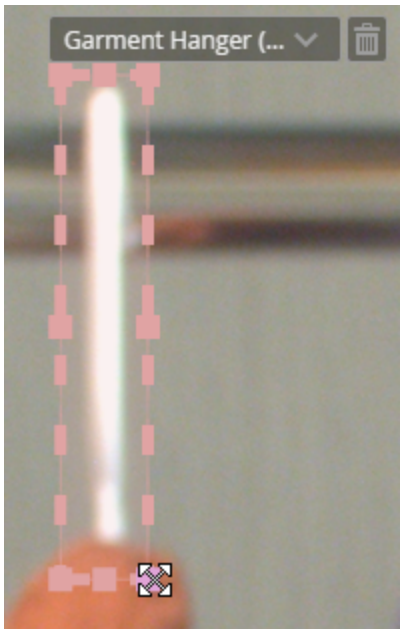
9. Click one of the "Pending" frames.



The tool displays the bounding boxes with "broken line" overlays.

10. Evaluate the size, shape, placement, and class of each predicted label.
11. To modify a label, click it in the data viewer to select it.

    Make sure to switch to "pan" mode ( ).
12. If necessary, do any of the following:



- Re-size or re-position the selected label bounding box.
- Change the class name via the drop-down or the keyboard shortcut of the class.
- Delete the prediction by clicking the trashcan icon.

13. If you are satisfied with the labels in the frame, click **Accept** above the data viewer.

    The tools displays the next frame containing predictions.

    As you progress through the frames, the tool will learn based on any changes you make in the frames (accepting, changing, or deleting predictions).
14. If you are not satisfied with the labels in the frame, click **Decline** to reject all of the predictions in the frame and go to the next.

15. After you have labeled all the objects in your dataset, click the **Train** tab to configure the training parameters.

    For information on configuring training, see *Configure Training* below.

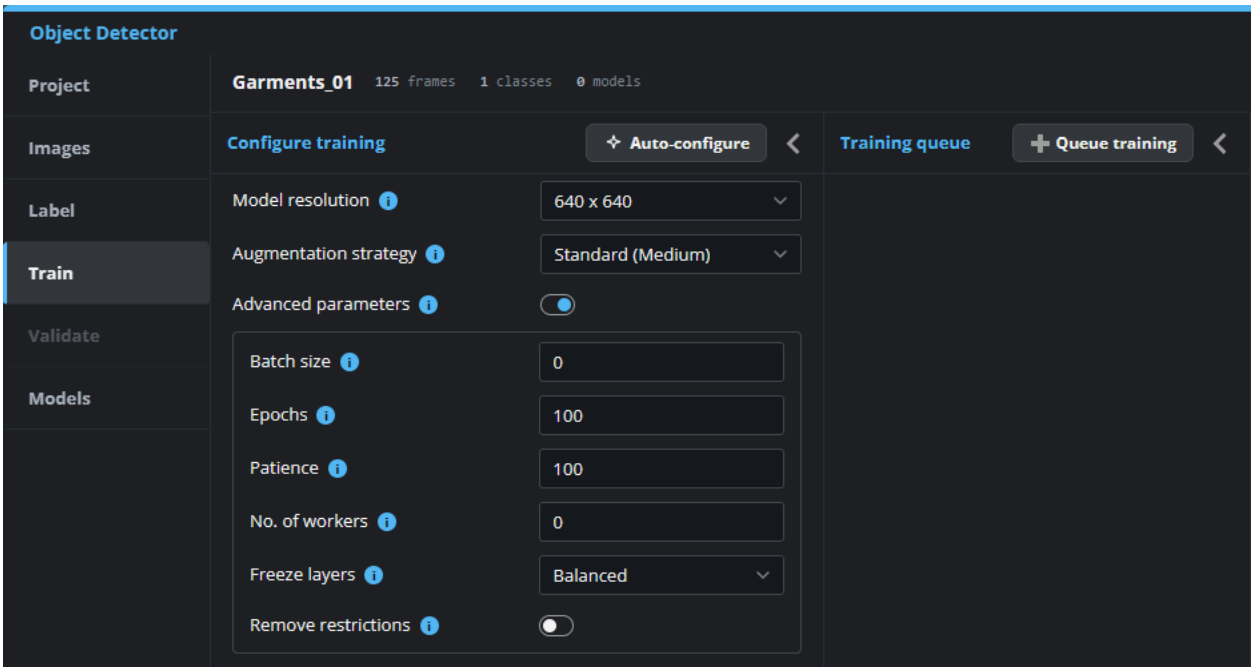
AUTO-LABEL AND AUTO-ACCEPT SETTINGS

TODO

CONFIGURE TRAINING

After labeling the project, you must configure training.

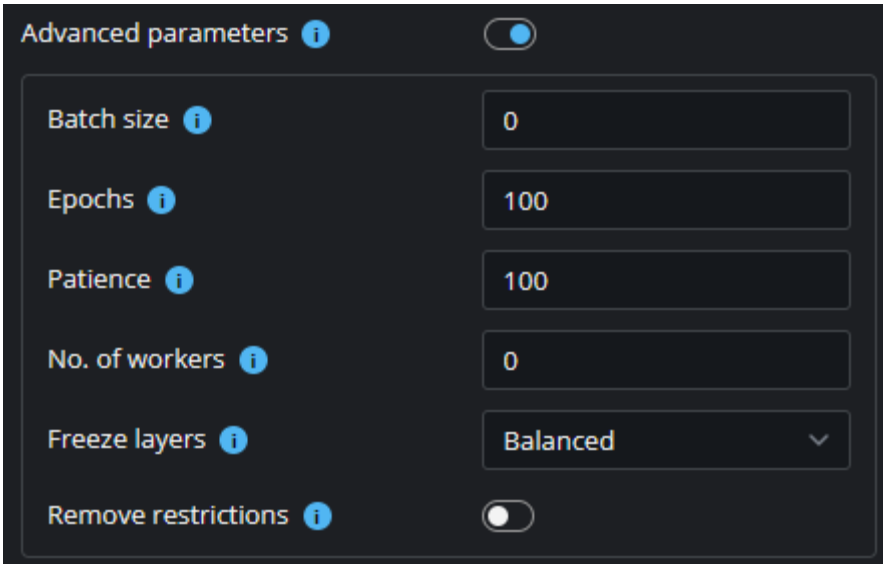
You can let the tool automatically configure the settings based on the size and number of labeled objects in your dataset by clicking **Auto-configure**. You can always change the settings later.



Training Parameters

Parameter	Description
Model resolution	Decimates the input data (if necessary) to the specified resolution for processing. A higher resolution improves accuracy but increases processing time for training.For most applications, leave this at 640x640.
Augmentation strategy	Choose one of the following based on the expected conditions of your application. **Precision**: Uses minimal transforms for controlled environments with fixed lighting and camera positions. **Standard**: Adds moderate variations to account for typical fluctuations in factory lighting and part orientation. **Robust** Applies aggressive image distortion to help the model identify defects in

Parameter	Description
	high-glare or cluttered scenes.
Advanced parameters	If the training results are generally good but the tool is reporting a small number of false positives or false negatives, you can modify these parameters to further improve results.



Batch size

The number of frames in a batch (a subset of the entire dataset passed to the training algorithm for an iteration).

If you have memory issues when training, decrease this value. With a value of 0, the tool determines the batch size automatically.

Epochs

The number of times the entire dataset passes through the training algorithm. To reduce false positives or negatives, increase this value. Note that training will take longer.

Patience

The number of epochs to wait without improvement in validation metrics before stopping the training early to prevent overfitting. As a guideline, start with a value around 10-20% of total epochs. The value must be less than or equal to the value in **Epochs**.

Learning rate

Parameter	Description
	Set this to a higher value to reduce training time. Note that a value that is too high can cause the model to miss the point where the difference between predicted and actual class is at its smallest (that is, overshooting it). A lower value causes the model to learn more slowly, but potentially more accurately; if the value is too low, the model can get stuck. **No. of workers** The number of worker threads for loading data. Increasing the number of workers can improve data throughput and help keep the GPU fed, which may speed up training. However, it also increases CPU load and RAM usage, and can cause instability if set too high. **Freeze layers** One of the following: **Full retrain**: Freezes 0 layers to update the entire model for maximum precision and is best for datasets with over 2,000 images. **Balanced**: Freezes most layers to keep foundational knowledge while learning your specific items. **Quick adapt** Freezes all layers to prevent overfitting on small datasets of fewer than 200 images. **Remove restrictions** When this setting is enabled, you can toggle **Automatic mixed precision** to speed up training and lower memory usage. By default, this option is disabled, and the following limitations are in place to ensure the system trains successfully and does not run out of memory: • The maximum number of tiles under the **Tiling** parameter is limited to 6. • The highest resolution in **Model resolution** is 512x512. • **Clamp** is limited to TBD. • Learning rate is limited to TBD.

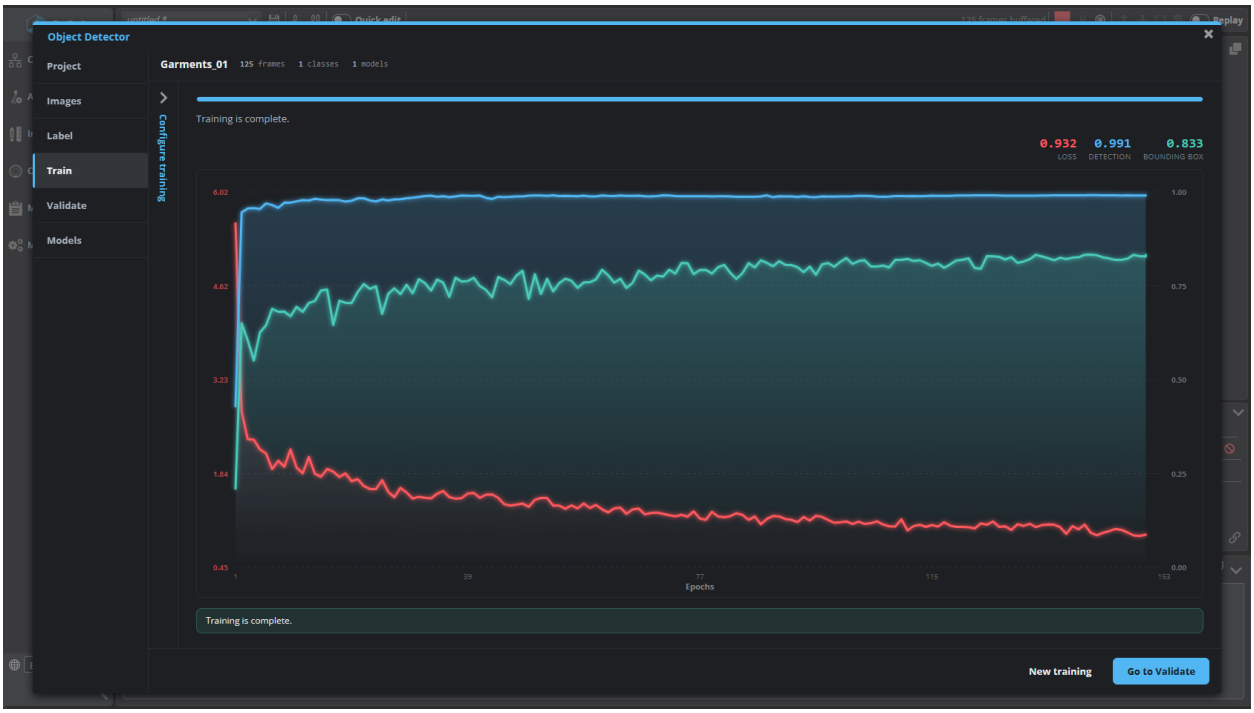
After you configure training, click **Start Training** to train the model. The tool will start by preparing the training data, providing progress feedback:

After the data is prepared, the tool will start training and generating graphs of metrics to chart the progress.

Loss: This metric represents the cumulative error for each frame in the Test set, determined after each training run iteration or epoch. To put it another way, it's the difference between the model's current prediction and the actual label you provided (sometimes called the "ground truth"). A decreasing number (the graph's line is moving down) means the model is improving.

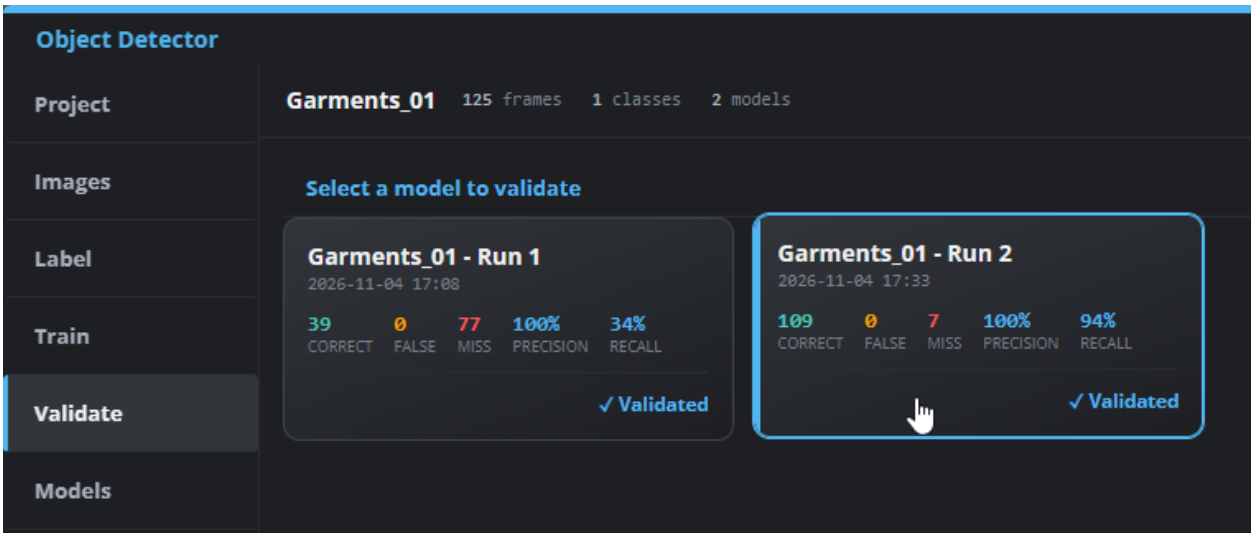
Detection: This metric shows how well the model identifies an object in a frame as belonging to one of the provided classes, and whether it is detected at all. A rising value means the model is learning to pick out the object from the background, and how "confident" it is of this differentiation.

Bounding Box: This metric shows how well predicted boxes overlap with boxes you provided during labeling, in terms of position and size. A value lower than the Detection metric is normal; a value of around 0.7 is considered "good."



VALIDATE THE MODEL

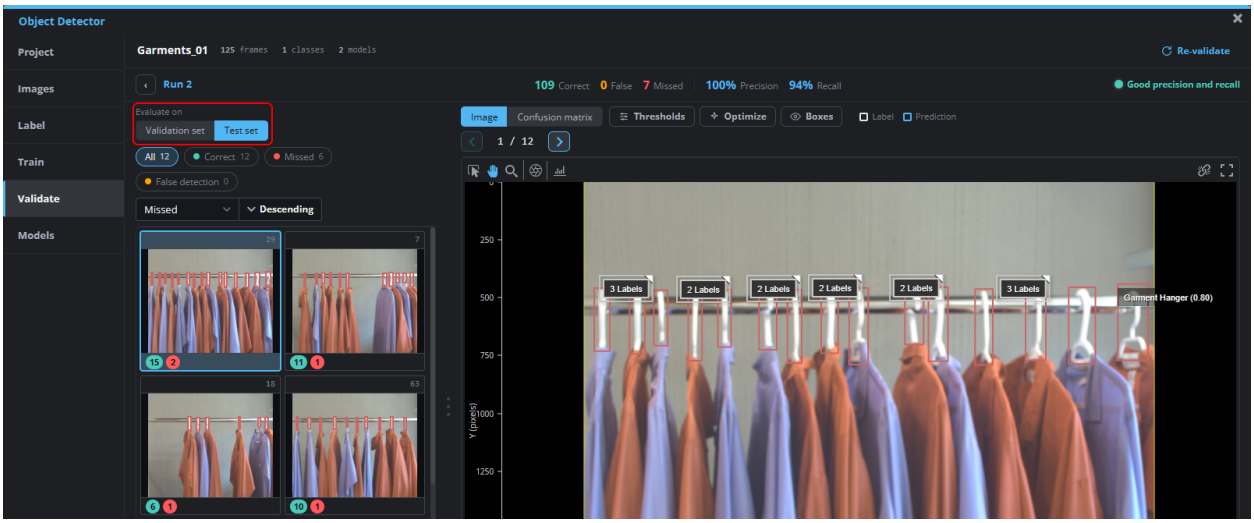
After training is complete, you can evaluate the model's performance on the **Validate** tab. Click a model to see how it performs.



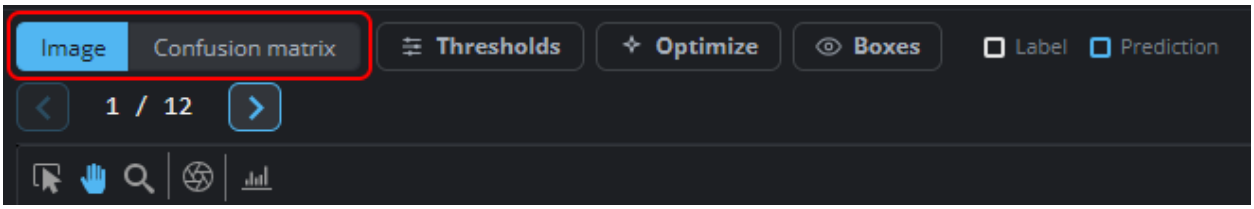
The tool displays two sets of frames you can evaluate: a Validation set, and a Test set.

Validation set - A subset of the frames used during the Validation phase of the training that helped tune the model during training. The scores may be optimistic.

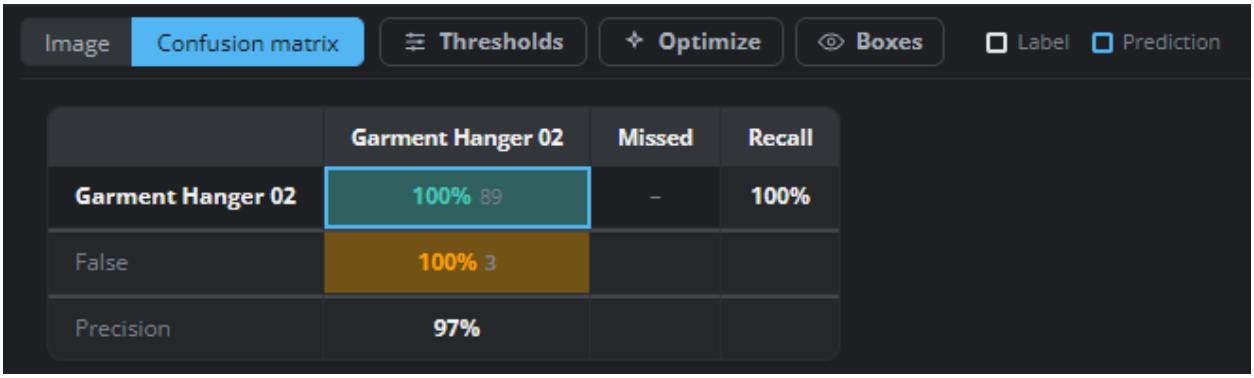
Test set - A set of frames that were intentionally excluded from training (the model has never seen these frames). The scores are more realistic.



Toggle between **Image** display and the **Confusion matrix** using the buttons above the data viewer.

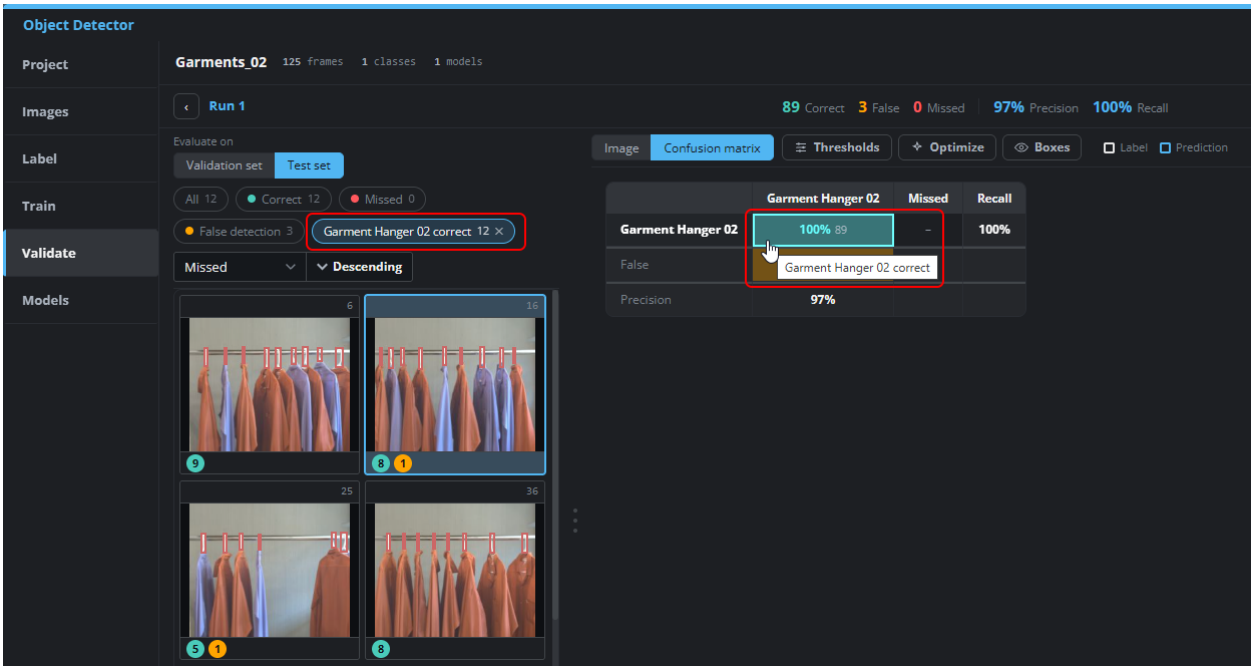


The Confusion matrix lets you see the model's performance details.



The first two columns and rows in the confusion matrix communicate how well the trained model predicted the classes of the objects in the training data.

Clicking the colored cells in the Confusion matrix table displays thumbnails of frames related to the result on the left:



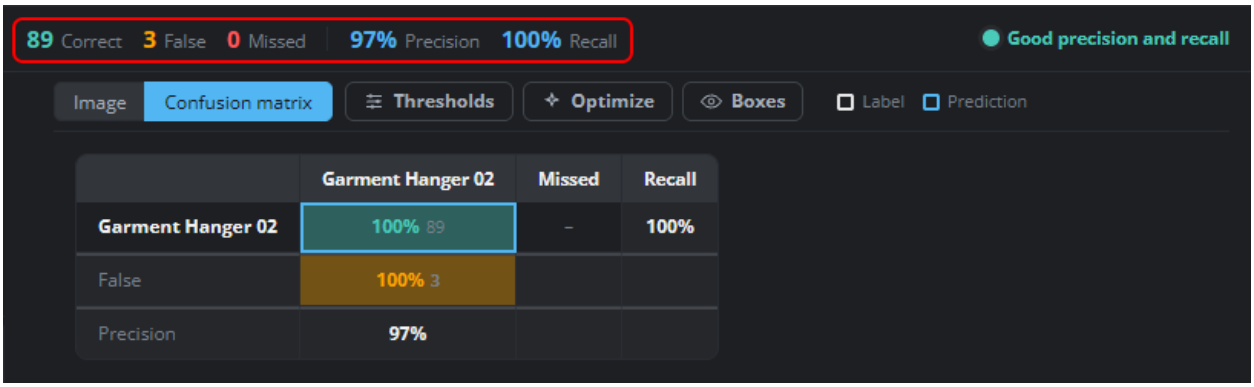
The following metrics help you decide whether the model is trained well enough for your application:

Missed - When an object of a given class is present in an image but the model doesn't detect it there.

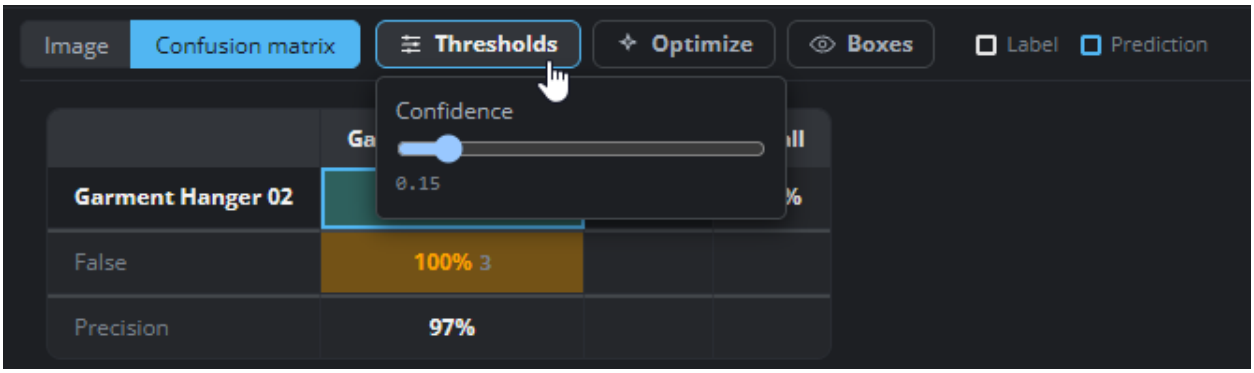
False - When the model incorrectly detects an object of a given class in an image.

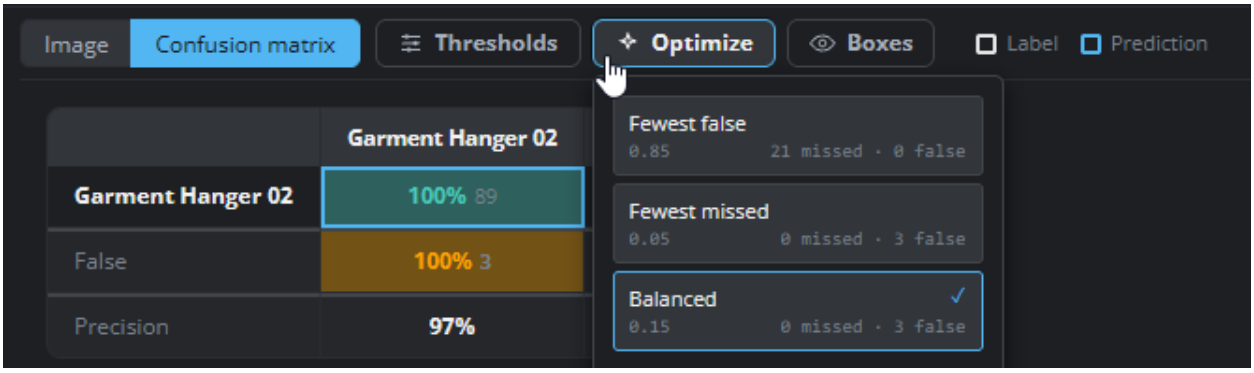
Recall - The percentage of the detected objects out of all the objects; this is a good measure of the model's sensitivity for detecting instances of a class.

Precision - How well the model's positive prediction works.

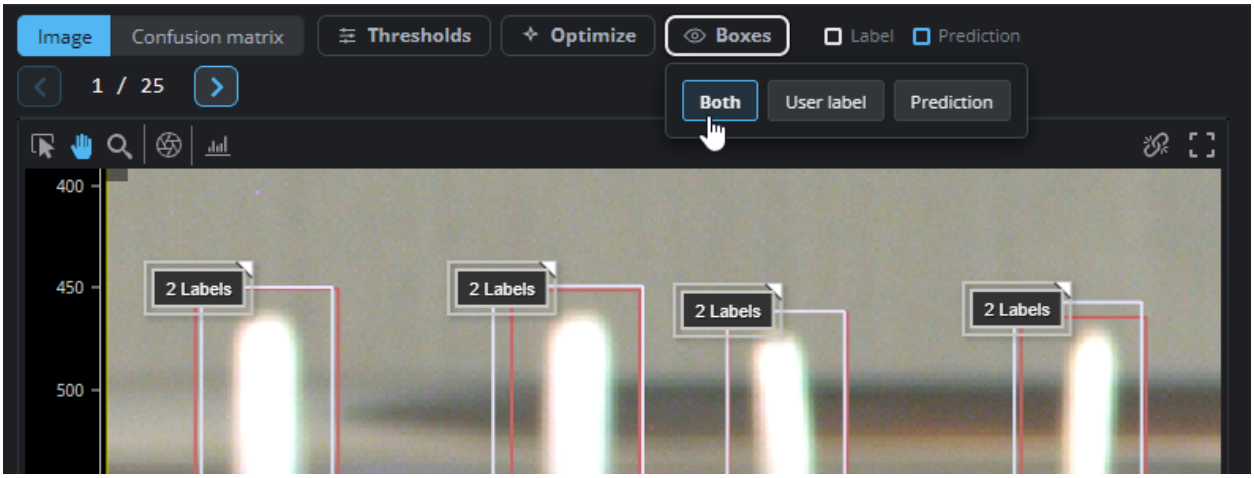


By adjusting the settings at the top of the **Validate** tab, you can evaluate the impact of different confidence score thresholds, or automatically optimize the model for different detection scenarios.



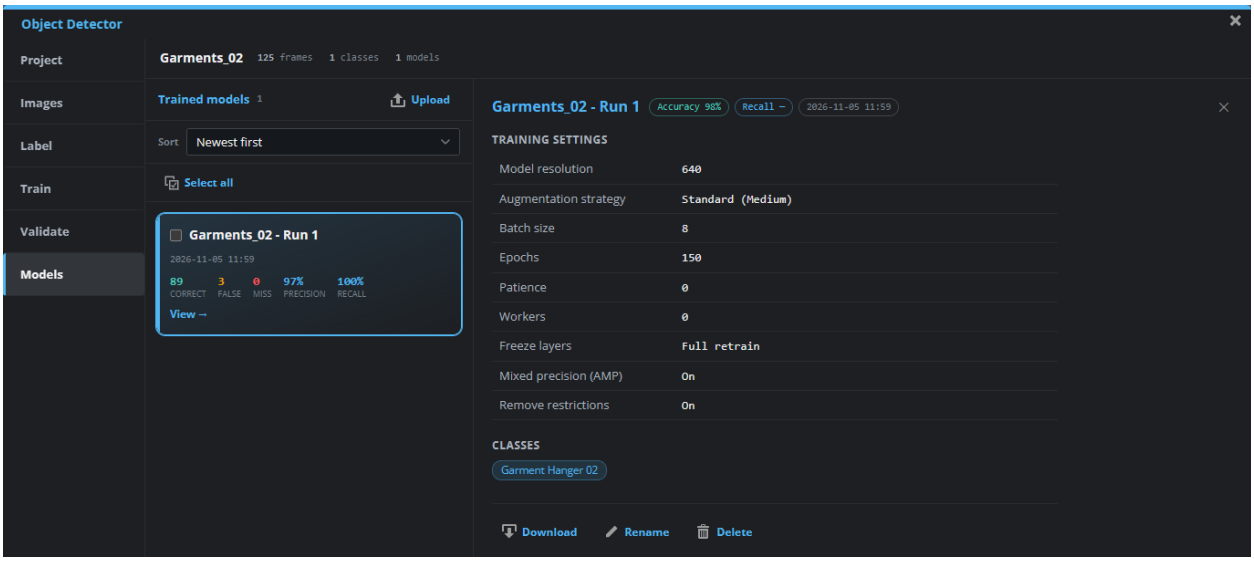


In Image mode, from the **Boxes** drop-down, you can choose whether to display the bounding boxes you drew before training, the bounding boxes of the model's predictions, or both. The boxes you drew are displayed in white, whereas the boxes of the model's prediction are displayed in the color associated with that class (shown in the boxes legend to the right of the drop-down).



MODELS TAB

The **Models** tab lets you see at a glance the model's training settings for a given run, the classes it contains, as well its accuracy and recall values.

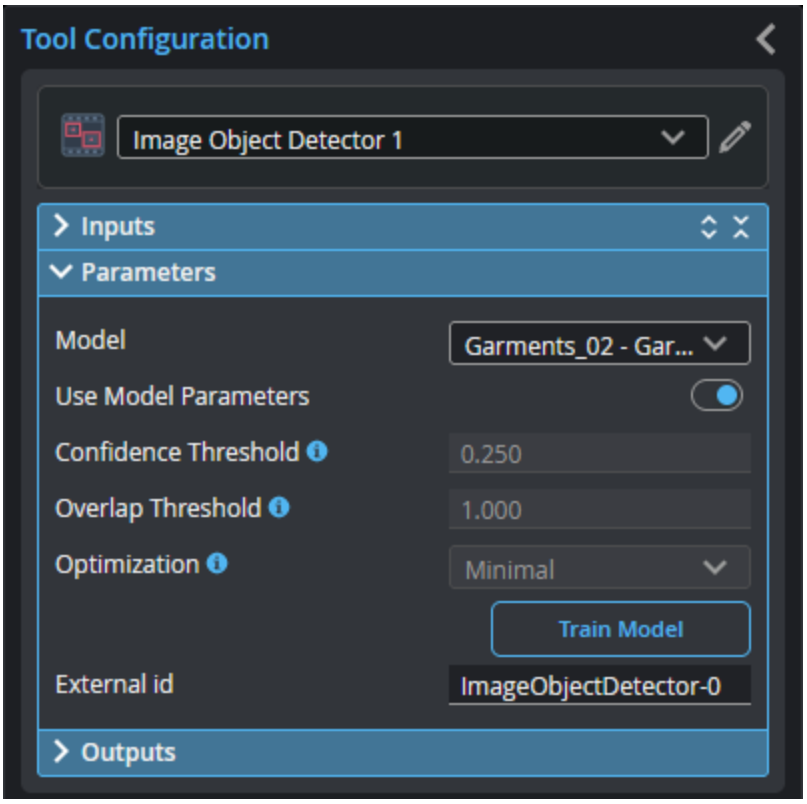


You can also manage your models here, renaming or deleting them, or download or upload models.

Parameters

In the **Parameters** section of the tool, you can launch the Train Model wizard, in which you can create a project, as well as the models, classes, and frames that the project includes. After you have created the project, the Train Model wizard lets you train the model on the frames. Once training is complete, you can use the project as an inference engine to classify new frames of data, that is, on the production line.

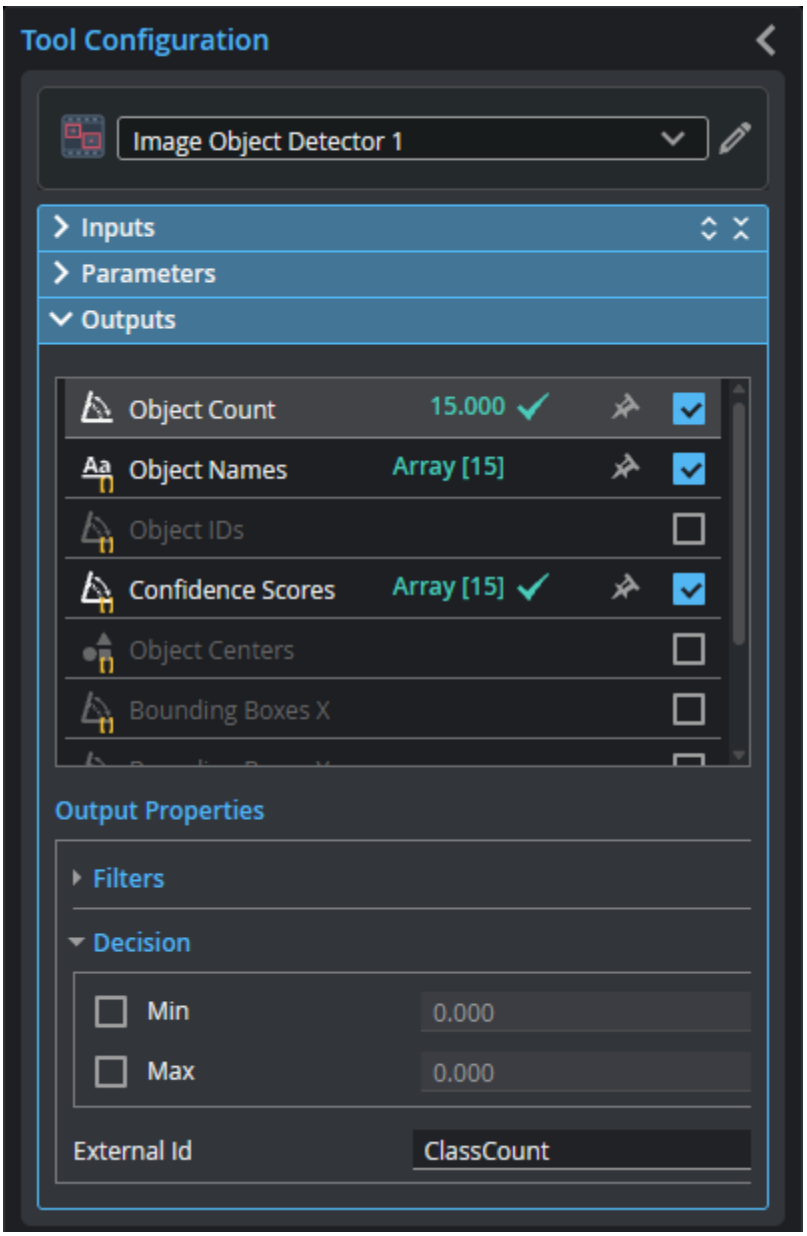
The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

Parameter	Description
Model	The trained model the tool uses to classify images during production.
Use Model Parameters	When this parameter is enabled, GoPxL uses the values set on the Model Performance page. Otherwise, you can override those values and provide different ones here.
Confidence Threshold	The confidence score value below which a detection is excluded from the results.
Overlap Threshold	The amount of overlap allowed between detected objects for them to be included in the results.
Optimization	The level of optimization applied to the model for improved runtime performance. One of the following: Minimal, Balanced, High. Higher optimization results in longer training time, but faster inference (that is, object detection).
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Note that some measurements return IDs, whereas others return class names. The IDs correspond to the class names added in the project. So for example, given the classes "lays classic", "cheetos", "doritos", and "ruffles", the corresponding IDs are 0, 1, 2, and 3. The ID values are ordered based on the order in which the classes were added in the project.

Measurements

Measurement
Object Count The number of objects detected in the frame.
Object Names **Object IDs** The classes and IDs of the objects detected in the frame.
Confidence Scores The score associated with the detection.
Bounding Boxes X **Bounding Boxes Y** **Bounding Boxes Width** **Bounding Boxes Height** An array of the positions and sizes of the bounding box encapsulating a detected object.

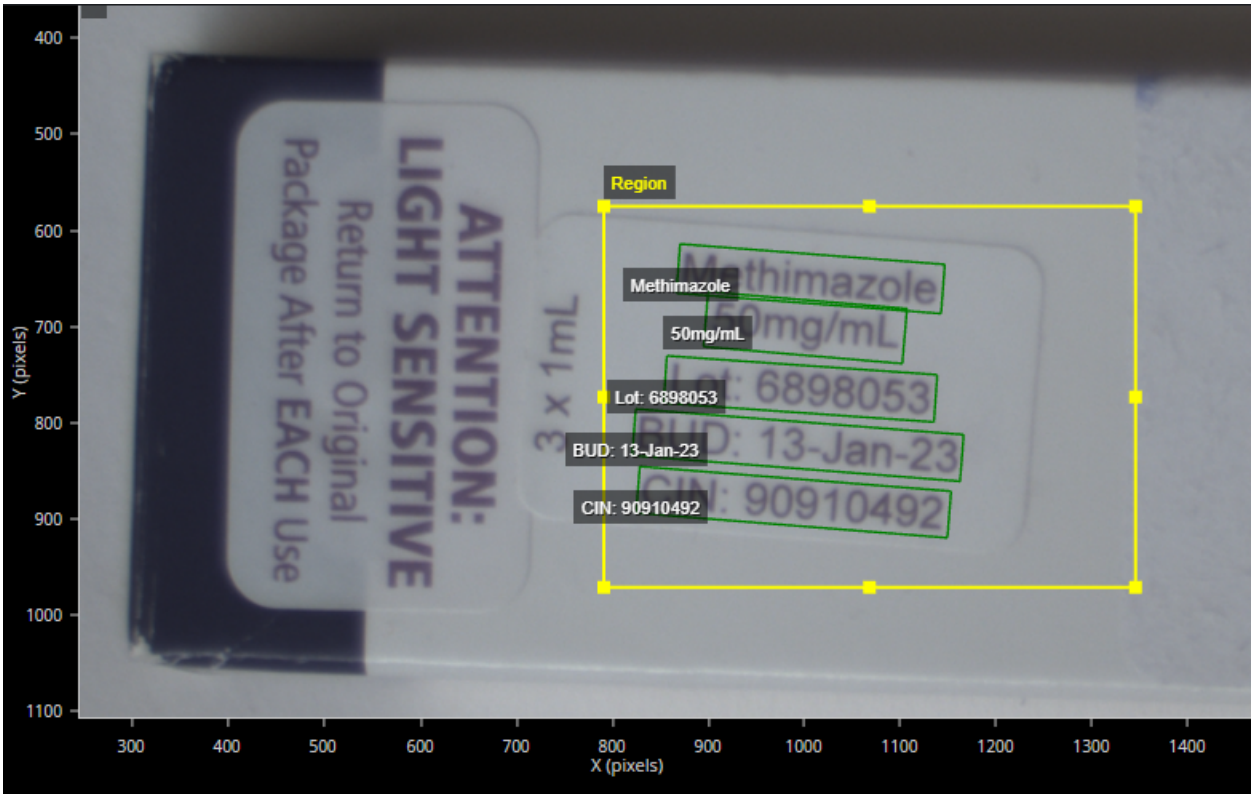
Features

Type	Description
Object Centers	An array of Point geometric feature representing the center of the bounding box of the detected objects.

Image OCR

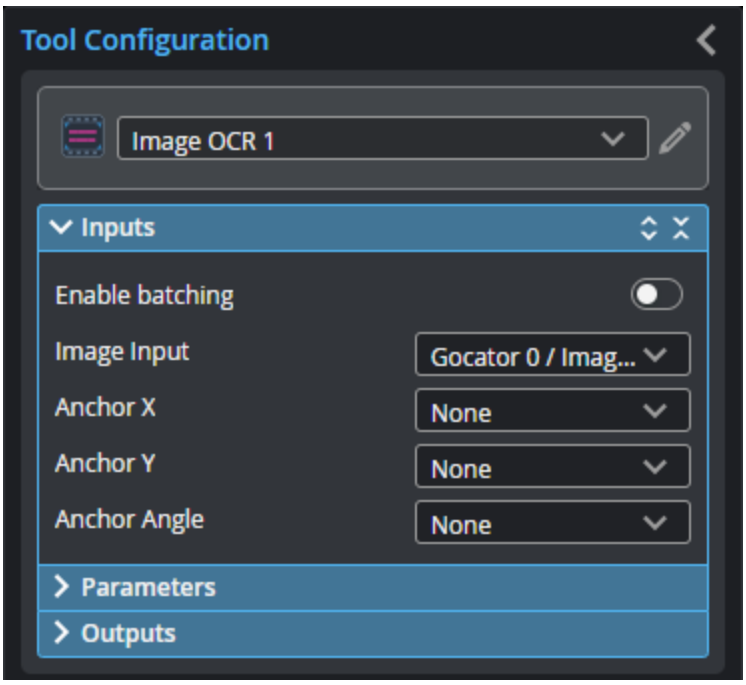
The Image OCR tool lets you perform optical character recognition to extract strings of text from Image data. The tool lets you optionally set the number of expected lines, as well as the expected structure and composition of the text (number and types of characters, and so on) to improve recognition. The tool is font-independent and already trained. Text can have any orientation in the image.

Lines of text (horizontal or vertical) currently can have a maximum 1:20 aspect ratio.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs



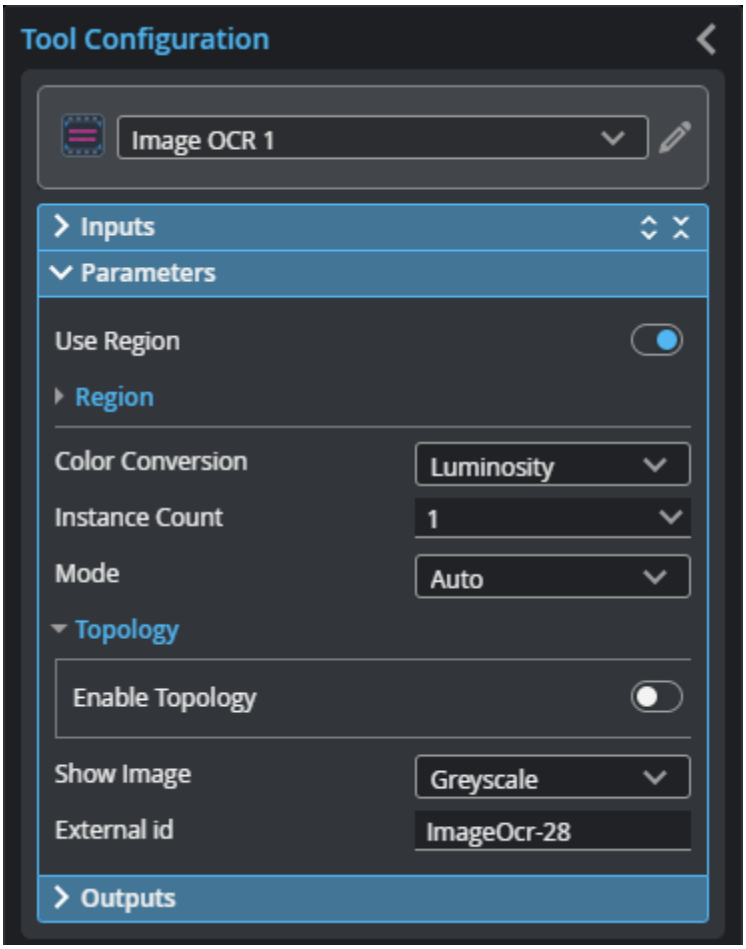
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

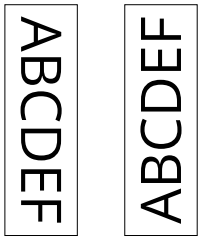
The following parameters are in the expandable **Parameters** section in the tool's configuration.

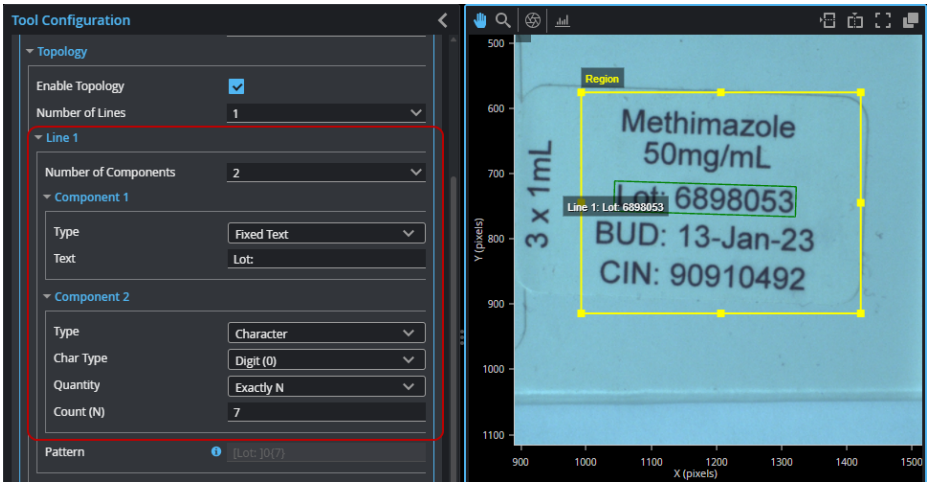


Parameters

Parameter	Description
Use Region Region	When **Use Region** is enabled, configure the region in the **Region** section.
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value: `pixel = 0.299R + 0.587G + 0.114B` When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB

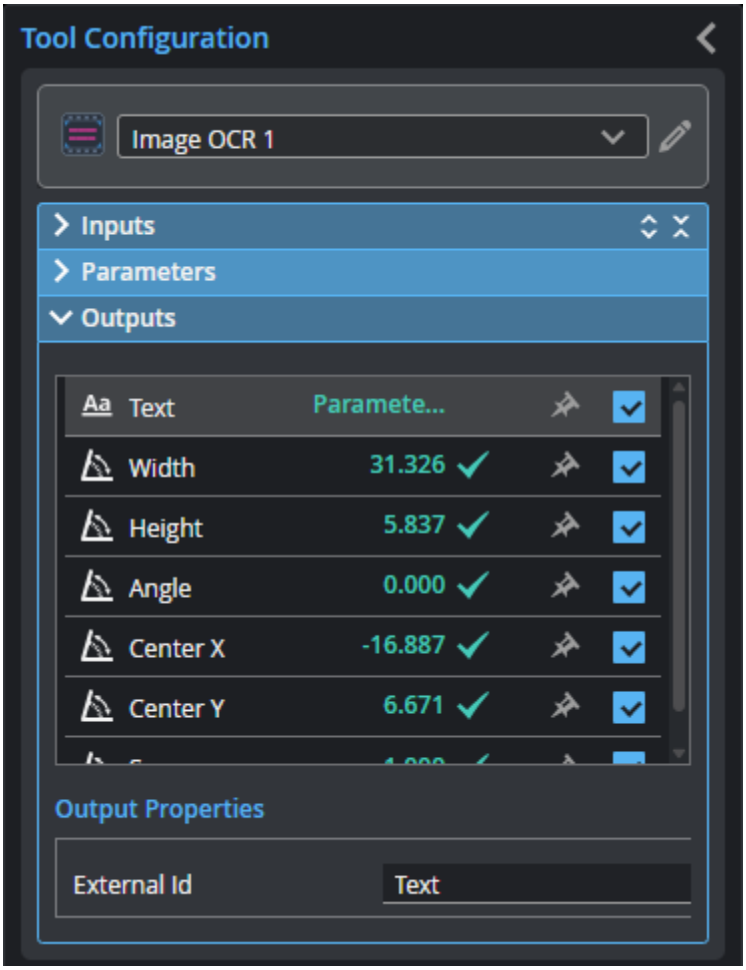
Parameter	Description
	pixels relative to this color.
Instance Count	The expected number of occurrences of recognized text to be returned by the tool. If multiple instances are found, these are returned as an array.
Mode	The tool's mode. One of the following: **Auto** The tool detects text in image before performing optical character recognition. More flexible but slower. The default mode. **Read Single Line** Reads a single line of text without running text detection. This mode can be up to roughly two times faster than Auto mode. To use this mode, you must enable a region, and size and position it so that it is close to the text. Use this mode when: • The tool's region of interest (ROI) can be sized tightly around the text. • The position of the text is consistent between frames (either due to consistent part position or anchoring). The text must run left to right. For example, only the examples on the left would be read; (b) and (c) can only be read if you either rotate the region or specify the text's orientation in **Text Direction** (after disabling **Auto Text Direction**). For best performance, disable **Auto Text Direction** and specify the orientation of the text in **Text Direction**.

Parameter	Description
	Left to right ✓ (all orientations) a.  b.  c.  Top down ✗ (no supported orientations) 
Auto Text Direction Text Direction	When **Auto Text Direction** is disabled, set **Text Direction** to the expected angle of the text. When the setting is enabled, the tool automatically determines the angle of the text; note however that when this parameter is enabled that tool performance is reduced.
Topology Enable Topology	Expand **Topology** and check **Enable Topology** to configure a topology. A topology lets you specify the expected structure and composition of the text, that is, the number of lines, and for each line, the expected types and number of characters, as well as any fixed text the line will contain. Configuring a topology can improve the reliability of the tool's optical character recognition and output. **Number of Lines** The expected number of lines in the instance. For each expandable **Line {n}** section, you set the number of components (**Number of Components**), and then the composition of the components. **Type** One of the following: Fixed Text or Character. When set to Fixed Text, you provide some expected text. Text that is expected in each frame or instance (such as "Lot:" or "Exp. date:") can help improve character recognition.

Parameter	Description
	When you set **Type** to Character, you provide the character type (**Char Type**) and the number of characters (an exact number, or a range). For example, the following topology settings capture a line starting with the text "Lot: ", followed by exactly 7 digits.  The character types correspond to the following: • A: an uppercase ASCII letter • a: a lowercase ASCII letter • 0: a digit • P: an ASCII punctuation character • V: a character belonging to either A or 0 • X: a character belonging to either A, a, 0 or P • _: a space • S: a character belonging to either X or a space **Pattern** This represents the search string.
Show Image	Determines how the image (or the image data in a region) is displayed in the data viewer. One of the following: **Original** Shows the original image data. In color sensors, this will be the color image. **Greyscale**

Parameter	Description
	Shows the image data converted to greyscale. The conversion is determined by the option you choose in the tool's **Color Conversion** parameter.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

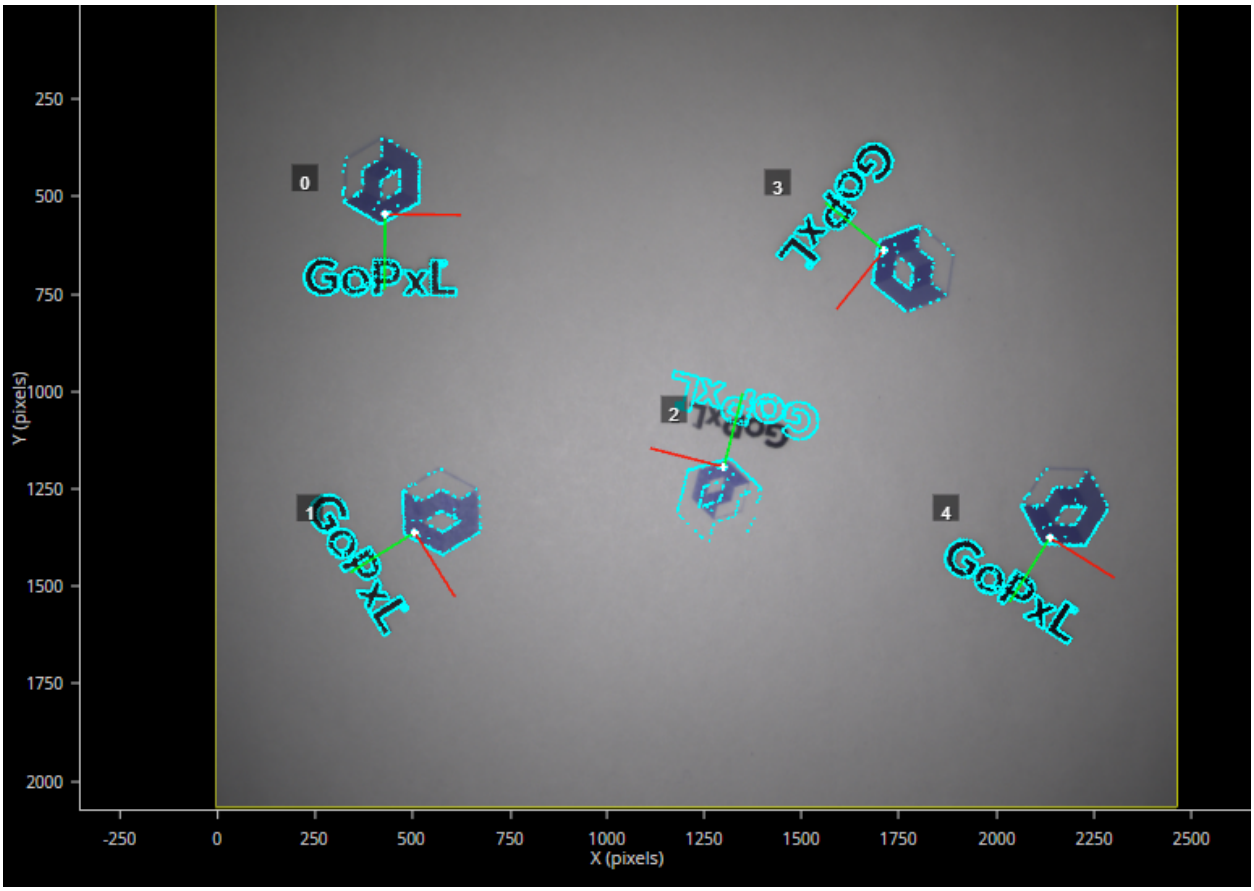
Measurements

Measurement
Text The recognized text.
Width **Height** **Angle** The width, height, and angle of the bounding box containing the text.
Center X **Center Y** The X and Y position of the bounding box containing the text.
Score The confidence score of the character recognition. Use this to evaluate how trustworthy detection is.

Image Pattern

The Image Pattern tool locates one or more instances of a feature or part that matches a reference pattern you define. You can define multiple patterns, and you can edit and delete existing patterns. You can also download patterns to back them up or transfer them to another device, and upload them to a device.

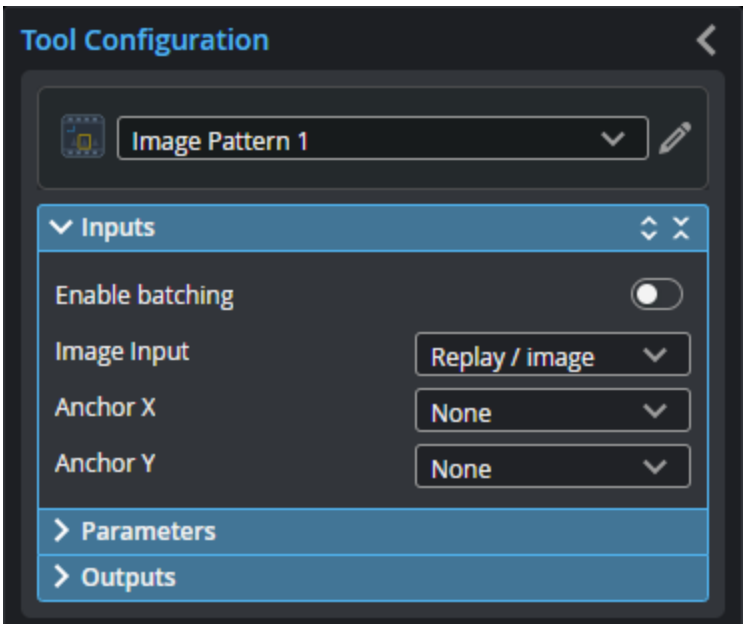
In the following, the tool finds five instances of a pattern. Note that the pattern displayed in the data viewer is not scaled up or down to correspond to the size of an instance, such as the middle pattern below.



The tool provides two algorithms:

- "Accurate": Ideal for complex patterns, as its detection is more reliable.
- "Fast": Faster, but less accurate.

Inputs



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

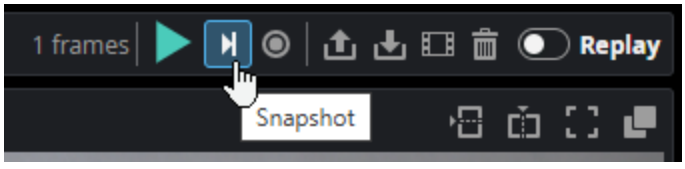
Inputs

Name	Description
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Creating a Pattern

To create a pattern, do the following:

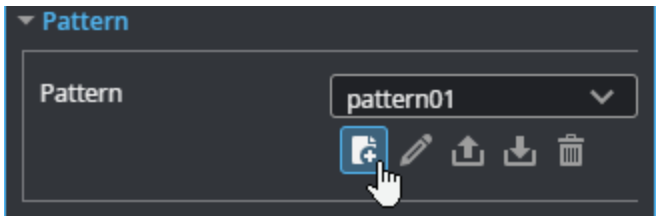
1. Make sure you have calibrated the camera and configured its acquisition settings on the **Acquire** page.
   For more information, see *Acquire - Alignment and Acquisition* on page 122.
2. Place a representative, "good " target under the camera.
3. Acquire an image by pressing the snapshot button.



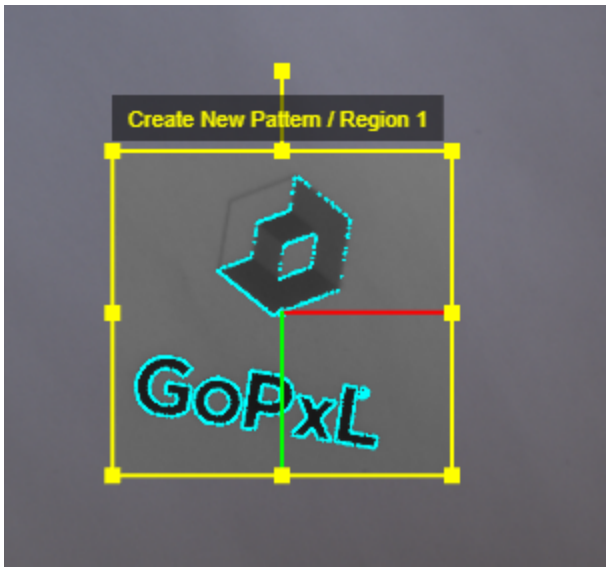
4. Add an Image Pattern tool.

The first time you add the tool in GoPxL, because there are no patterns on the device, the tool automatically displays the **Create New Pattern** settings. If this is the case, go to step 7.

5. If you have previously created a pattern, expand the **Pattern** section and click the Create button.

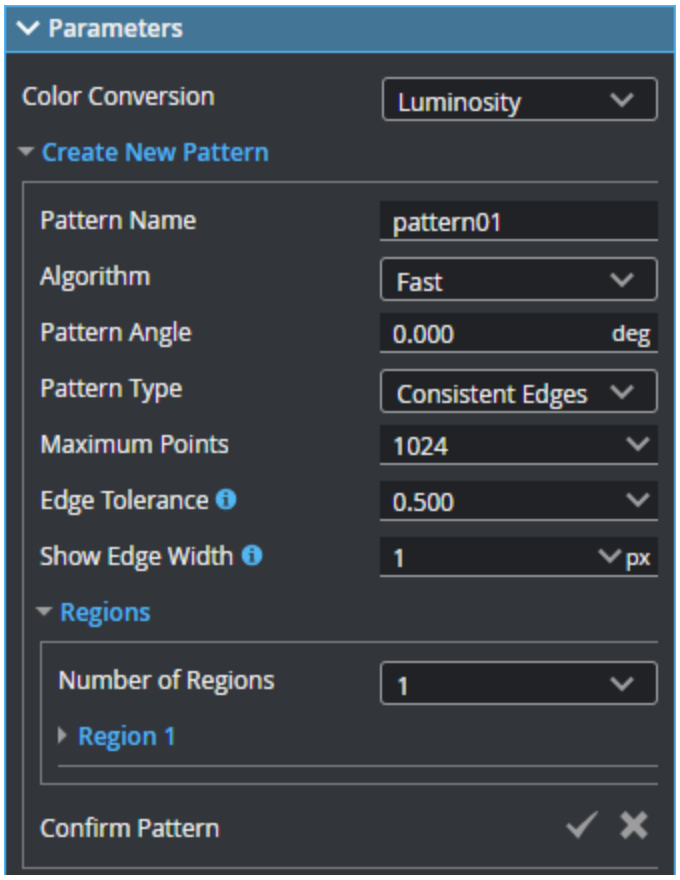


6. Move and resize the region of interest (ROI) so that it encompasses the feature or part on your target.



Limit the size of the ROI to improve matching with the pattern later on.

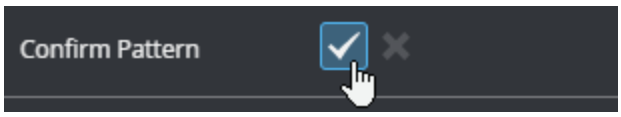
7. In the **Create New Pattern** section, provide a name in **Pattern Name**.



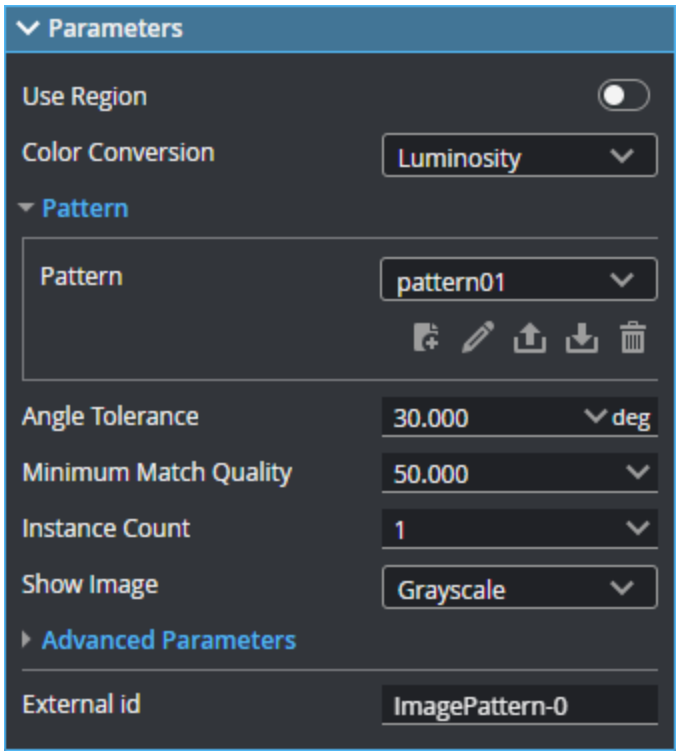
8. Modify the settings for the pattern and your application.

   For information on the settings, see *Pattern Parameters* on page 456.

9. Click the check button next to **Confirm Pattern** at the bottom of the **Create New Pattern** section.



   The pattern is added to the list of patterns available to Image Pattern tools. The tool hides the pattern parameters and displays the tool's matching parameters.

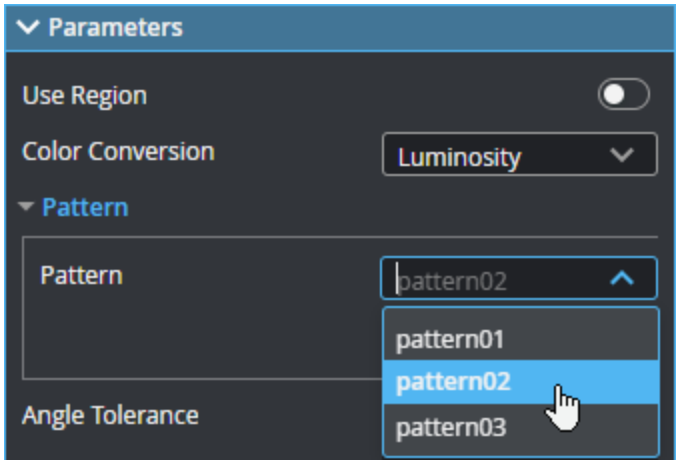


For information on the matching parameters, see *Tool (Matching) Parameters* on page 459.

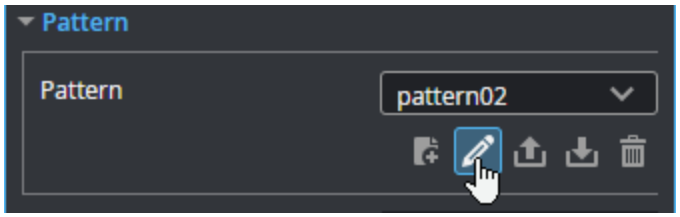
Editing a Pattern

To edit a pattern, do the following:

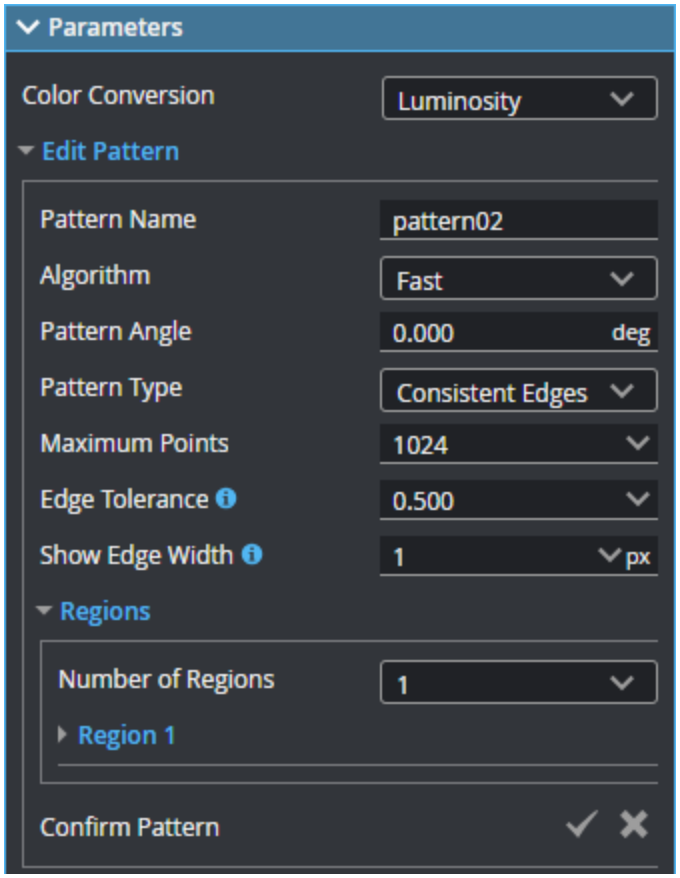
1. In the tool's **Parameters** section, expand the Pattern section, and choose a pattern.



2. Click the Edit button.



The tool displays the pattern parameters.



3. Modify the pattern parameters as needed.

   For information on the pattern parameters, see *Pattern Parameters* below.

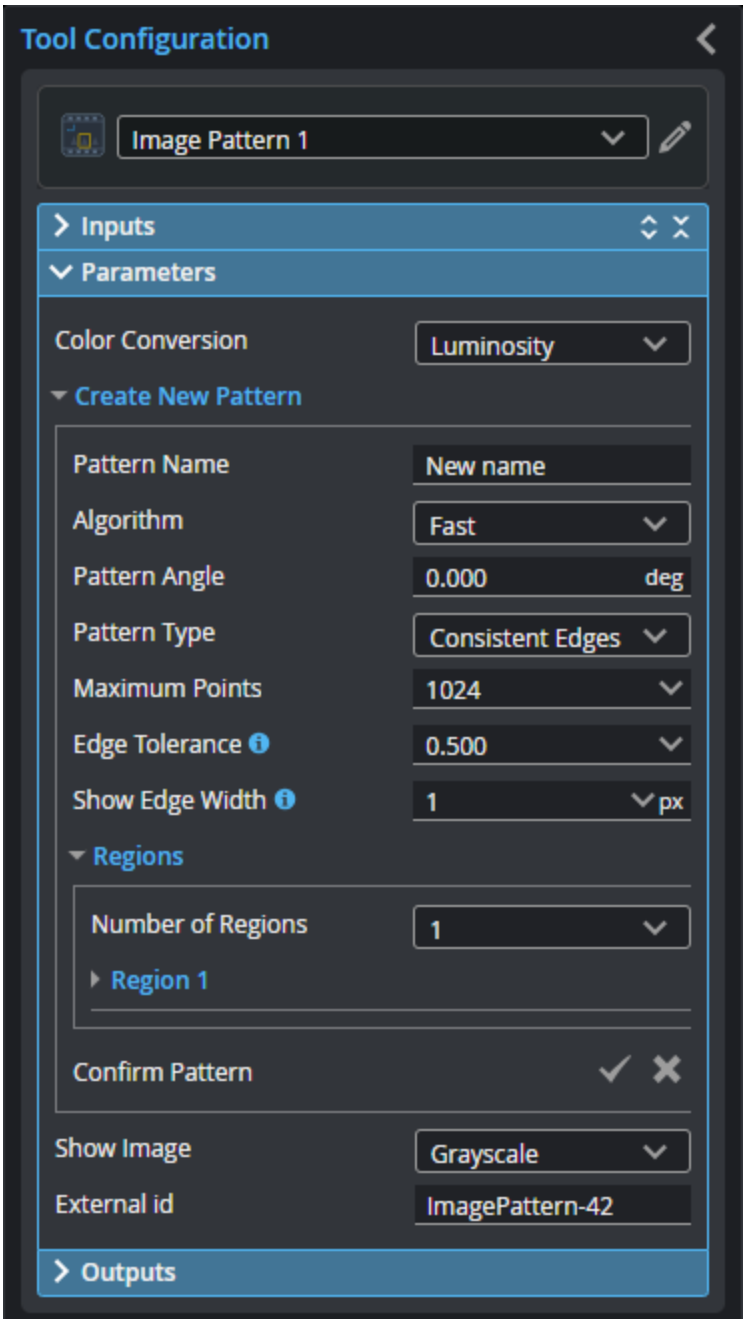
Parameters

You create, configure, and manage patterns in the expandable **Parameters** section.

Once you have at least one pattern created, you load it to configure the tool's matching parameters. For more information, see *Tool (Matching) Parameters* on page 459.

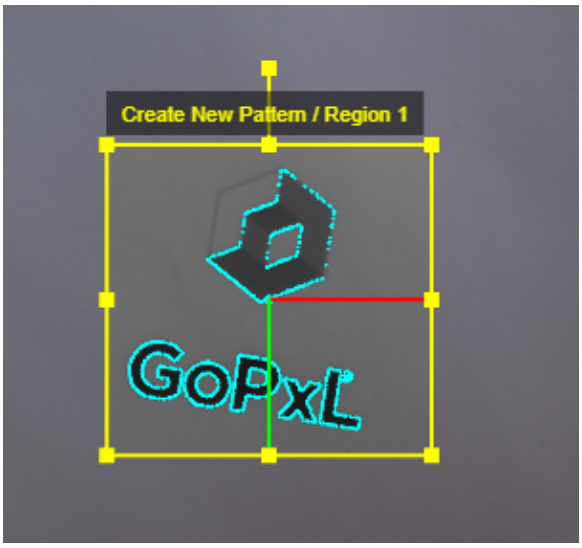
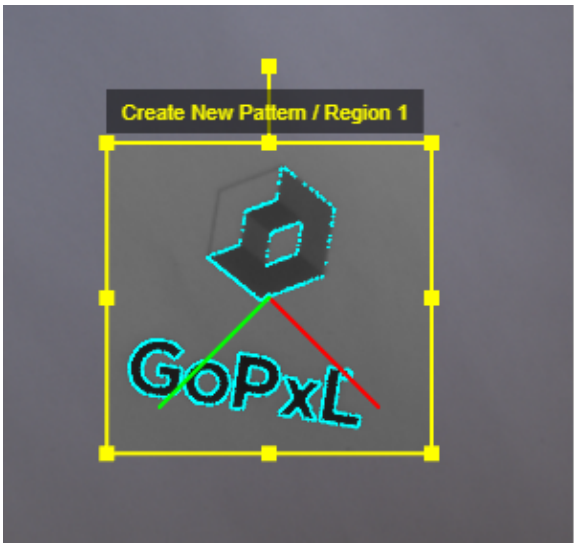
PATTERN PARAMETERS

You use the parameters described below to define a pattern.



Pattern Parameters

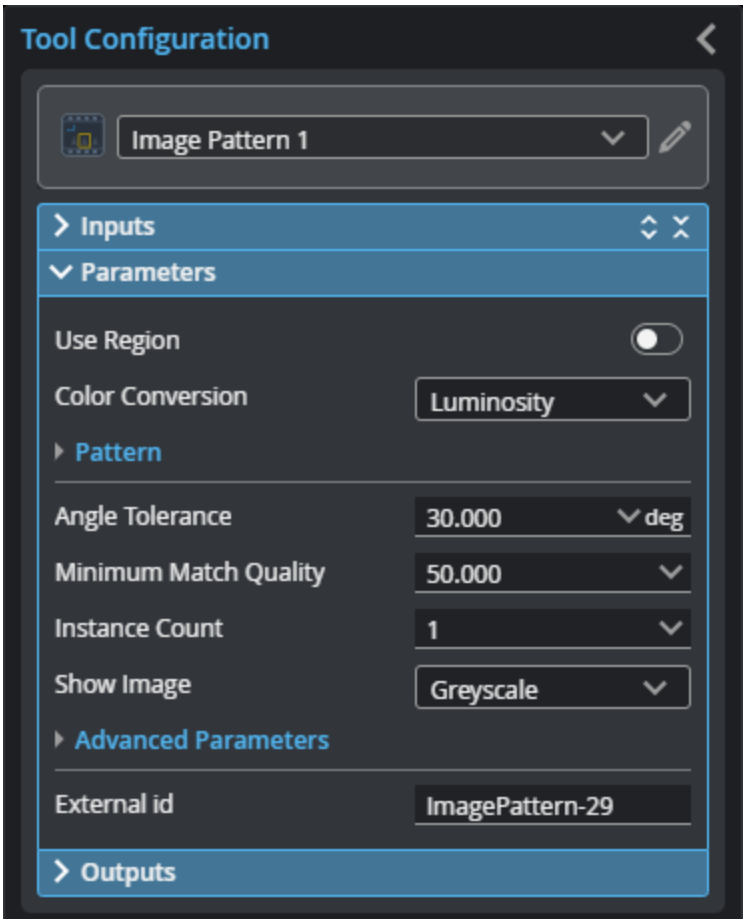
Parameter	Description
Pattern Name	The name of the pattern.
Algorithm	The algorithm the tool uses. One of the following: **Fast** A faster algorithm appropriate for many applications.

Parameter	Description
	Accurate This algorithm only uses the size and location of the region of interest. Ideal for complex patterns, as its detection reliability is better.
Pattern Angle (Fast algorithm)	The angle of the model. You typically leave this as 0.0 degrees. At zero degrees:  At 45 degrees: 
Pattern Type (Fast algorithm)	The type of the pattern. One of the following:

Parameter	Description
	Consistent Edges Use this type when your pattern contrasts well with the neighboring pixels, with sharp edges, or when your pattern is very different from the rest of the image in the region or regions of interest. **Thin Structure** Use this type when your pattern is made up primarily of thin elements, when the edges of your pattern are consistent between thin elements and the neighboring pixels, or when the contrast is the same for each thin element. When this mode is selected, set **Thin Structure Mode** to one of the following: Auto: Automatically chooses the best contrast for the thin elements. Dark: Chooses thin elements that are darker than their neighborhood. Bright: Chooses thin elements that are brighter than their neighborhood.
Maximum Points (Fast algorithm)	The maximum number of pixels the tool uses to create the pattern. Setting this to a lower value may reduce processing time when the tool tries to match features in an image to the pattern. Make sure that the tool matches the pattern to features as your application requires.
Edge Tolerance (Fast algorithm)	The minimum gradient (steepness) for a pixel to be considered part of an edge. Higher tolerance will only pick the best edges.
Show Edge Width	Displays the detected direction at each edge point. The setting doesn't affect detection accuracy.
Region / Regions	Settings related to the region. The Fast algorithm lets you define multiple, flexible regions. The Accurate algorithm lets you define a single, rectangular region. For more information, see *Regions* on page 214.

TOOL (MATCHING) PARAMETERS

After you have created a pattern for your application, set the parameters described below for matching the patterns to your targets in production.

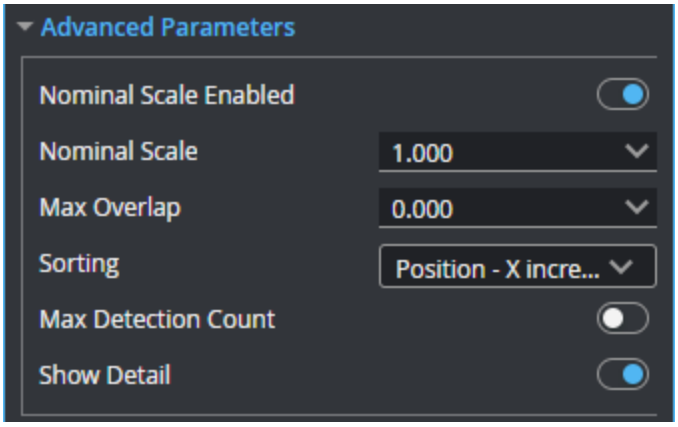


Matching Parameters

Parameter	Description
Use Region	Determines whether the tool limits searches for matching instances a user-defined region.
Region	Settings related to the region. For more information, see *Regions* on page 214.
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value: `pixel = 0.299R + 0.587G + 0.114B` When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green,

Parameter	Description
	and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.
Pattern	An expandable section that lets you choose a previously saved pattern to use for matching features in images. After you have chosen a pattern, you can set the tool's matching parameters (see below).
Angle Tolerance	The acceptable maximum angle (positive or negative rotation) of features in an image. Set this to 360 to allow any angle.
Minimum Match Quality	Minimum percentage of the pattern that must match in the image data for the tool to consider the instance as a valid match Instances of the pattern whose Match Quality measurement is less than this value will not be considered a match.
Instance Count	The maximum number of instances the tool will return.
Show Image	Determines how the image (or the image data in a region) is displayed in the data viewer. One of the following: **Original** Shows the original image data. In color sensors, this will be the color image. **Greyscale** Shows the image data converted to greyscale. The conversion is determined by the option you choose in the tool's **Color Conversion** parameter.
Advanced Parameters	An expandable section containing advanced parameters. For more information, see *Advanced Matching Parameters* on the next page.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

The matching parameters described below are available in the expandable **Advanced Parameters** section.

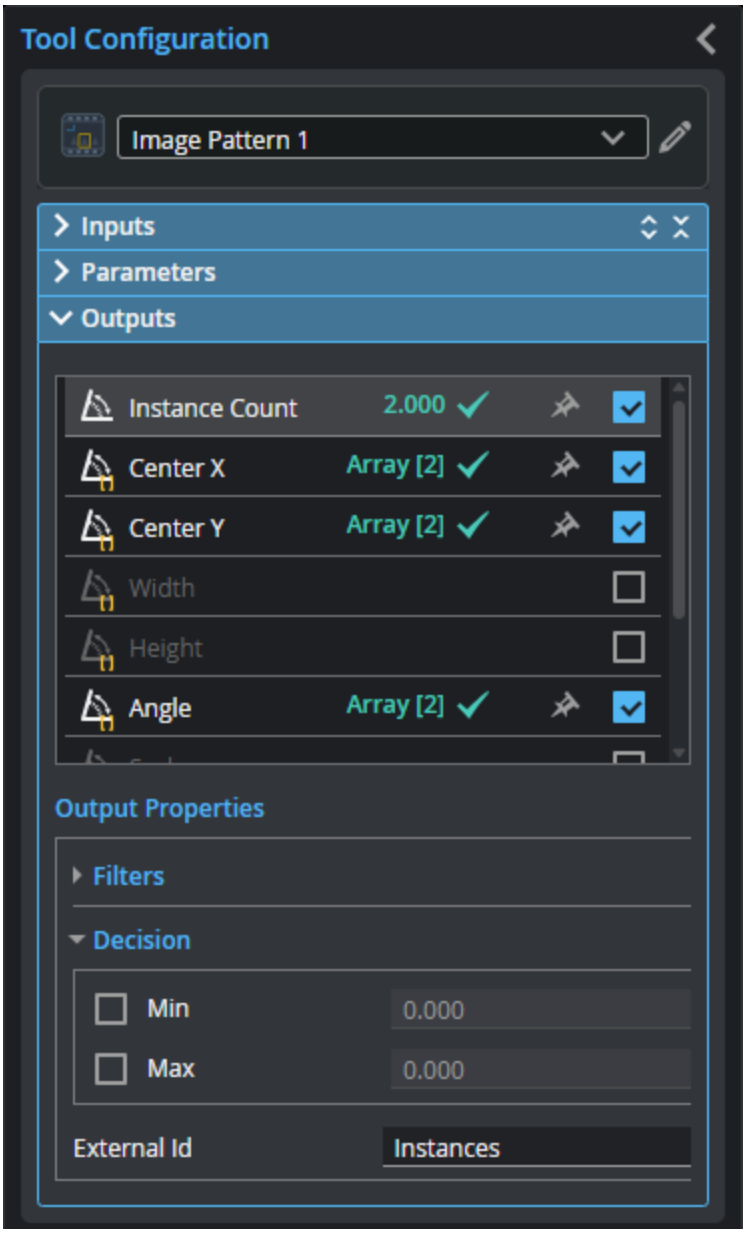


Advanced Matching Parameters

Parameter	Description
Nominal Scale Enabled	If **Nominal Scale Enabled** is checked, the tool displays a **Nominal Scale** setting. Otherwise, the tool displays **Minimum Scale** and **Maximum Scale** settings and uses the user-defined range.
Nominal Scale	The scale factor the tool requires to recognize an instance. Displayed when **Nominal Scale Enabled** is enabled.
Minimum Scale Maximum Scale	The minimum and maximum scale factors set the acceptable scaling range of an instance. Displayed when **Nominal Scale Enabled** is disabled.
Max Overlap	A value between 0 and 1 representing the maximum amount of overlap between instances, where 0.0 means all instances must be disconnected, and 1.0 means they can fully overlap.
Sorting	Orders measurements and features of the instances. One of the following: • Position - X increasing • Position - X decreasing • Position - Y increasing • Position - Y decreasing
Max Detection Count	Limits the number of initial match candidates to the value you provide. The number must be equal to or greater than the value of **Instance Count**. A large number can help the tool find difficult or partial matches, but increases processing time. A small number can reduce processing time, but reduces accuracy, especially with difficult or partial matches.
Show Detail	Toggles whether to overlay tool details. **Center only**: The center of the found pattern and its XYZ axes. **Center and All Edges**: The center of the found pattern and its XYZ axes, as well as the outline of the full pattern template. **Center and Matched Edges**: The center of the found pattern and its XYZ axes, as well as the matched edges only.
Instance Count	The maximum number of instances the tool will locate.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Measurements

Measurement
Instance Count Returns the total number of instances (matches).
Center X **Center Y** The X and Y positions of the center of the instance.
Width **Height** The width and height of the instance, calculated as model size × scale factor.
Angle The angle of the instance in relation to the loaded pattern.
Scale The scale of the instance in relation to the loaded pattern.
Match Quality The match quality of the instance.

Features

Type	Description
Point	The point representing the center of an instance.
Line	The line representing the rotated X axis, with respect to the loaded pattern.



For more information on geometric features, see *Geometric Features* on page 227.

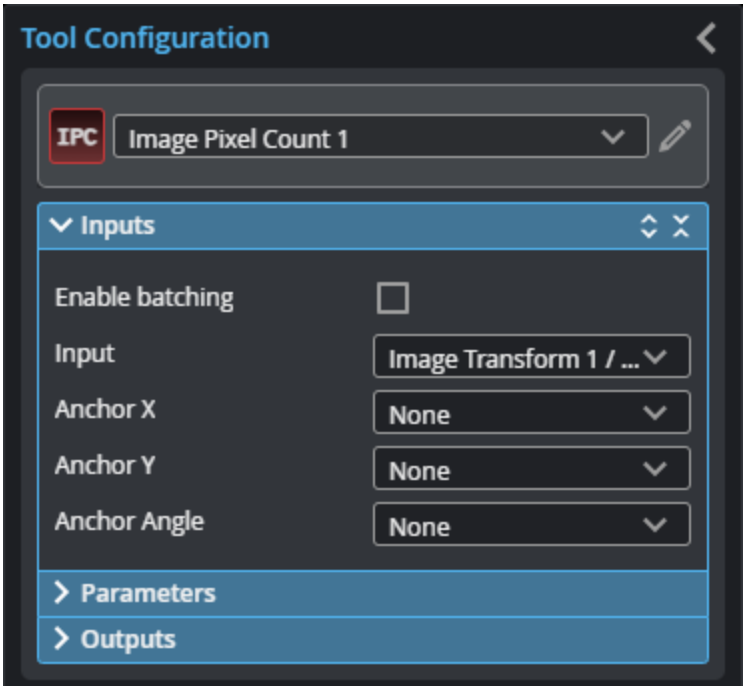
Image Pixel Count

The Image Pixel Count tool returns a count of the number of pixels in an image.

> When **Enable Array Output** is checked, the tool outputs the blobs in an array called Blob Surface. Be *sure* to enable Blob Surface in the **Outputs** section after enabling this parameter.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs



> To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

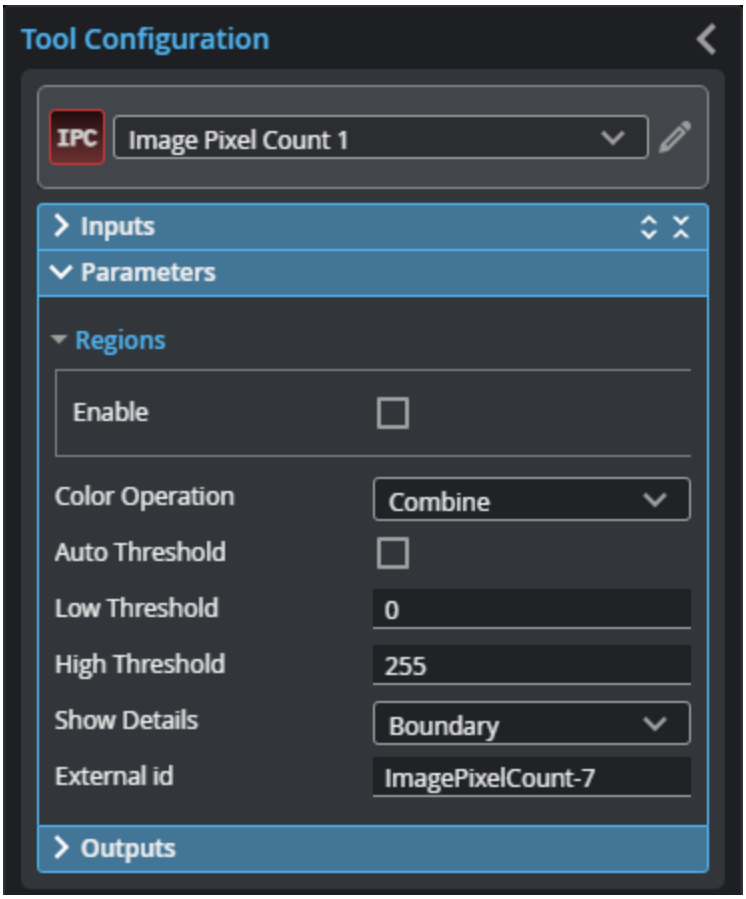
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Name	Description
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

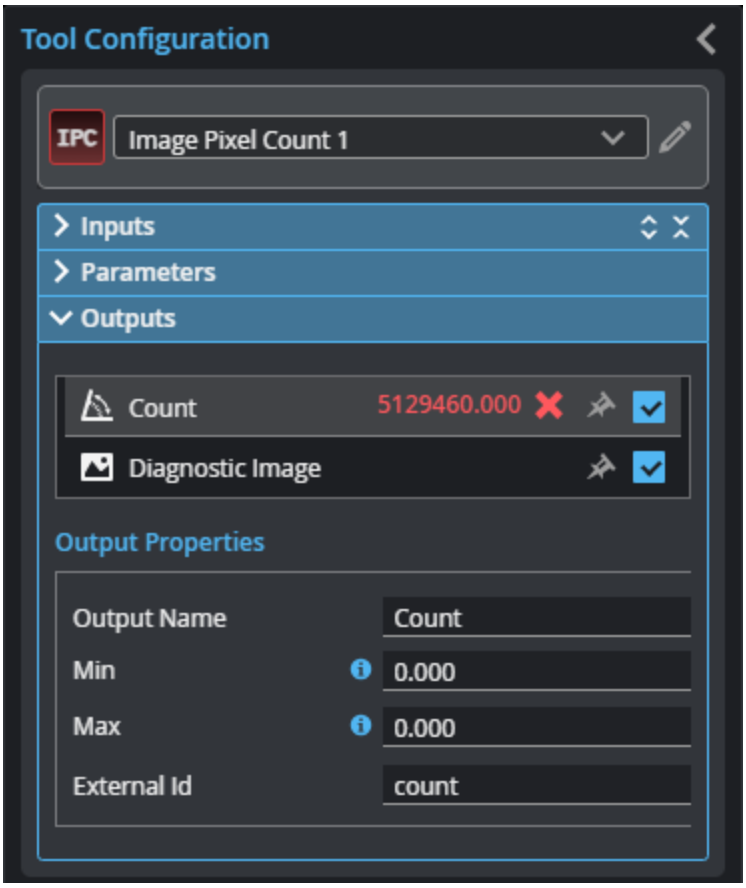


Parameters

Parameter	Description
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Measurements

Measurement	
Count	
The number of pixels.	

Data

Type	Description
Diagnostic Image	Image data you can use to evaluate the impact of the tool's parameters.

Image QR Code

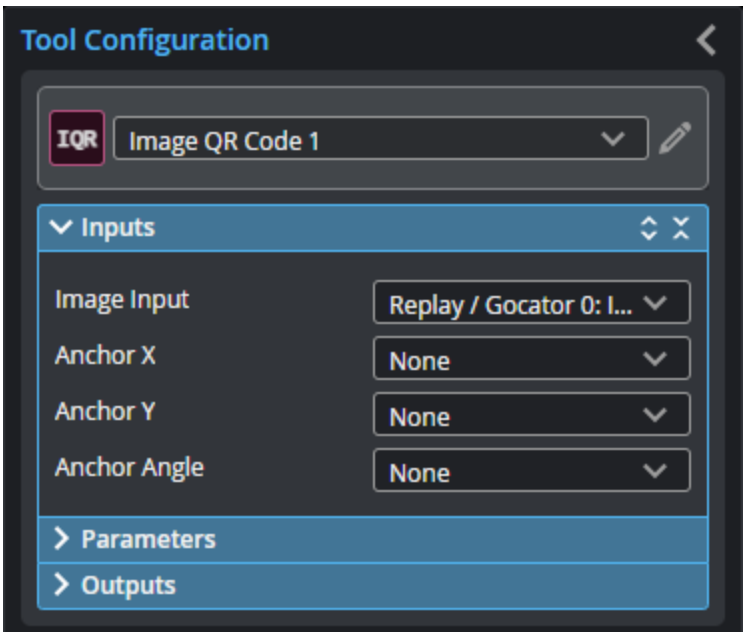
The QR Code tool lets you decode QR code data encoded in Image data. The tool supports grayscale and color images.

The tool returns the decoded string, as well as other measures such as its dimensions and the ECC family to which it belongs. The tool also provides measurements that can help you assess the print quality of the code symbols. Indicators from the following standards are available:

- ISO 15415
- ISO 29158



Inputs



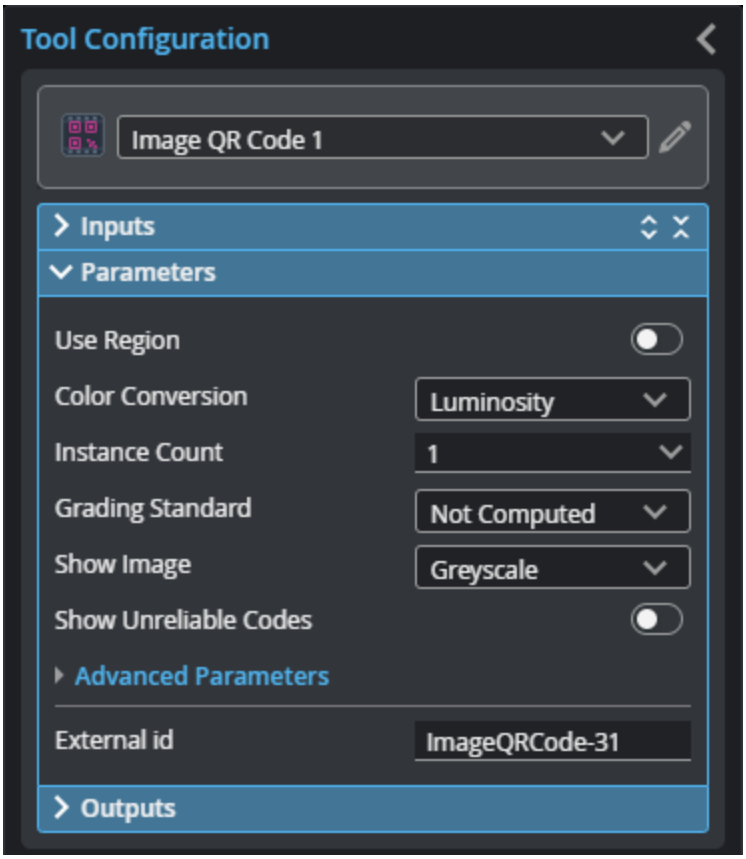
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Use Region Region	Enable **Use Region** and set the values for the regions.
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value: `pixel = 0.299R + 0.587G + 0.114B` When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.
Instance Count	The maximum number of matrix codes to decode in a frame.

Parameter	Description
Grading Standard	The standard the tool will use to assess the quality of the matrix code. Grading measurements are added in the tool's **Outputs** section. One of the following: • ISO 15415 • ISO 29158
Compute Grading	The type of grading the tool returns in the grading measurements. One of the following: **Number** - Grade measurements return a numerical value between 0 and 4. **Letter** - Grade measurements return A, B, C, D, or F. **Letter and Number** - Both types of outputs are returned. Numeric measurements are distinguished from letter-based ones by "Number" in the name.
Read Mode	Whether the tool prioritizes speed or detection quality. **Speed** The tool stops searching for matrix codes as soon as one of the following happens: • It finds the number of codes selected in **Instance Count**. • It has finished processing the image. This mode results in shorter processing times. **Quality** The tool tries to improve its detection until it has finished processing the image. This mode results in better grading results.
Show Image	Determines how the image (or the image data in a region) is displayed in the data viewer. One of the following: **Original** Shows the original image data. In color sensors, this will be the color image. **Greyscale** Shows the image data converted to greyscale. The conversion is determined by the option you choose in the tool's **Color Conversion** parameter.
Show Unreliable Codes	Shows unreliable codes (objects that are likely data matrix codes, but can't be decoded).
Advanced Parameters	Expandable section containing advanced settings. For more information, see

Parameter	Description
	Parameters - Advanced below.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

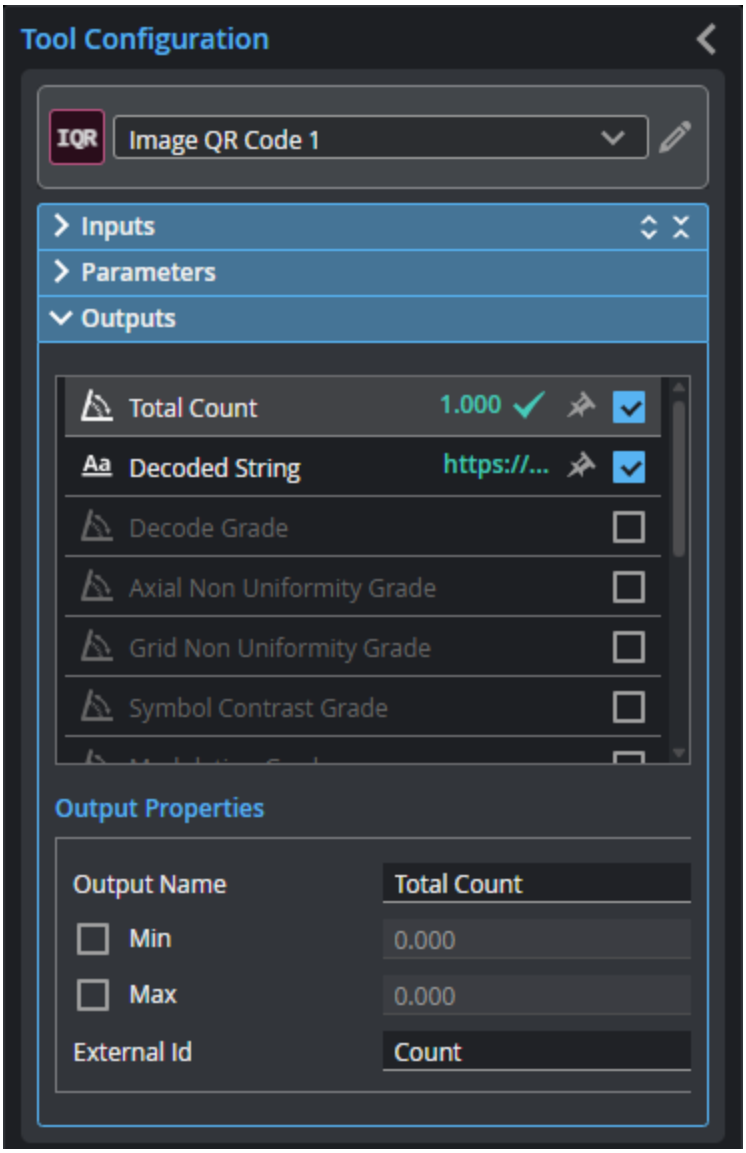
Parameters - Advanced

Parameter	Description
Detection Mode	The detection method the tool uses to find QR codes. **Adaptive Threshold** - Detects finder patterns based on adaptive thresholding. **Gradient** - Detects finder patterns based on gradients.
Search Model Params	The QR code models the tool searches for. **QR Model 1** - Searches for QR Model 1 **QR Model 2** - Searches for QR Model 2 or 2005. **Micro QR** - Searches for Micro QR.
Search Parameters	**Min Version** / **Max Version** - The minimum and maximum of QR codes the tool searches for. **Acceptable Deformation** - A percentage representing the value of the short side of the code divided by its long side. **Min Score** Minimum pattern finder score needed to consider a finder pattern found. **Correction Aggressiveness** The minimum threshold for cell polarity confidence. When the confidence is below the threshold, the tool further processes the code to improve polarity detection. The cell polarity confidence is the confidence in the cell digitization result, on a scale from 0 to 1. Increasing the threshold can improve reading of overprinted or underprinted QR codes. Large values can affect the speed and increase false positive corrections. **Scan Precision** One of the following: • Automatic: The tool determines scan precision automatically. • Fine: The tool performs a fine scan to find the QR codes. Best for small QR codes.

Parameter	Description
	• Coarse: The tool performs a coarse scan to find the QR codes. Best for medium to large QR codes. **Minimum Contrast Threshold** The minimum difference in grayscale value between a pixel and its local background for the pixel to be part of the foreground.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Some measurements are only displayed when you select a grading standard in **Compute Grading** (see *Parameters* on page 470). "Modules" refers to the cells or "dots" in the grid of the matrix code.

Measurements (all grading standards)

Measurement
Total Count The number of matrix codes decoded.
Decoded String The decoded string, packed as an array of float-point values.

When **Compute Grading** is set to "Letter and Number", the tool outputs both "Number Grade" measurements containing a value between 4 and 0 and "Grade" outputs that contain A, B, C, D, or F. If the setting is either "Letter" or "Number", the tool only returns "Grade" outputs, which contain the selected type.

Measurements (ISO 15415)

Measurement
Decode Grade **Decoding Number Grade** Whether the code was decoded.
Axial Non Uniformity Grade **Axial Non Uniformity Number Grade** Grade for uneven scaling along the X and Y axis.
Grid Non Uniformity Grade **Grid Non Uniformity Number Grade** How much the measured position of the grid intersection point deviates from the expected position. Codes sheered along the X and/or Y axis can result in a low grade. The grade can be unstable and its usefulness is limited.
Symbol Contrast Grade **Symbol Contrast Number Grade** The difference between the dark and light areas of a code, including the quiet zones.
Modulation Grade **Modulation Number Grade**

Measurement
Uniformity of the reflectance of dark and light areas throughout the code. Irregular or mottled regions can result in a low grade.
Reflectance Margin Grade **Reflectance Margin Number Grade** How distinguishable the modules are relative to the global threshold (the mean reflectance of the brightest and the darkest module).
Fixed Pattern Damage Grade **Fixed Pattern Damage Number Grade** A measure of any damage (missing data or data where it is not expected) to the finder pattern (the L-shaped along the left and bottom of the code), the quiet zone (margins outside the matrix), or the clock track (the alternating "dots" along the top). The value returned is an average of the grades of these areas of concern. A higher value indicates the matrix code is more correctly located and identified.
Unused Error Correction Grade **Unused Error Correction Number Grade** Grade representing how little error correction to reconstruct data in damaged or poorly printed codes was used. A higher grade means less error correction was applied.
Horizontal Print Growth **Vertical Print Growth** **Horizontal Print Growth Number** **Vertical Print Growth Number** How much the matrix code may have grown or shrunk compared to the expected size horizontally and vertically, respecitively.

Measurements (ISO 29158)

Measurement
Cell Contrast Grade **Cell Contrast Number Grade** Relative difference between the mean reflectance of the darkest and brightest cells in the code.
Cell Modulation Grade **Cell Modulation Number Grade** The variability and reliability of the black and white colors. A low grade means the tool will have difficulty distinguishing between black and white cells.
Minimum Reflectance Grade **Minimum Reflectance Number Grade** The ratio between the smallest and largest gray values in the barcode. For the "Number" grade, the value is 4 if the smallest gray value of the barcode is <= half of its highest value, and 0 otherwise.
Fixed Pattern Damage Grade

Measurement
Fixed Pattern Damage Number Grade How likely the symbol is correctly located and identified in the code.
Mean Light In Required Bounds **Mean Light In Required Number Bounds** Whether mean light is 70%~86% of the maximum gray level.
Mean Light **Mean Number Light** The mean value of the pixels in the center of the light cells.

Image Track

The Image Track tool lets you perform quality control and inspection of gaps, glue bead, pipes, and rectangular features in images. The tool supports multiple regions.

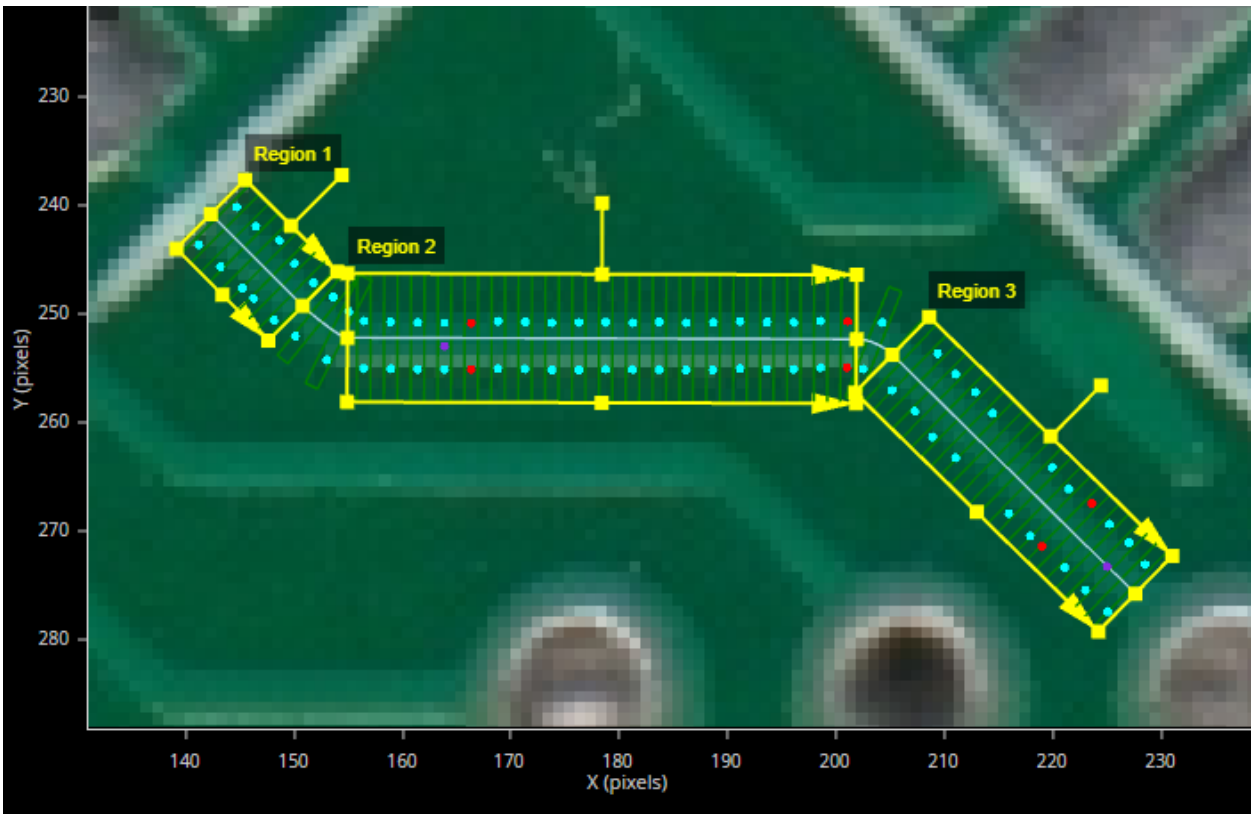
As you add regions, the tool displays a new region "connected" to the previous region. To define the expected track, you resize, position, and rotate the regions you add. The tool can "fill in" curves in spaces between regions, or ignore the spaces.

The tool tries to find track edges (rising or falling intensity transitions greater than the provided threshold) by searching in the defined sample areas (perpendicular to the center line) from the outer edge of the regions toward the center line.

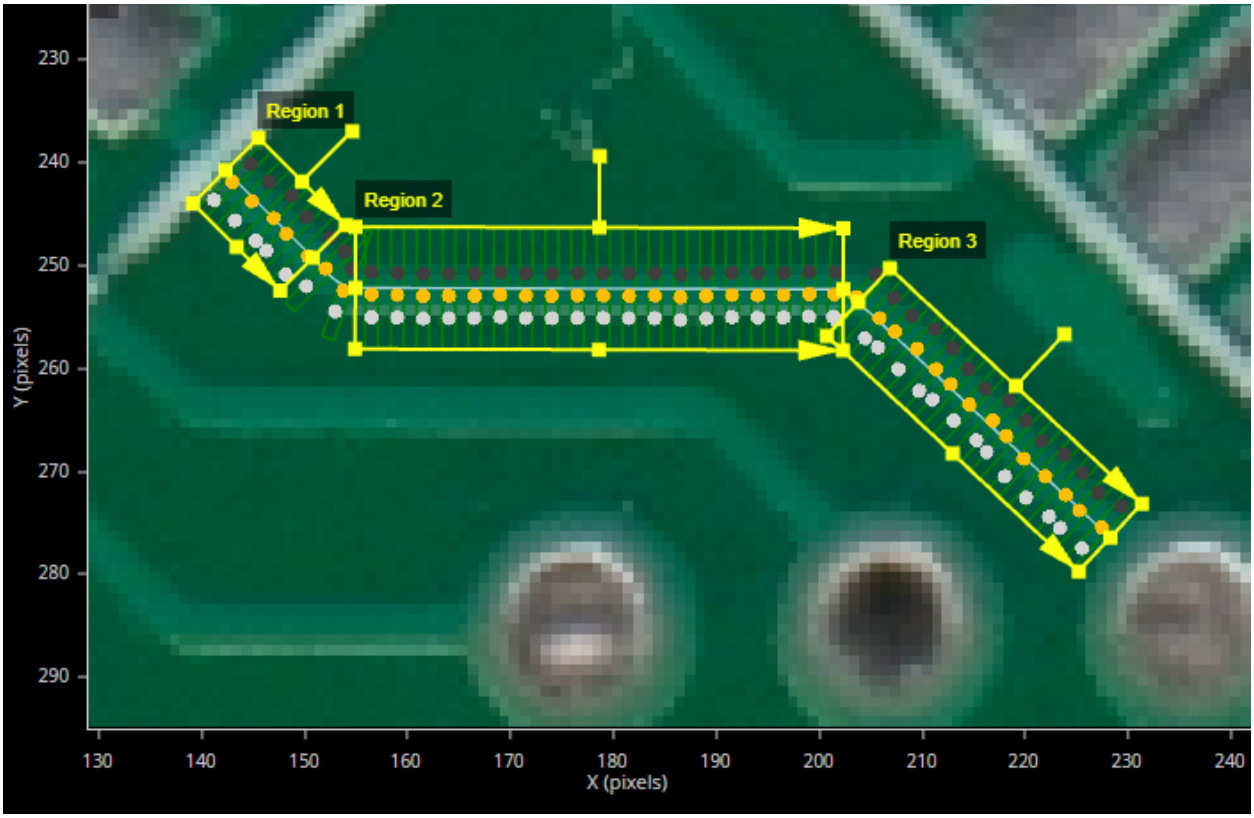
The tool displays the following dots:

- Cyan dots represent valid edge points.
- Red dots represent invalid edge points (those excluded by the **Advanced Parameters** > **Distance Outlier Fraction** parameter).
- Violet dots represent either the Min Distance or Max Distance.

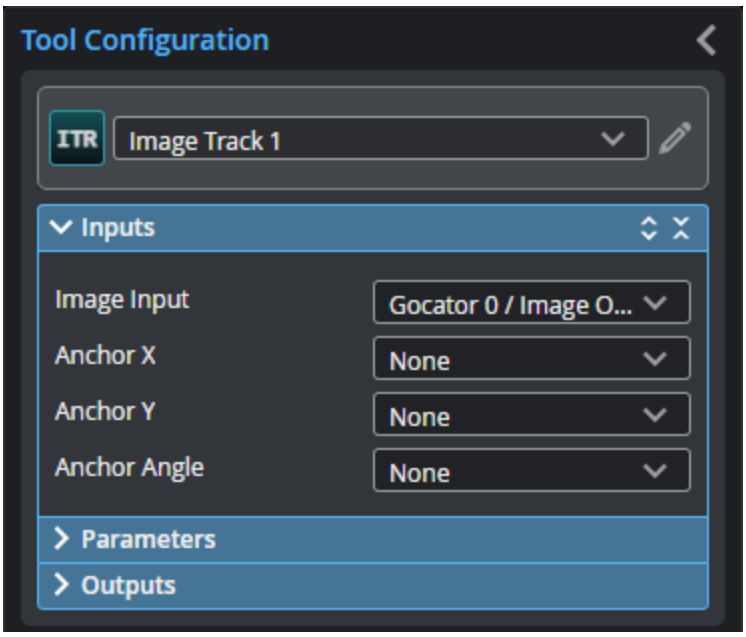
In the following, three connected regions follow a PCB trace. The arrows indicate the "direction" of the track and serve to show where new regions will be added.



The tool displays the Point geometric features related to the center points and the points at the edge of the track as larger dots. The colors are selected automatically by the tool, but you can change these in the Displayed Outputs panel; for more information, see *Displaying and Pinning Outputs* on page 94. To display these points, enable them in the tool's **Outputs** section and select or pin the output.



Inputs



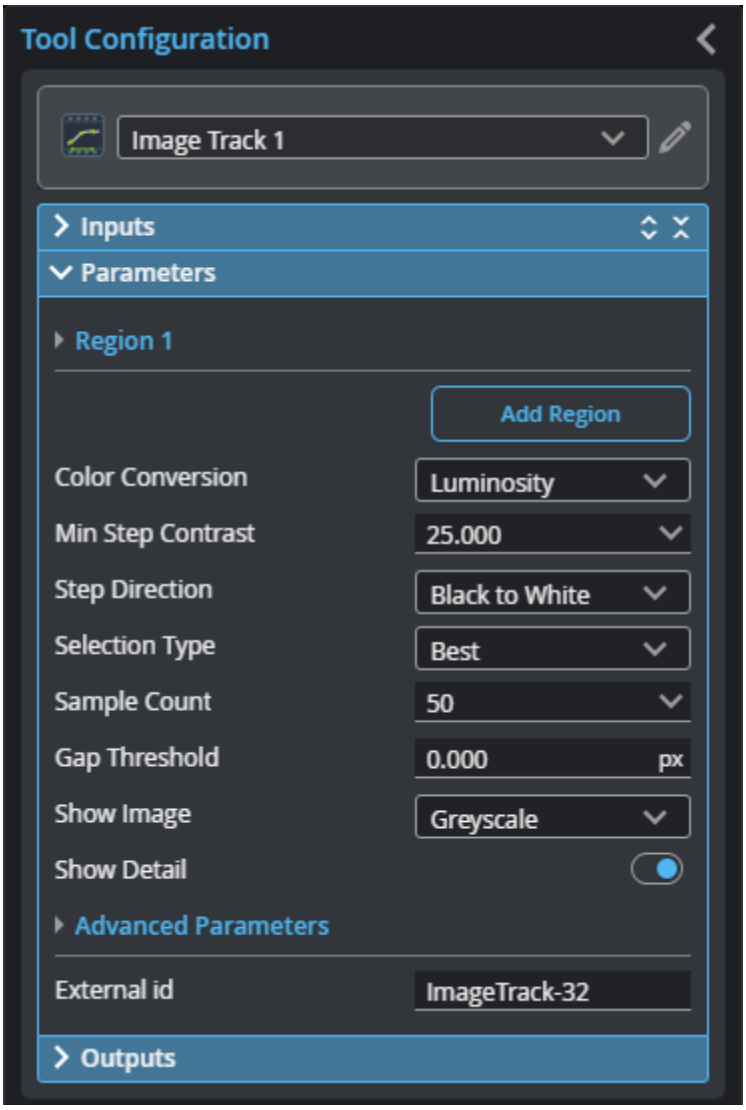
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Image Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

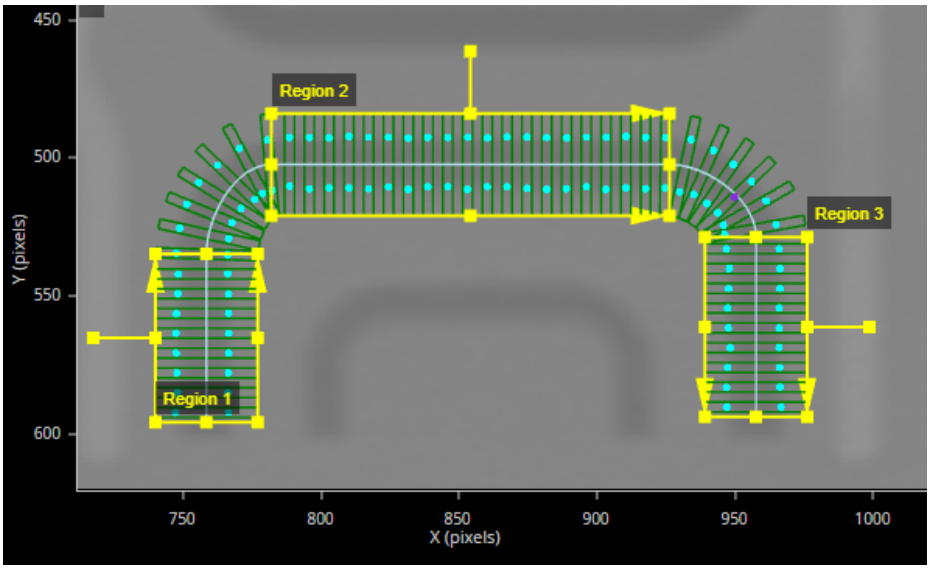
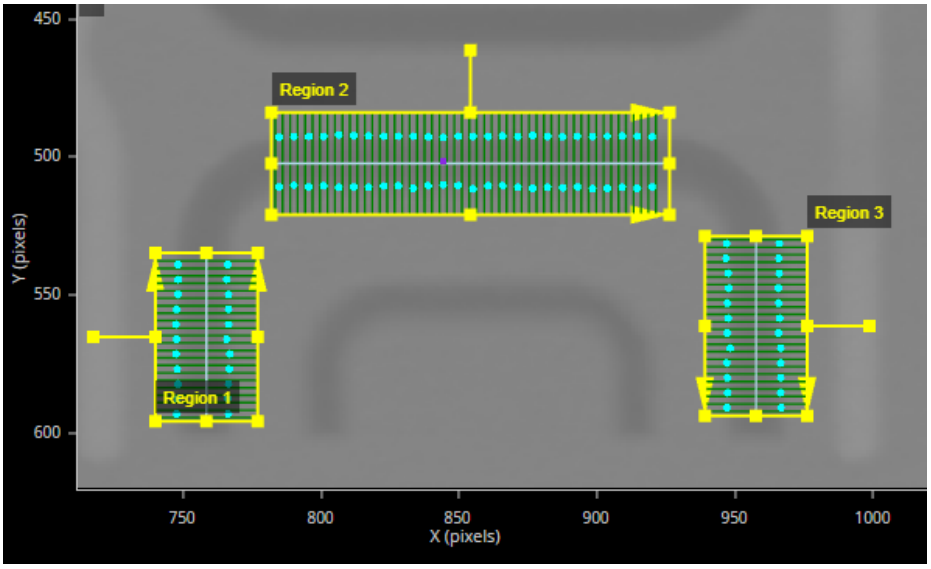
Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

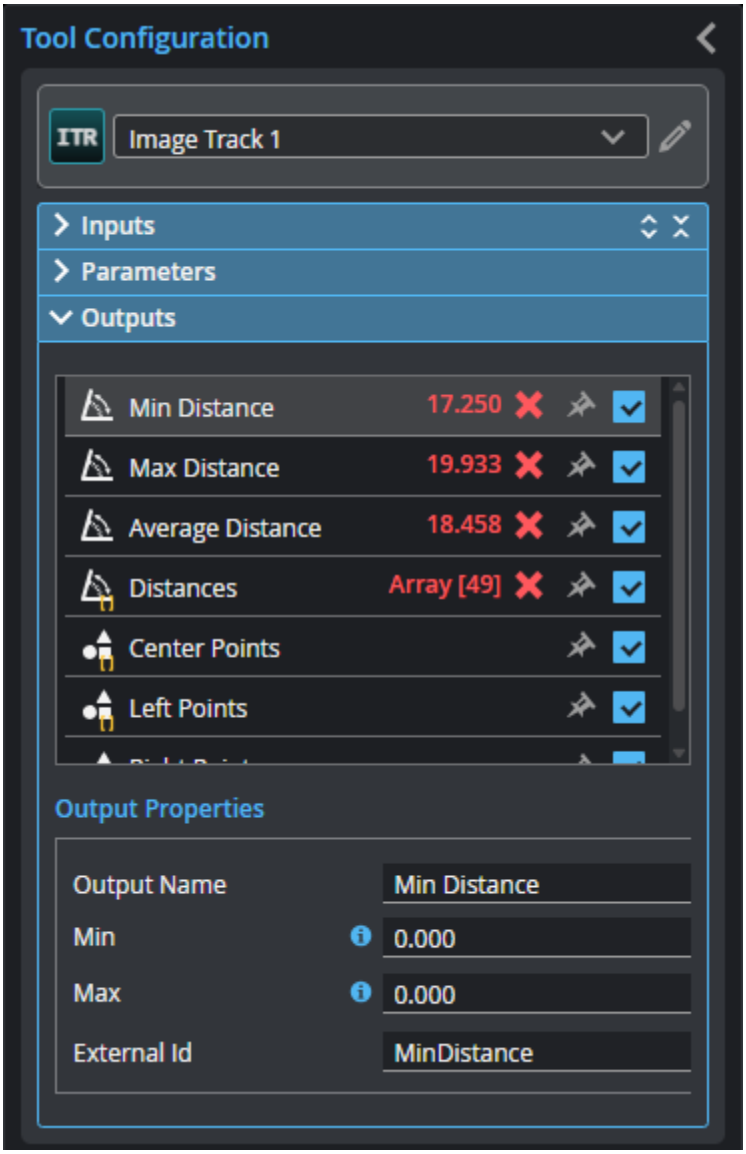
Parameter	Description
Region {n}	When expanded, displays the region-related settings. In addition to size, location, and rotation, you can define whether the tool adds sample areas between a region and the previous region by setting **Previous Path** to Spline. In the following, splines (with **Spline Roundness** set to 1) are enabled for Region 2 and Region 3.

Parameter	Description
	 When a spline is not enabled, sampling is limited to the regions. 
Add Region	Adds a region.
Color Conversion	Lets you choose how a color image is converted internally to a monochrome image for use with the tool. Use this parameter to choose the color channel (or a combination of channels) that helps the tool detect features in your application better. Some tools compare the values of the converted image to a threshold you set. Other tools simply process the converted image directly. When an Image tool's **Color Conversion** parameter is set to Luminosity, the tool uses the following formula to determine the converted value:

Parameter	Description
	`pixel = 0.299R + 0.587G + 0.114B` When an Image tool's **Color Conversion** parameter is set to Red, Green, or Blue, the tool uses that channel to create the greyscale image on which tools run. When the **Color Conversion** parameter is set to Color Picker, set the red, green, and blue values to a color. The tool calculates the distance of the image's RGB pixels relative to this color.
Threshold	The intensity threshold above or below which a pixel will be considered the edge of a track. On color devices, the RGB values are converted to an 8-bit monochrome image.
Step Direction	Whether the tool looks for the rising (increasing) or falling (decreasing) intensity values in the sample areas.
Selection Type	One of the following: Best or First. Determines whether the best transition (greatest intensity difference) or first transition exceeding the value of **Threshold** is used as an edge point.
Sample Count	The number of areas in which the tool searches for an edge. The areas are spaced equally across all regions, as well as areas between regions where splines are enabled.
Show Image	Determines how the image (or the image data in a region) is displayed in the data viewer. One of the following: **Original** Shows the original image data. In color sensors, this will be the color image. **Greyscale** Shows the image data converted to greyscale. The conversion is determined by the option you choose in the tool's **Color Conversion** parameter.
Show Detail	Displays or hides area lines and center lines in the regions.
Advanced Parameters	When expanded, displays advanced parameters. **Distance Outlier Fraction** The percentage of outlier points to exclude. Setting this to a small value can help the tool fit the line better. **Step Smoothing** The window in which the tool performs smoothing. The size of the window along which the tool calculates an average for determining steps. The setting is useful for averaging out noise.

Parameter	Description
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Measurements

Measurement
Min Distance

Measurement
Max Distance **Average Distance** The minimum, maximum, and average distances of edge points from the center line.
Distances An array of distances from the center line.

Data

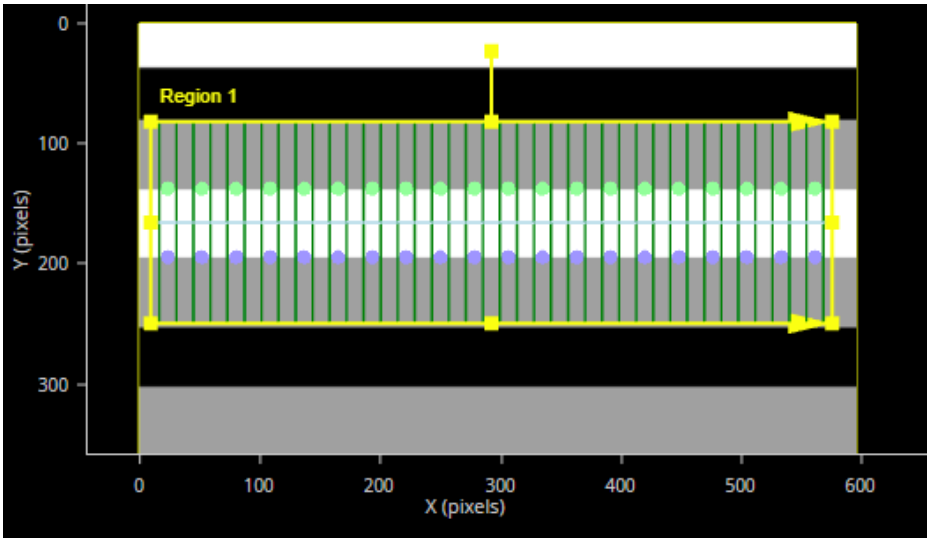
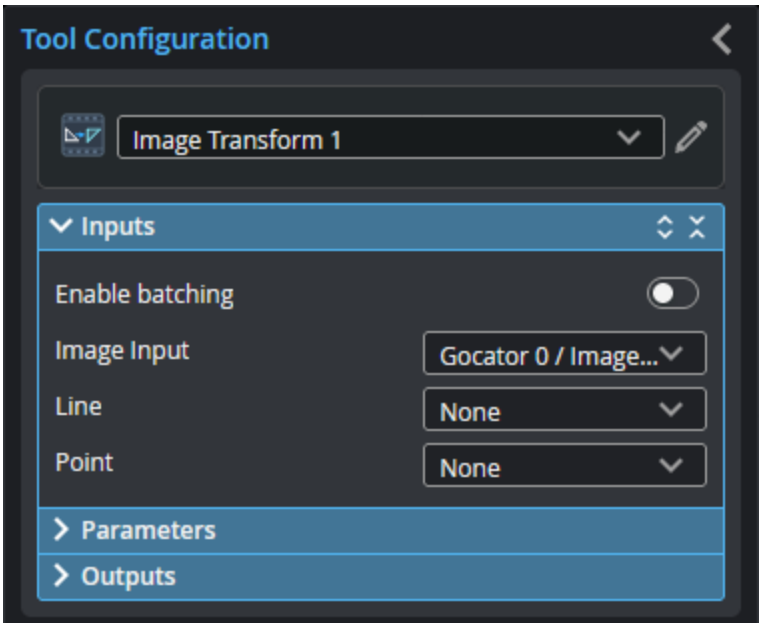
Type	Description
Center Points **Left Points** **Right Points**	Arrays of the Point geometric features: center points, as well as points at the edge of the track. The Left and Right array names are arbitrary. In the following, the green points at the top of the region are "Left" points. The violet ones are "Right" points. 

Image Transform

The Image Transform tool generates a new image based on the coordinate system of geometric features the tool uses as input. The tool can take a line or origin point to define this new coordinate system. You can then apply the built-in measurement tools or GDK tools to this new data. The tool can help improve repeatability of your measurements.

Inputs



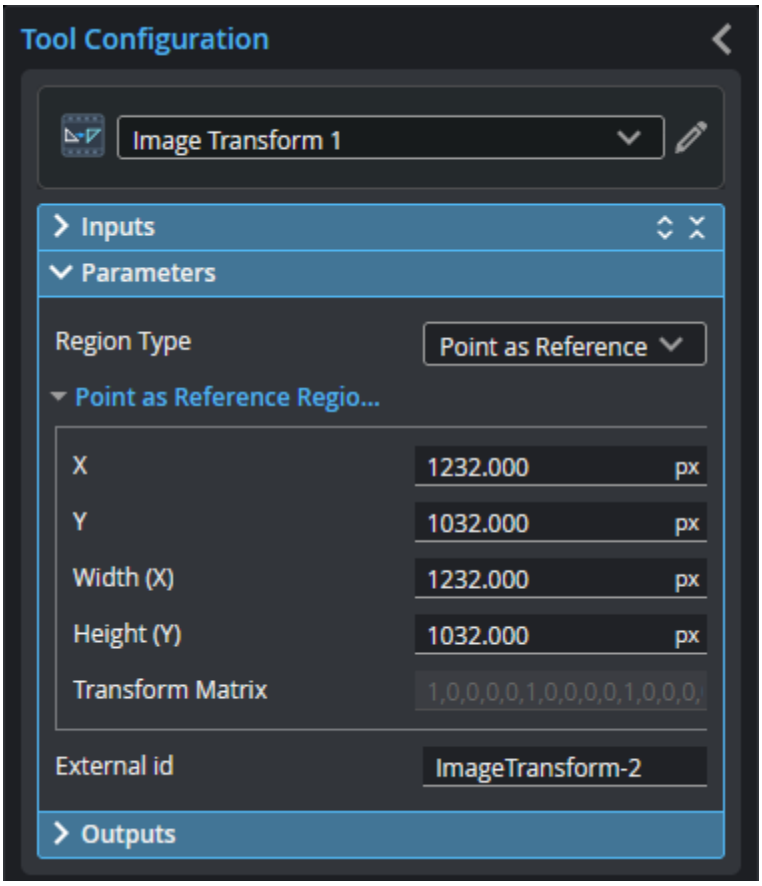
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Image Input	The data the tool applies measurements to or processes.
Line Point	The Line and Point geometric features the tool uses to perform transformations.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

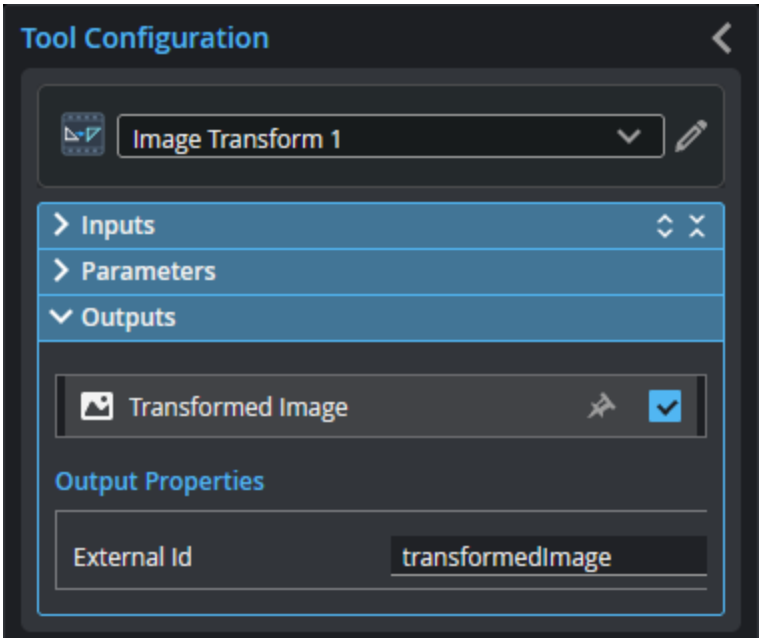


Parameters

Parameter	Description
Region Type	The type of the region. One of the following: Full Image or Point as Reference. **Full Image** The entire image is transformed using the Line and Point geometric feature inputs, if they are set, and a fixed transformation is used if **Add Fixed Transform** is enabled and its related values are set. **Point as Reference** Crops the image using the size and position parameters under **Point as Reference Region**. If the Line and Point geometric feature inputs are set, the tool uses them to transform the image.
Output Size	Determines the size of the size of the output. Use this to avoid cropping when the tool results in a rotated image. One of the following: **Fixed Size** The original size of the input image is used for the transformed output. **Fit to Content**

Parameter	Description
	The output image is resized to ensure that no part of the transformed image is cropped.
Add Fixed Transform	When enabled, displays positional offset and angle fields you can use to manually set transformations. If you also set geometric features as inputs, 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.

Outputs



Data

Type	Description
Transformed Image	The transformed image.

Measurement Formula Tool

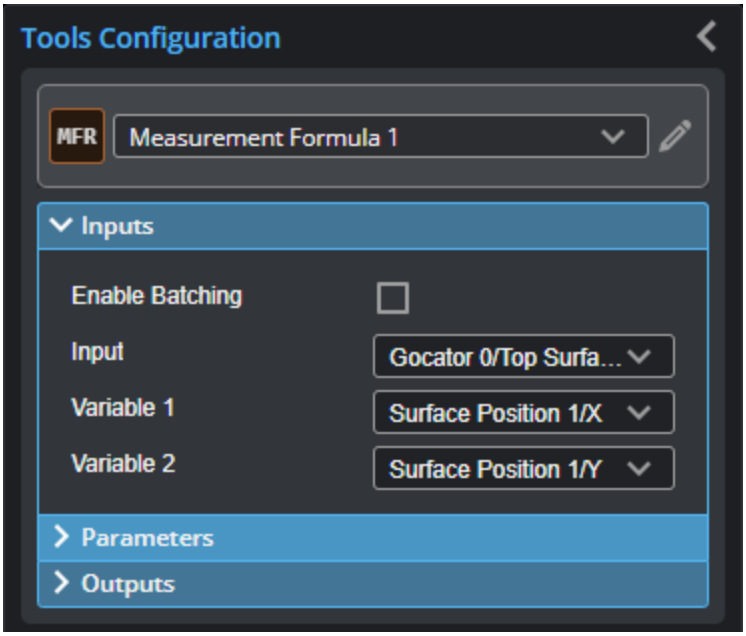
The Measurement Formula tool lets you create a mathematical expression containing aliases to one or more measurements from other measurement tools, up to a maximum of 20. The result of the calculation is returned in the tool's Result measurement.

The tool's default timeout is 5 seconds: if the tool fails to complete the calculation after 5 seconds, the result is set to Invalid.

The tool uses the C++ Mathematical Expression Toolkit Library (ExprTk). For full details on what it supports, see the toolkit's page (http://www.partow.net/programming/exprtk/).

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

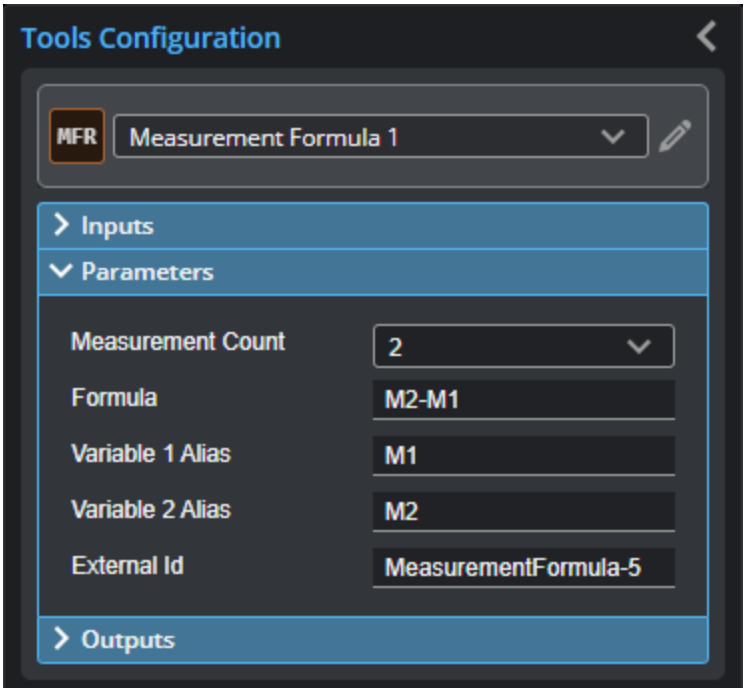


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Input	Set this parameter to any available input. The selected input has no effect on the tool's calculations.
Variable {n}	A measurement from another tool that you use in the formula. The **Measurement Count** parameter controls the number of variable inputs.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

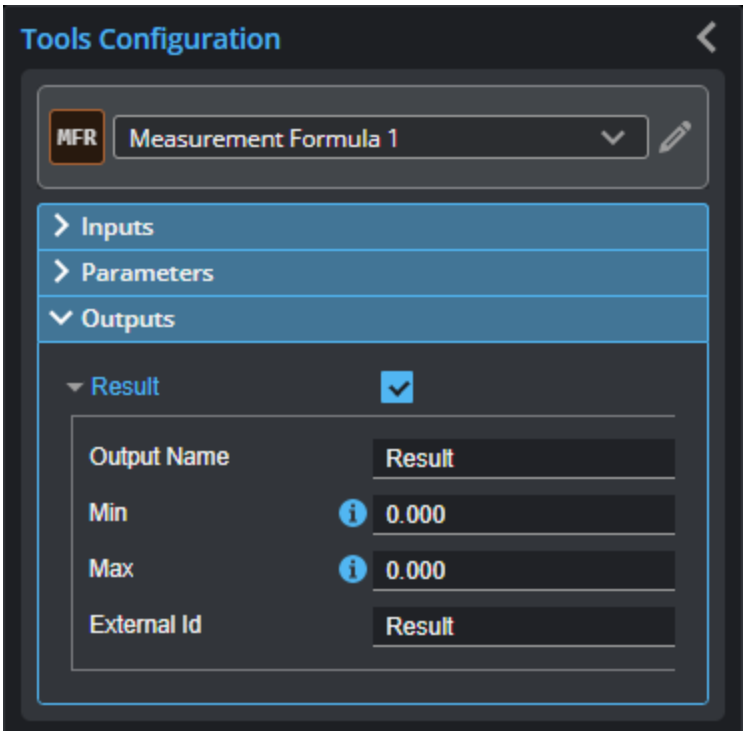
Parameter	Description
Measurement Count	Sets the number of variable inputs in the **Inputs** section. For each, a **Variable {n}** input and a **Variable {n} Alias** parameter is added, and for use in the expression.
Formula	The mathematical expression. Use the variable aliases from the **Variable {n} Alias** parameters in the expression. For example: • M1*3+M2/2+4 • tan(M1/180*pi) • max(M1, M2, M3) The tool supports complex mathematical formulas. For example, the following: $\sqrt{\frac{1-a\times e^{2t}+w^{\pi}}{sin(\frac{2x}{\pi})+cos(\frac{\pi}{y})}}$ can be produced using the following formula: `sqrt((1-a*exp(2t)+w^(pi))/(sin(2x/pi)+cos(pi/y)))`

Parameter	Description

	The following shows how to perform a *while* loop:
	`y:=0; while(x<10){y:=y+x;x+=1;};y`
	Note that ":=" is the assignment operator, and "=" is the equality operator.
	The following shows how to use an if-else structure:
	`if(x>2){x:=x+2;}else{x:=x-2;};x`

Variable {n} Alias	The alias of Variable {n} (that is, a measurement set in the **Inputs** section). Use these aliases in **Formula**.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	
Result	The result of the calculation in **Formula**.

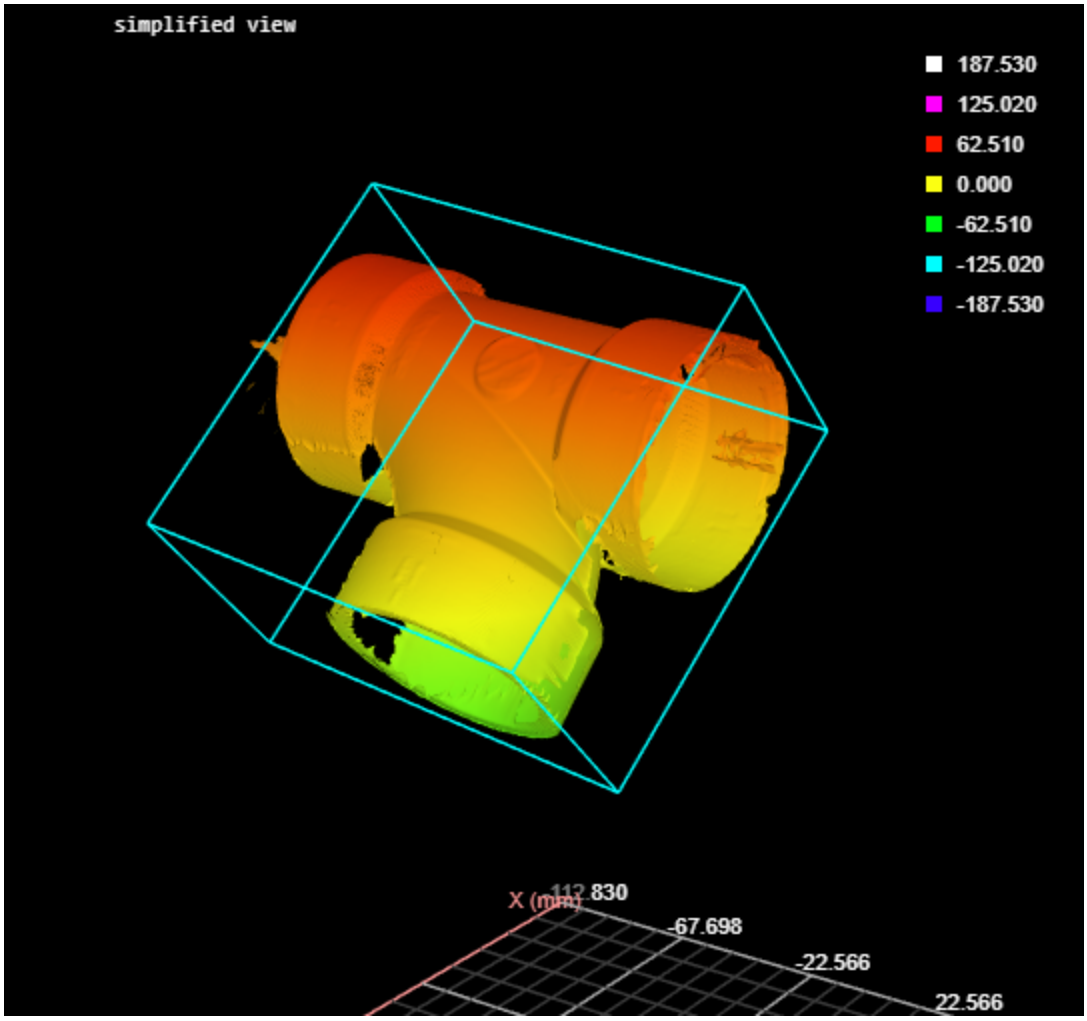
Mesh Tools

The Mesh tools are only intended for use with G2 sensors.

Mesh Bounding Box

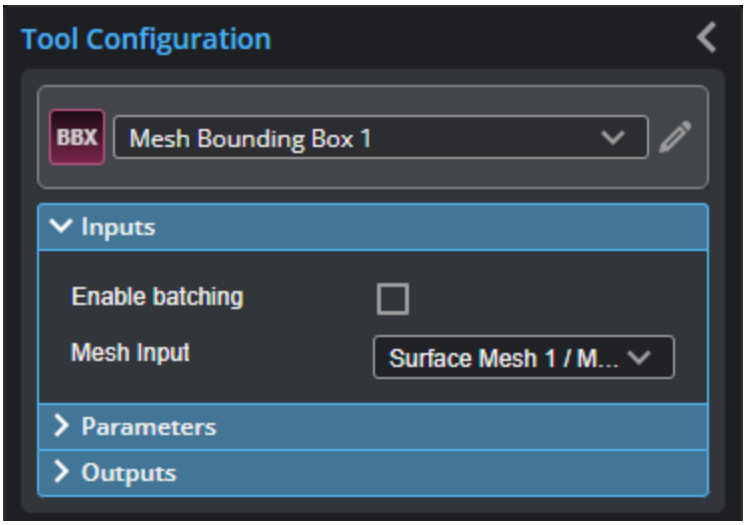
> This tool is *only* intended for use with G2 sensors.

The Mesh Bounding Box tool takes in Mesh scan data (produced by the Surface Mesh tool and some other Mesh tools) and returns measurements related to the bounding box encapsulating the scan data in the region of interest, such as the rotation of the bounding box, the dimensions of the bounding box, and its location. In addition to a Point geometric feature, the tool returns the Mesh data in the bounding box. You can apply one of the other Mesh tools to this data, or after extracting Surface data using Mesh Projection or Mesh Plane, you can apply any built-in or custom GDK-based tool to the extracted surface data.



Inputs

You configure the tool's inputs in the expandable **Inputs** section.

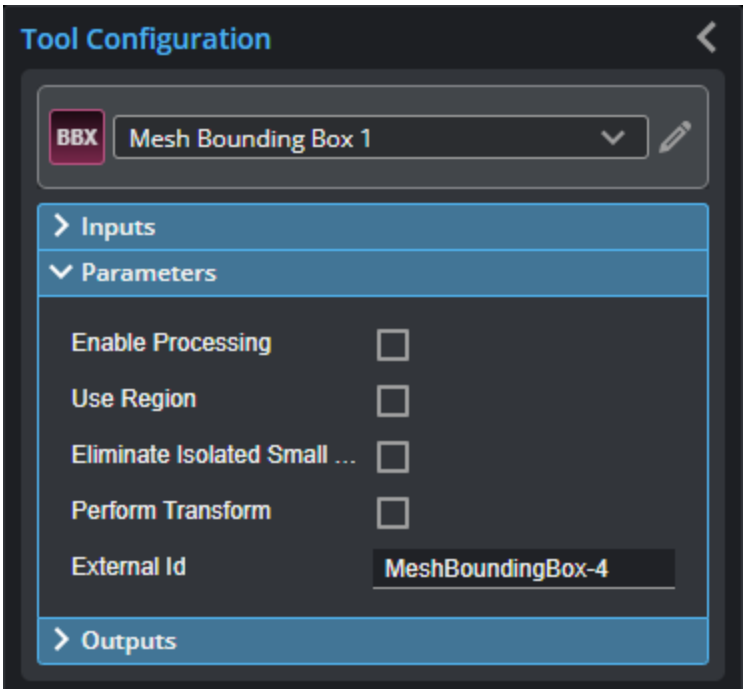


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Mesh Input	The data the tool applies measurements to or processes.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

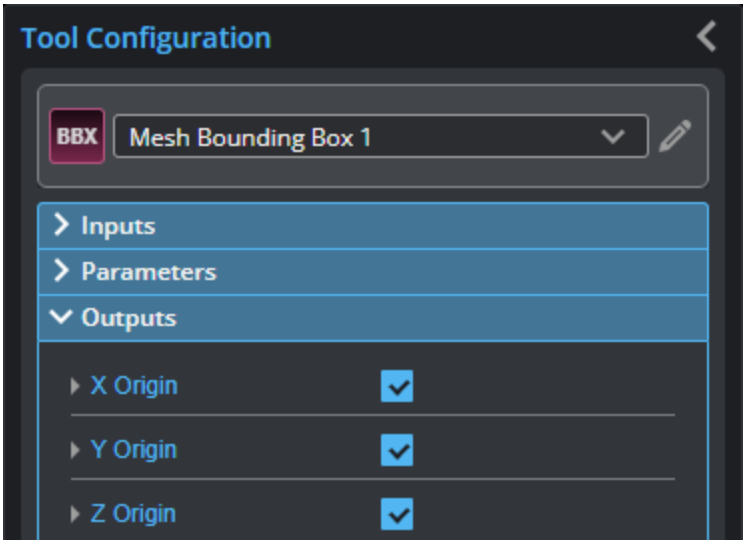


Parameters

Parameter	Description
Enable Processing	Causes the tool to start processing scan data from individual sensors. Make sure to properly configure the tool *before* enabling this option.
Use Region	Determines whether the tool limits the bounding box fit to Mesh data in a user-defined region to fit a bounding box. Enabling this option displays parameters you use to define the size and position of the region.
Eliminate Isolated Small Surface	Excludes small, unconnected data from the Mesh output.
Perform Transform Transform Mode	When **Perform Transform** is enabled, you can choose the which axes are the major, minor, and tertiary axes. The tool also centers the Mesh data at origin 0. This lets you align the part data however you want. **Transform Mode** is one of the following: • Minimal Alignment: The closest coordinate axes are arranged for alignment. • X > Y > Z Order • X > Z > Y Order • Y > X > Z Order • Y > Z > X Order • Z > X > Y Order • Z > Y > X Order
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Measurements

Measurement
X Origin **Y Origin** **Z Origin** These measurements return the X, Y, and Z position of the center of the fitted bounding box, respectively.
X Angle **Y Angle** **Z Angle** The angle of the fitted bounding box around the X, Y, and Z axis, respectively.
Width **Length**

Measurement

Height

The width, length, and height of the fitted bounding box.

Features

Type	Description
Point	A point representing the center of the fitted bounding box.



For more information on geometric features, see *Geometric Features* on page 227.

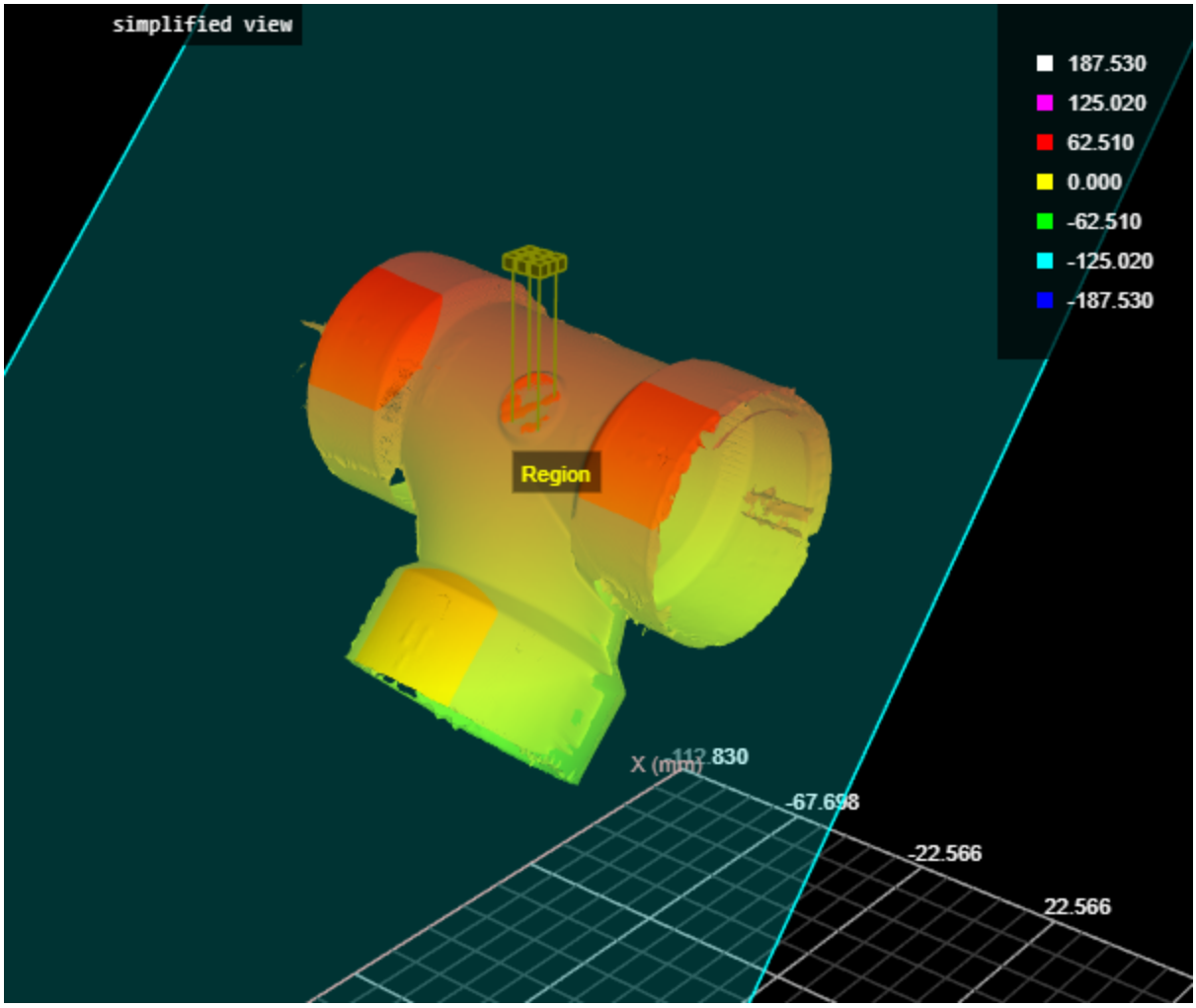
Data

Type	Description
Mesh	The Mesh data contained in the bounding box.

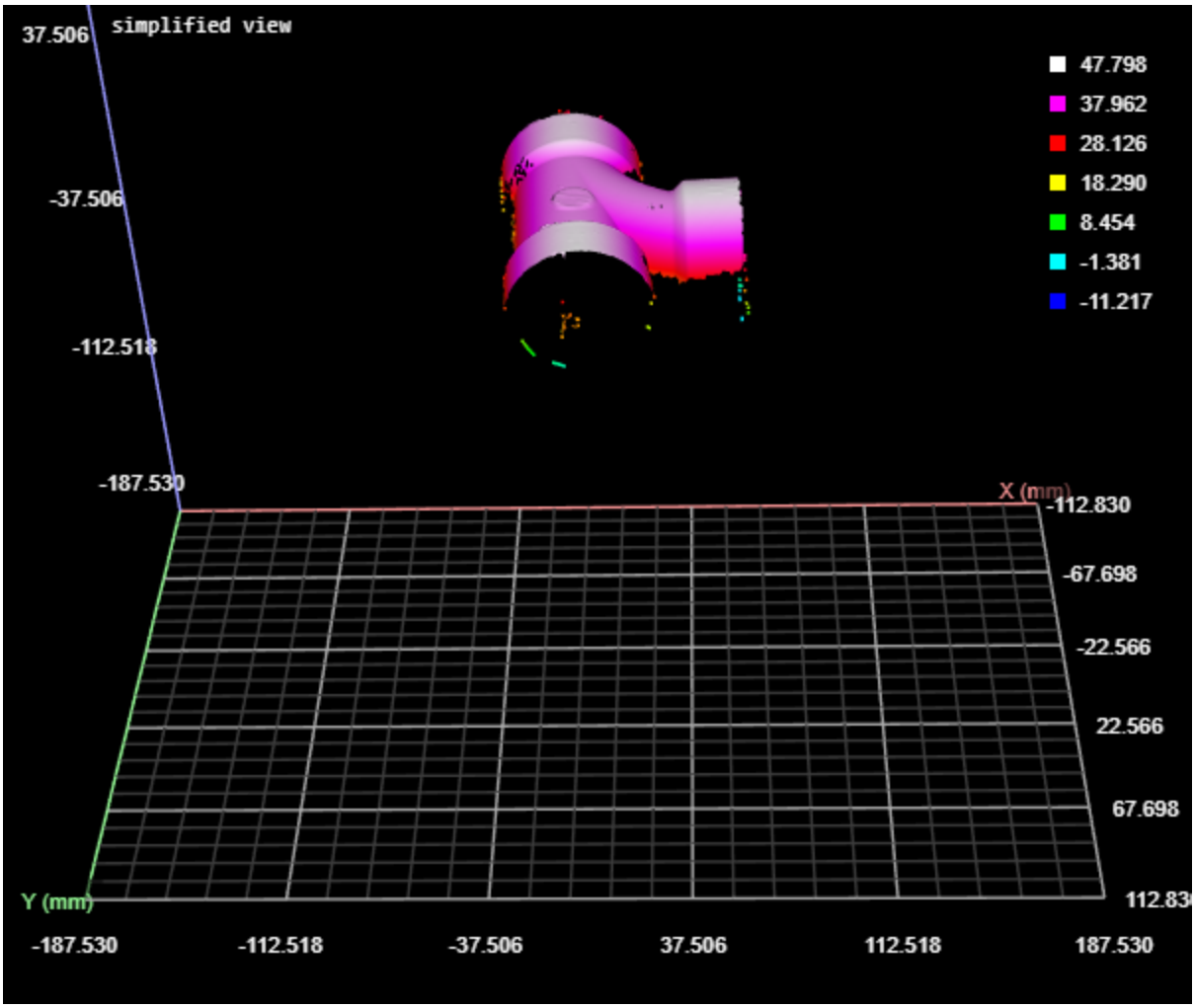
Mesh Plane

This tool is *only* intended for use with G2 sensors.

The Mesh Plane tool takes in Mesh scan data (produced by the Surface Mesh tool and some other Mesh tools) and returns measurements on the plane fitted within the region of interest, such as deviations of the data points relative to the plane. The tool also returns a Plane geometric feature that can be used as input by the Mesh Projection tool (see *Mesh Projection* on page 505). Finally, the tool returns front and back Surface data extracted from the plane: you can apply any built-in or custom GDK-based tools to the resulting data. This means that with 360-degree scan data, you can, for example, apply measurements to the sides or bottoms of your target, rather than just the top.



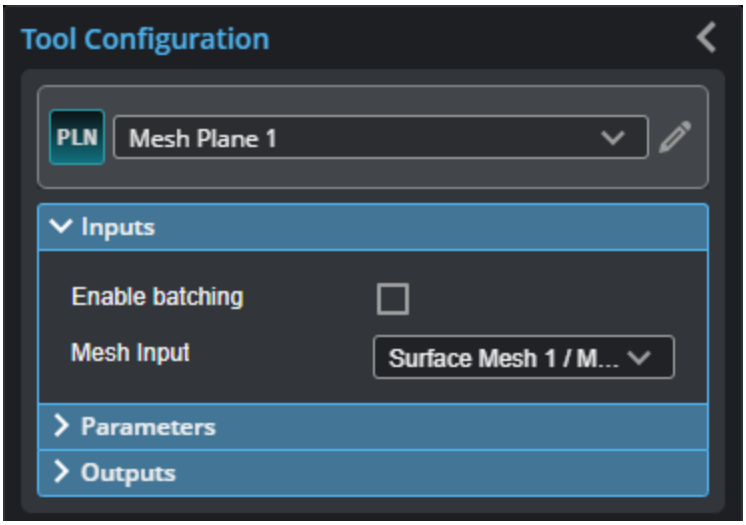
Mesh data with a region placed on a circular flat area. The plane fitted to the data in this region is shown in cyan.



The Front Surface data output is rotated by the plane's X, Y, and Z rotation.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

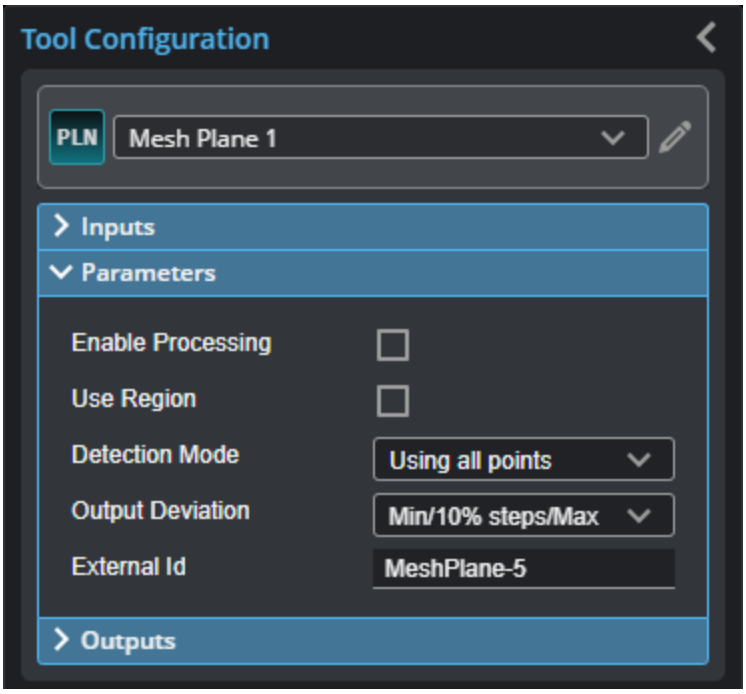


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Mesh Input	The data the tool applies measurements to or processes.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



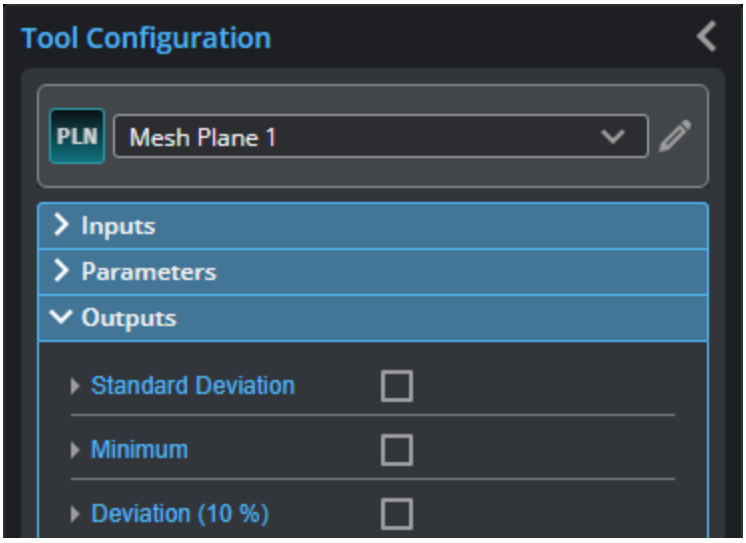
Parameters

Parameter	Description
Enable Processing	Enables processing.
Use Region	When enabled, displays region settings.
Region	Region settings.
Detection Mode	The plane detection mode. One of the following: **With Largest Area** **With Maximum Distance** **With Minimum Distance** Chooses the plane at the maximum or minimum distance in the region, respectively, from the 0 origin. Use these options when more than one plane fit is possible in the region. Works in conjunction with **Search Direction**. **Using all points** Uses all data points of the scan data in the region.
Search Direction	The search direction the tool will use to fit a plane. For example, when **Search Direction** is set to +Z, the tool starts searching from origin Z = 0 and moves along the positive Z axis. This parameter is only useful when **Plane Detection Mode** is set to one of the following: • With Largest Area • With Maximum Distance • With Minimum Distance

Parameter	Description
	The corresponding surface normals are taken into account in the processing so that the uninvolved points can be sorted out relatively quickly and safely. The fixed search angle is 45 degrees around the set direction. When **Search Direction** is set to **Input Direction**, the tool displays additional parameters: **Tilt Angle** and **Direction Angle**. **Tilt Angle** - The angle between the Z axis and the vector. **Direction Angle** - The vector is projected onto the XY plane and then rotated around the X axis. Specifically: X = sin(TiltAngle) * cos(DirectionAngle) Y = sin(TiltAngle) * sin(DirectionAngle) Z = cos(TiltAngle)
Output Deviation	Determines which deviations are output as measurements, which can be a combination of minimum and maximum, and a set of Deviation (x %) measurements (with the specified step between them). Can also be set so that no deviations are output. Use this to get a rough idea of the distribution of the deviation values (or a histogram of the deviations).
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Measurements

Measurement
Standard Deviation The standard deviation of the data points from the fitted plane.
Minimum **Maximum** The minimum and maximum error of the data points from the fitted plane, respectively.
Deviation (x%) Deviations of the data points from the fitted plane, sorted into stepped percentiles. You set number of steps using the **Output Deviation** parameter.

Features

Type	Description
Plane	A plane geometric feature.



For more information on geometric features, see *Geometric Features* on page 227.

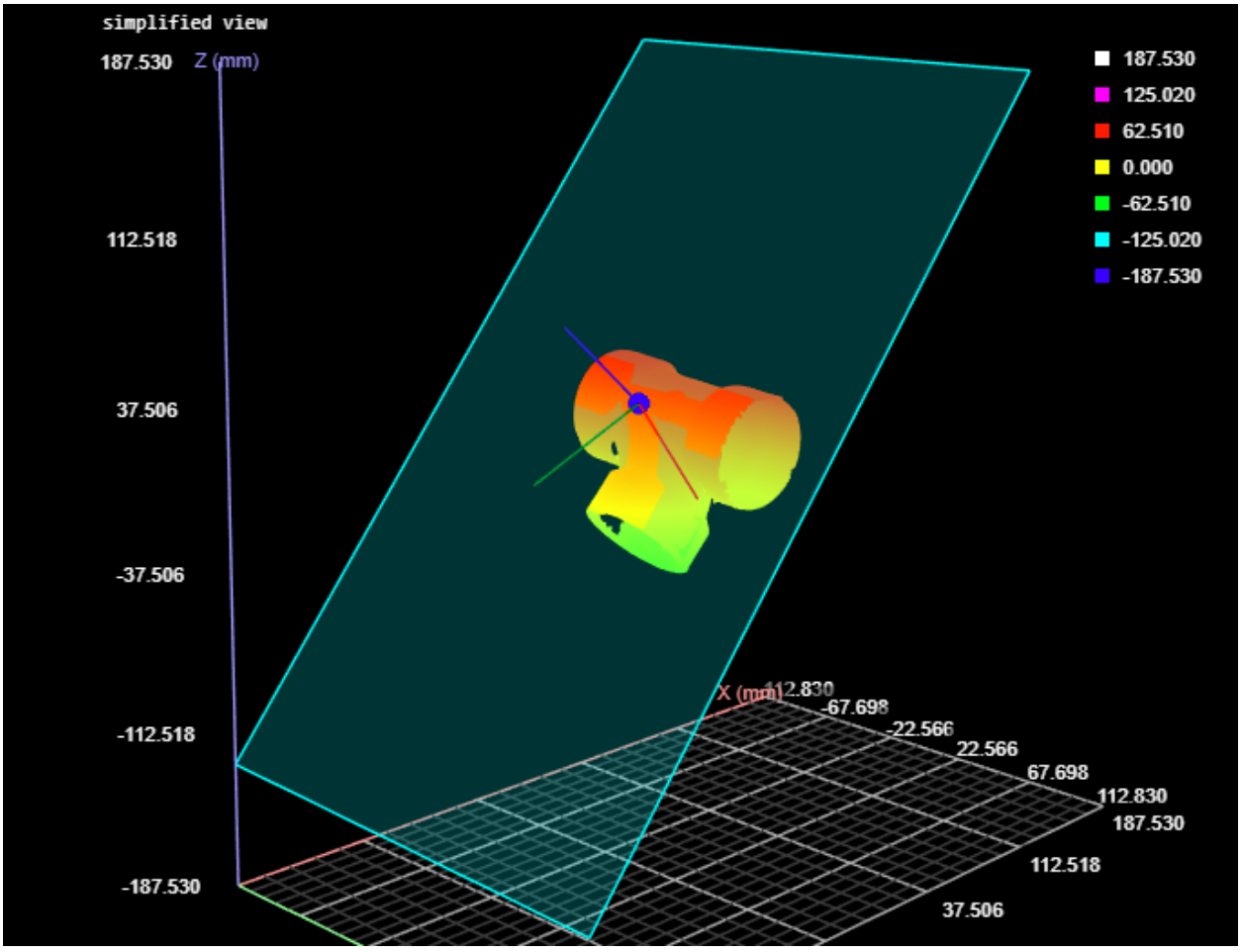
Data

Type	Description
Front Surface	Surface data representing the front of the meshed target.
Back Surface	Surface data representing the back of the meshed target.
Difference Surface	A Surface output that shows the fit error at each point in the height map.

Mesh Projection

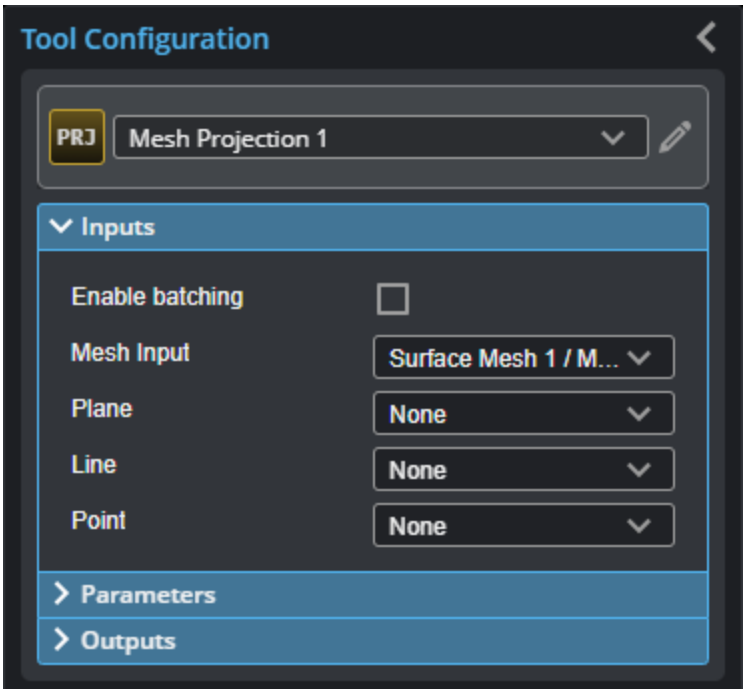
> This tool is *only* intended for use with G2 sensors.

The Mesh Projection tool takes in Mesh scan data (produced by the Surface Mesh tool) and extracts Surface data. The tool can optionally take plane, line, or point geometric features to perform transformations on the output surface data (if no geometric features are used as inputs, the surface parallel to the XY plane is output), or you can manually apply fixed transformation. You can then apply any built-in or custom GDK-based Surface tool to the resulting Surface data. This means that with 360-degree scan data, you can, for example, apply measurements to the sides or bottoms of your target, rather than just the top.



Inputs

You configure the tool's inputs in the expandable **Inputs** section.

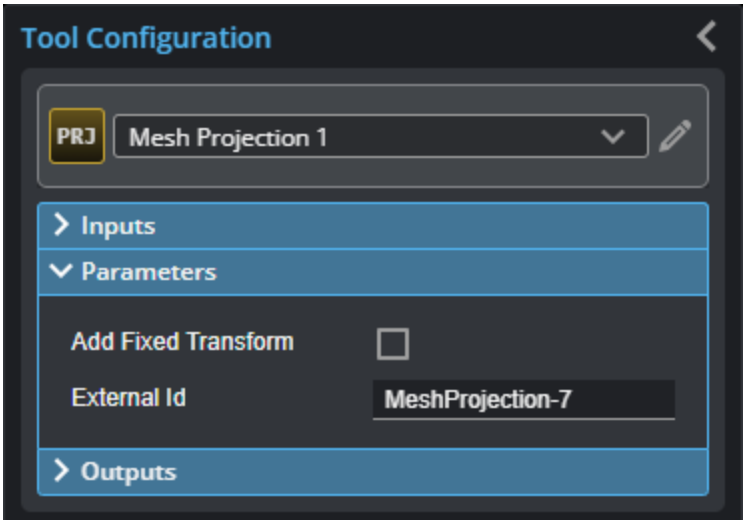


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with a maximum of two elements. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with a maximum of two elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Mesh Input	The data the tool applies measurements to or processes.
Plane	The tool aligns the XY plane to the selected plane geometric feature in the output Surface data.
Line	The tool aligns the X axis to the selected line geometric feature in the output Surface data.
Point	The tool uses the selected point geometric feature the origin in the output Surface data.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

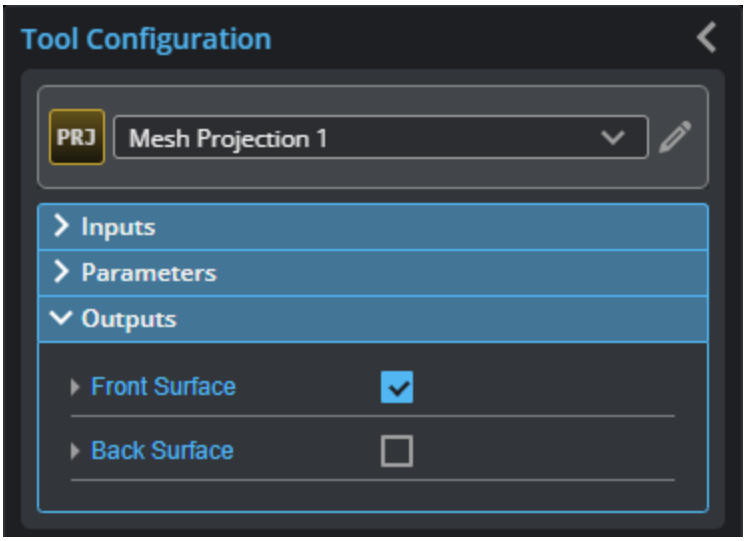


Parameters

Parameter	Description
Add Fixed Transform	When this parameter is enabled, you can provided fixed X, Y, and Z offsets, as well as X, Y, and Z angles, which the tool uses in the output Surface data.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Data

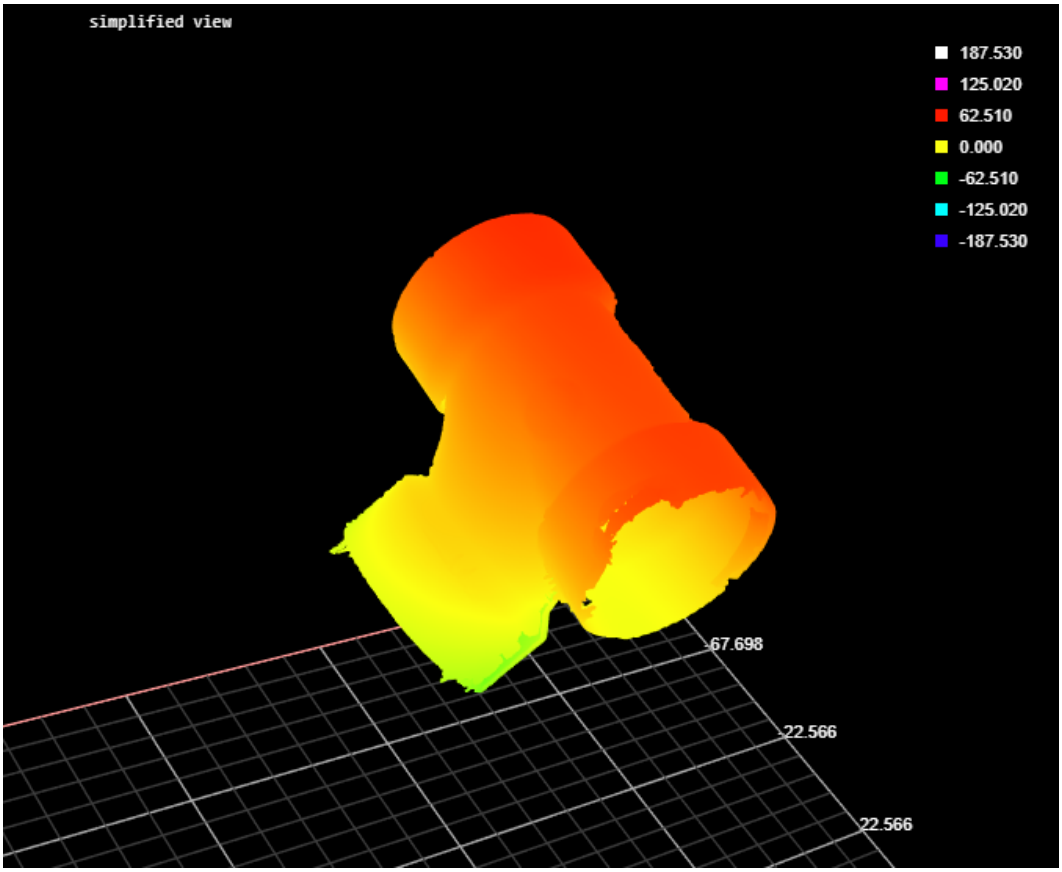
Type	Description
Front Surface	Surface data representing the front of the meshed target.
Back Surface	Surface data representing the back of the meshed target.

Mesh Template Matching

	This tool is *only* intended for use with G2 sensors.

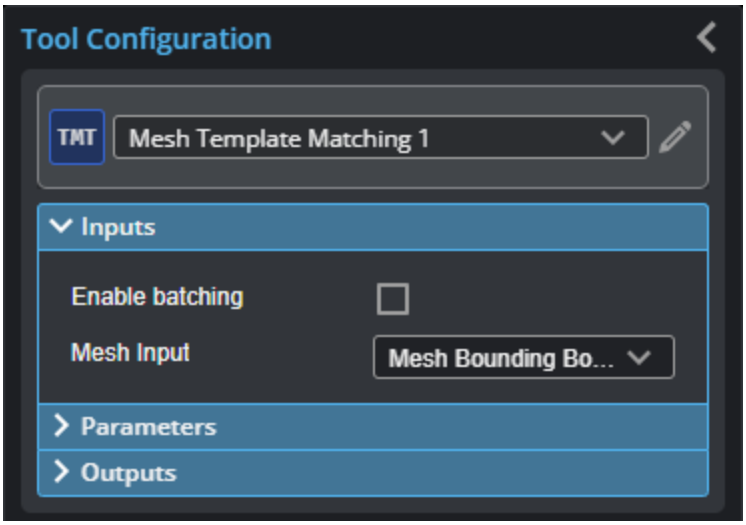
	This tool is not supported on A and B revision Gocator 2100 and 2300 sensors that are not accelerated (either by a PC-based application or by GoMax). The tool is supported in emulator scenarios.

The Mesh Template Matching tool takes in Mesh scan data (produced by the Surface Mesh tool) and a template you previously defined based on a "golden part" (itself created using the Mesh Template Matching tool). The tool returns measurements related to the position and orientation of the scan data relative to the template, such as offsets and rotations, as well as standard deviations between the scan data and the template. The tool can also output Mesh scan data.



Inputs

You configure the tool's inputs in the expandable **Inputs** section.

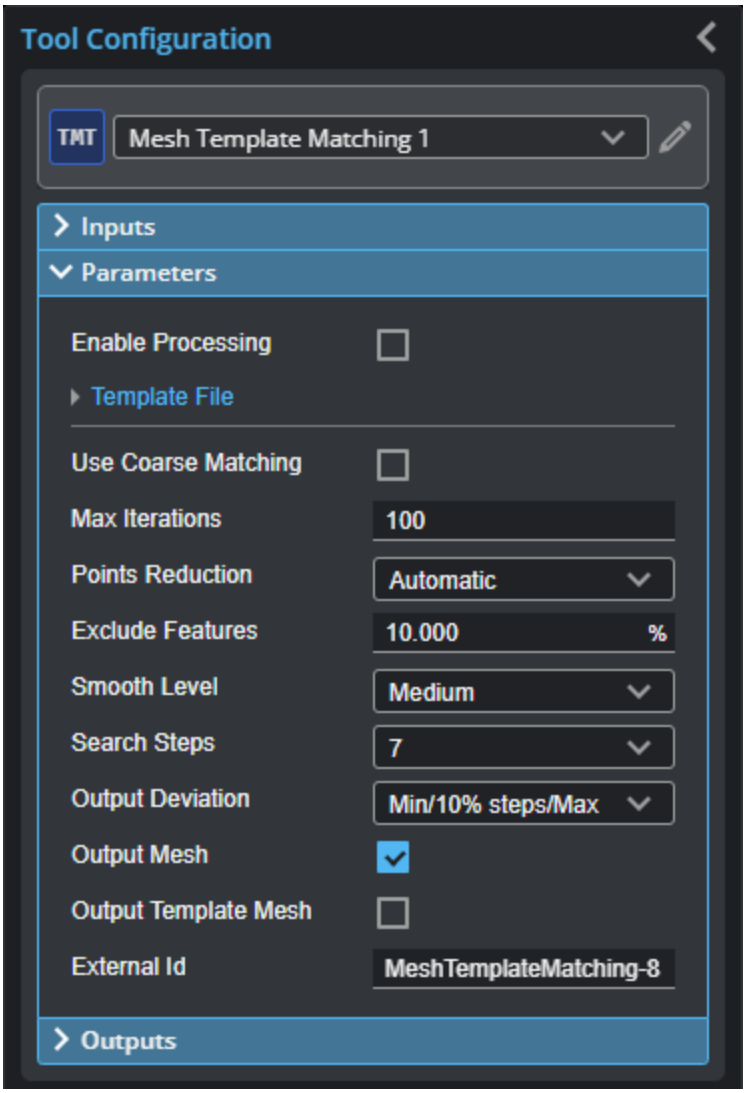


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Mesh Input	The data the tool applies measurements to or processes.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



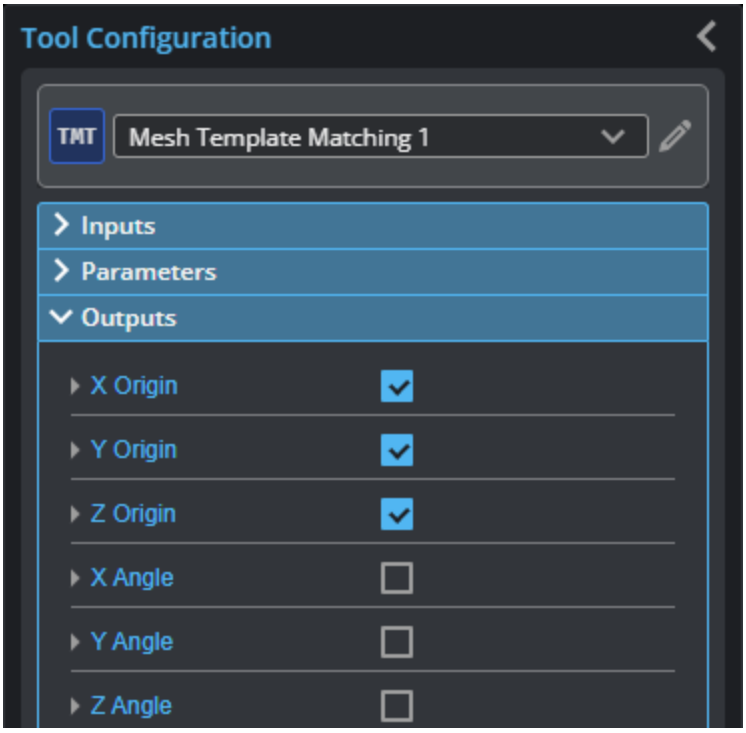
Parameters

Parameter	Description
Enable Processing	When this option is enabled, the tool compares the Mesh data to the loaded template.
Template File	Expanding section containing template-related parameters.
Operation Template Name Template File	The tool's operation mode. One of the following: **Normal**: When Enable Processing is enabled, the tool compares the Mesh scan data and the loaded template. **Load**: Displays a list of Mesh template files you can load (in the **Template File** drop-down). Supports the following three formats: • BCD: Internally used by the tool. To reduce the conversion processing time, a BCD file with the same name is automatically generated if the data is imported with a different format.

Parameter	Description
	• OBJ: A format for defining the 3D geometry for the surface of one or more objects. • STL: A file format commonly used for 3D printing and computer-aided design (CAD). **Save**: Saves the current frame of Mesh scan data as a template (in C:\GoTools\Mesh Template Matching\) in BCD format. Type the name of the template in the **File Name** field, and then check **Enable Processing** to save the file. A user message is displayed if the template is saved successfully. Note this option is only available if there is no template currently loaded in this tool. **Delete**: Deletes the template file you select in the **Template File** field. **Refresh**: Refreshes the template file list. **Remove**: Unload an existing template to make the Save option visible. So that user is aware that just the current input mesh data will be declared and saved as a template, when selecting the Save option.
Use Course Matching	Enable this parameter when the body surfaces to be matched are not symmetrical, and the original orientation and position deviate significantly from the measured data set.
Max Iterations	The maximum number of iterations the tool uses to match the Mesh scan data with the template. This parameter can be used to speed up the optimization process to some degree. Typically, leave this at the default value.
Points Reduction	Controls the number of points used in the matching process, which can improve processing time.
Exclude Features	Use this when there are high or low features on the part that should not be included in the matching. For example, at 10%, the tool excludes 10% of the points with maximum or minimum deviation from the matching process.
Smooth Level	The amount of smoothing the tool applies. LMI recommends leaving this setting at its default.
Search Steps	Determines the neighborhood level in which to search for connection point pairs.
Output Deviation	Determines which deviations are output as measurements, which can be a combination of minimum and maximum, and a set of Deviation (x %) measurements (with the specified step between them). Can also be set so that no deviations are output. Use this to get a rough idea of the distribution of the deviation values (or a histogram of the deviations).
Output Mesh	Outputs the transformed mesh.
Output Template Mesh	Outputs a mesh template.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Measurements

Measurement
X Origin
Y Origin
Z Origin
X Angle
Y Angle
Z Angle

Measurement
These measurements represent the transform matrix that matches the input mesh against the template.
Standard Deviation The standard deviation of pairs of matching points.
Minimum **Maximum** The minimum and maximum deviation of pairs of matching points.
Deviation (x%) The deviations of pairs of matching points, sorted into stepped percentiles. You set the number of steps using the **Output Deviation** parameter.

Data

Type	Description
Mesh	The transformed Mesh. Only listed if the **Output Mesh** parameter is enabled.
Mesh Template	A template Mesh. Only listed if the **Output Template Mesh** parameter is enabled.

Profile Tools

This section describes the profile measurement tools available in Gocator sensors.

Profile measurement tools can be used on sections extracted from Surface data. For more information on sections, see *Surface Section* on page 943.

Profile 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).

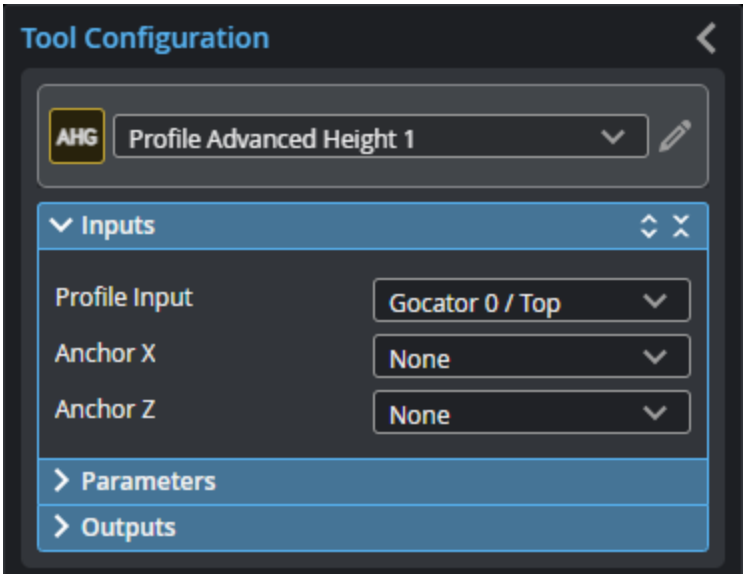
> All instances of the Advanced Height tool share the same template file set in **File**. For this reason, you must be careful when editing or removing template files shared by another instance of the tool.

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).

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



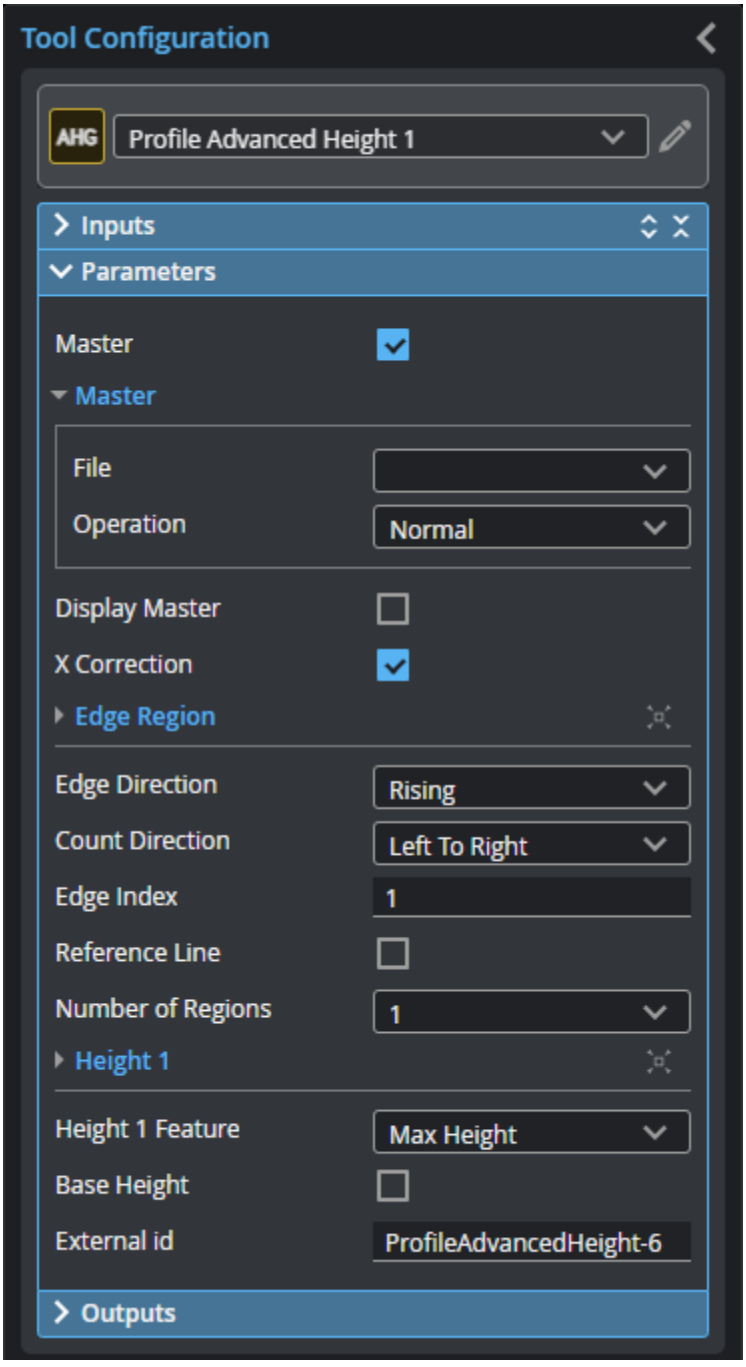
> To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Profile Input	The data the tool applies measurements to or processes.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

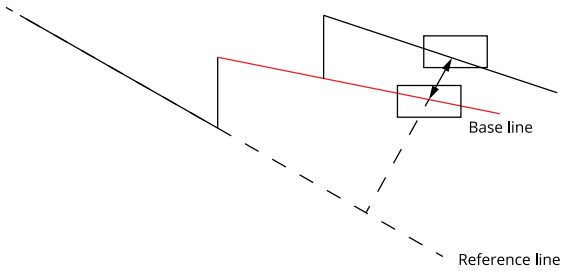
Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Master	Toggles a set of settings related to master comparison. For more information, see *Master Comparison* on the next page.

Parameter	Description
Reference Line	Toggles a set of settings related to the reference line. For more information, see *Reference Line* on the next page.
Number of Regions	Sets the number of height region measurements the tool returns. For each height region, the tool displays a **Height {n}** section containing the region's position and size. The tool also displays a **Height{n} Feature** drop-down that lets you select the type of feature for that height region.
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.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Master Comparison

When you check the **Master** option, the tool displays several additional settings and disables measurement anchoring from other tools.

The **File** and **Operation** parameters are in an expandable **Master** section.

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. Use this when the master does not exist. • **Load**: Loads the master file selected in **File**. • **Save**: Saves the current profile as the master. If the file already exists, the master is overwritten with the new data.

Parameter	Description
	• **Delete**: Deletes the master file selected in **File**. • **Refresh**: Refreshes the list of files 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. Only displayed if **Master** is enabled. For more information, see *X Correction* below.

X Correction

When you check the **Master** option and enable **X Correction**, the tool displays several additional settings.

X Correction Parameters

Parameter	Description
Edge Region	Lets you set the edge 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

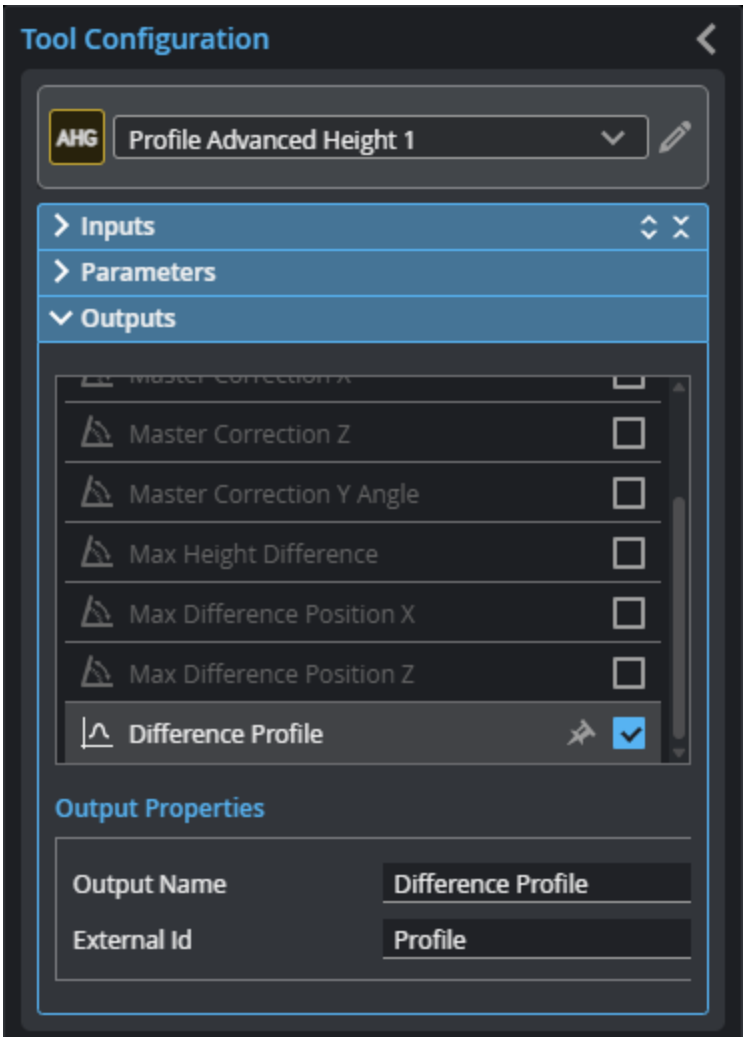
When you enable **Reference Line**, the tool displays several additional settings. The reference line is used to set the measurement direction (perpendicular to the reference line).

Reference Line Parameters

Parameter	Description
Line Region	The number of line regions the tool uses.
Line {n}	Contains settings that let you edit the size and position of the line's region. You can also edit this region in the data viewer.
Fitting Method	Indicates the fitting method the tool uses. One of the following: **Simple** or **Robust**.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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

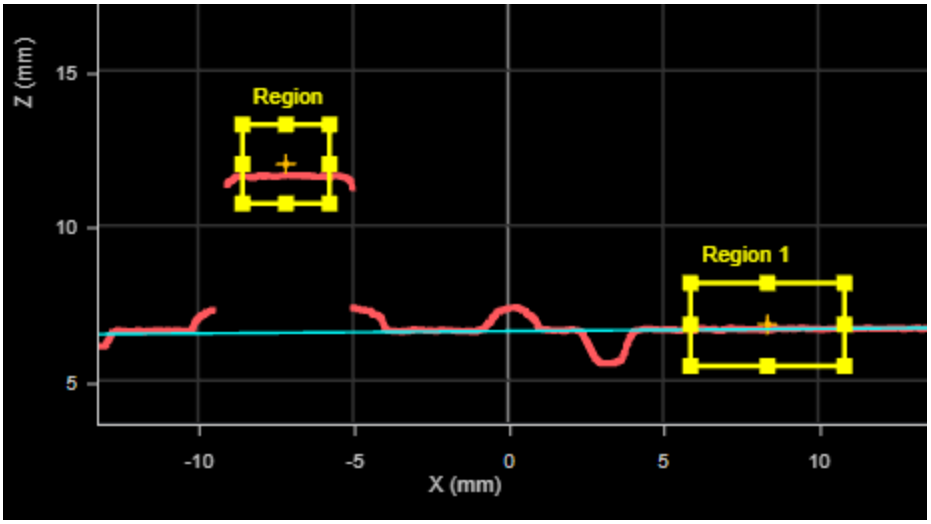
Measurement
Master Correction Z **Master Correction Y Angle** The amount of correction applied to the profile with respect to the master.
Max Height Difference The maximum height difference.
Max Difference Position X **Max Difference Position Z** The X and Z positions of the maximum height difference.

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.

Profile Area

The Area tool determines the cross-sectional area within a region.

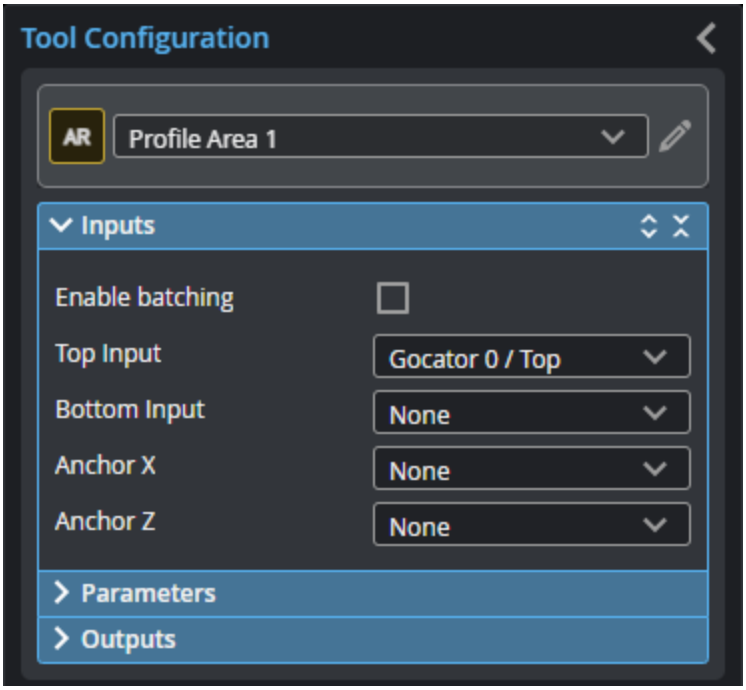


Profile Area tool with baseline set to the best-fitted line in Region 1

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



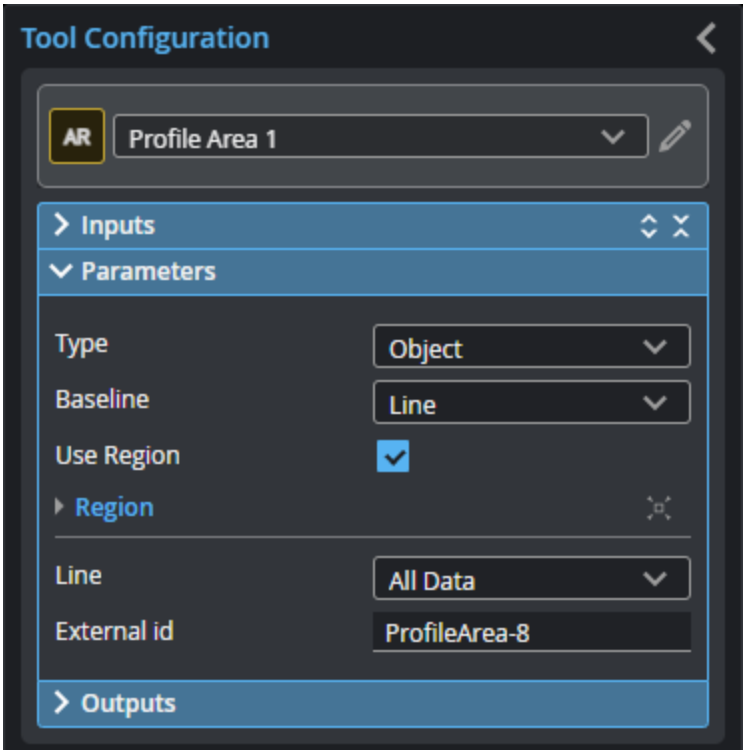
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Top Input	The data the tool applies measurements to or processes.
Bottom Input	This tool can optionally take a second, bottom input.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



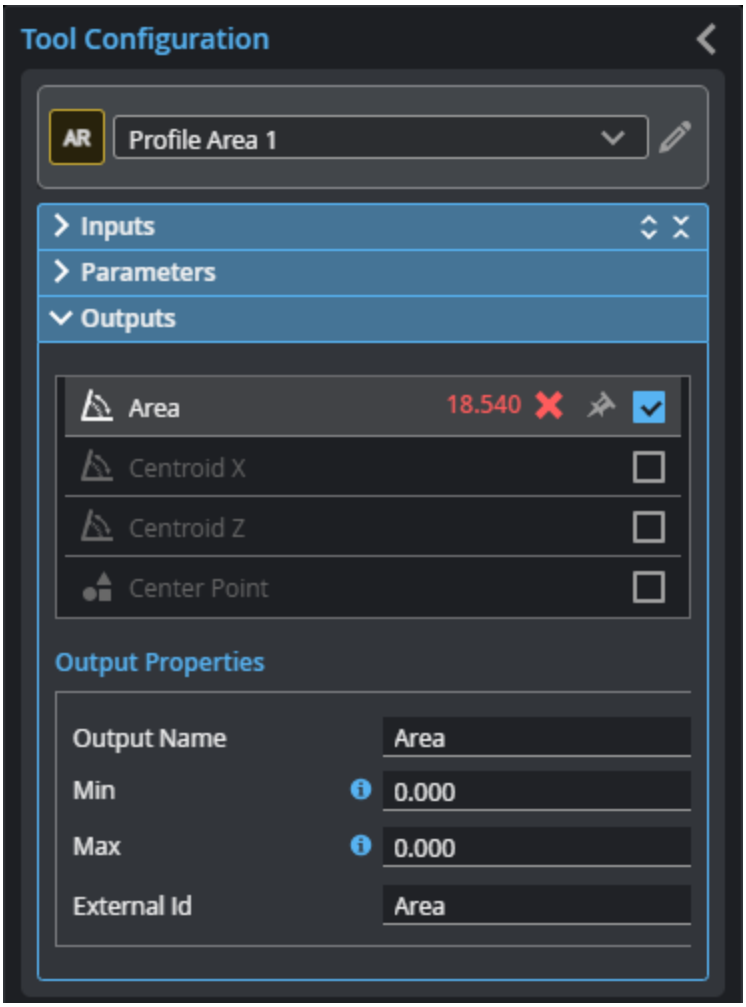
Parameters

Parameter	Description
Type	**Object** area type is for convex shapes above the baseline. Regions below the

Parameter	Description
	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 227 for more information on fit lines. When this parameter is set to **X-Axis**, the baseline is set to z = 0.
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
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. The tool uses the data in those regions to fit a line. **All Data**: The tool uses all of the data in the scan data. For more information on regions, see *Regions* on page 214). For more information on fit lines, see *Fit Lines* on page 227.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

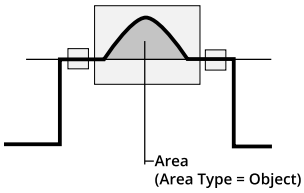


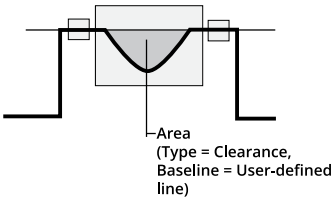
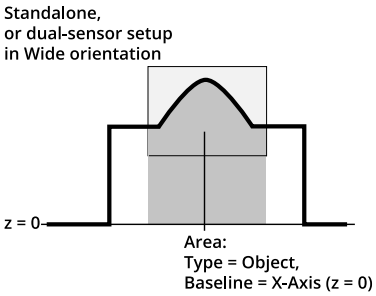
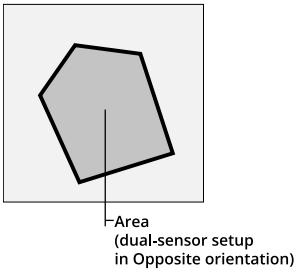
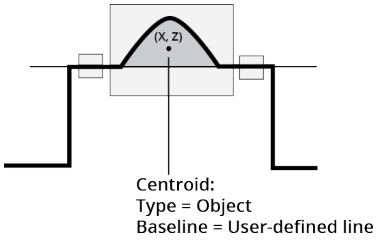
To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	Illustration
Area Measures the cross-sectional area within a region that is above or below a fitted baseline.	

Measurement	Illustration
	
	
	
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

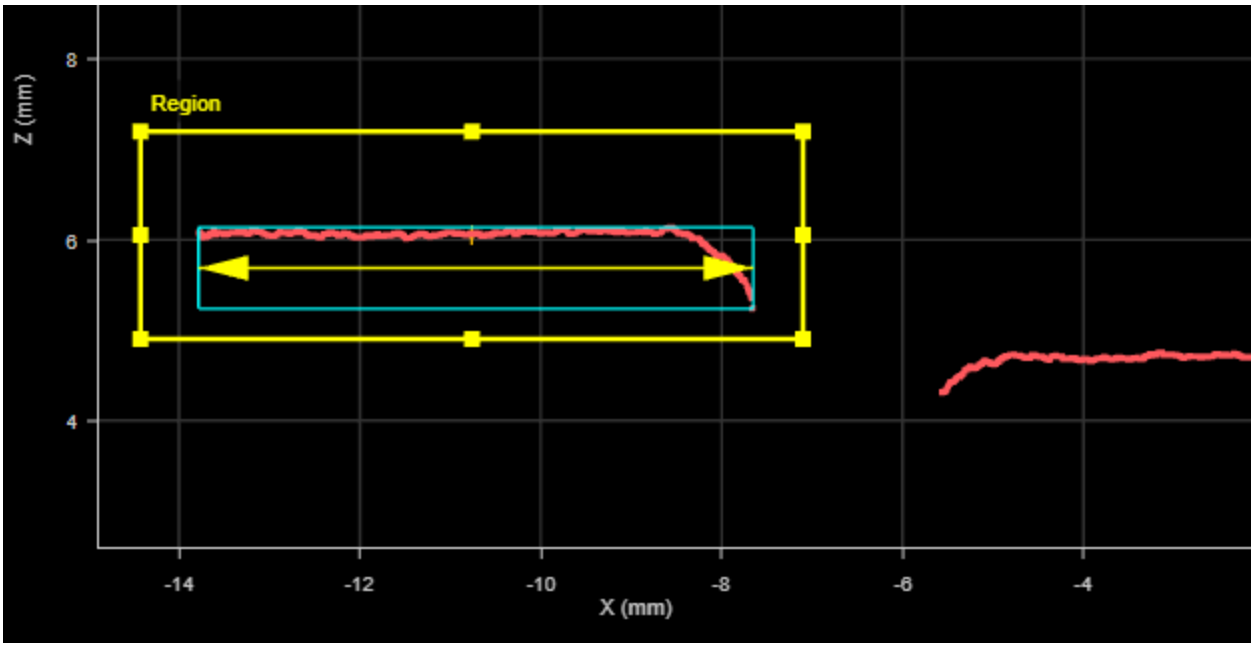
Type	Description
Center Point	The center point of the area.

For more information on geometric features, see *Geometric Features* on page 227.

Profile 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.).

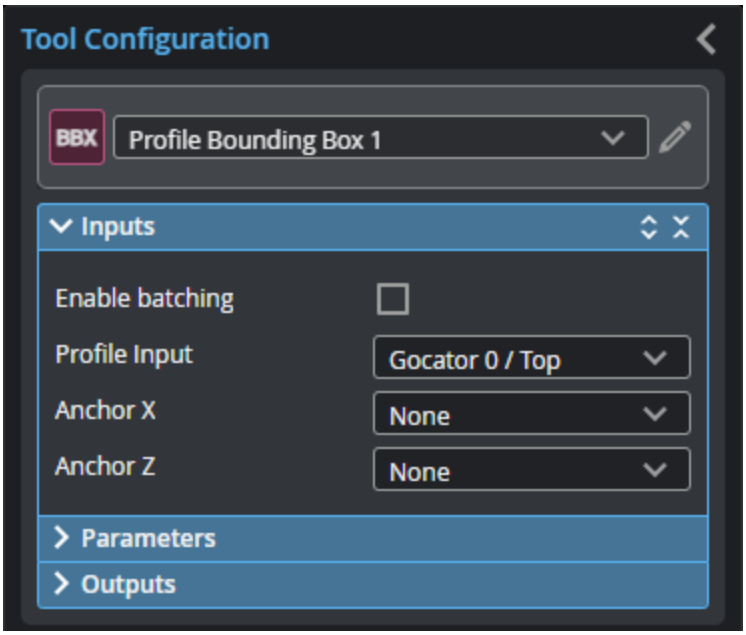
The bounding box provides the absolute position from which the Position centroids tools are referenced.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



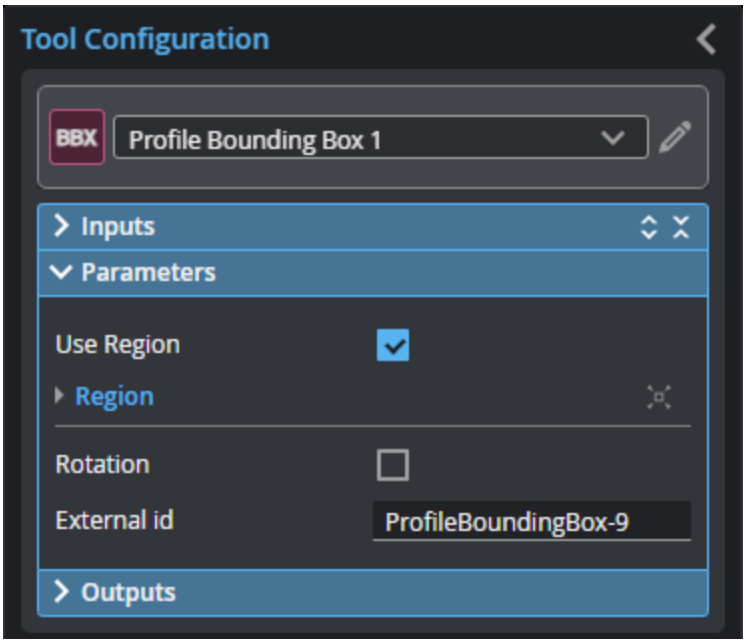
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with a maximum of two elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. If **Enable Batching** is *disabled* and the passed array contains more than two elements, GoPxL displays an error. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

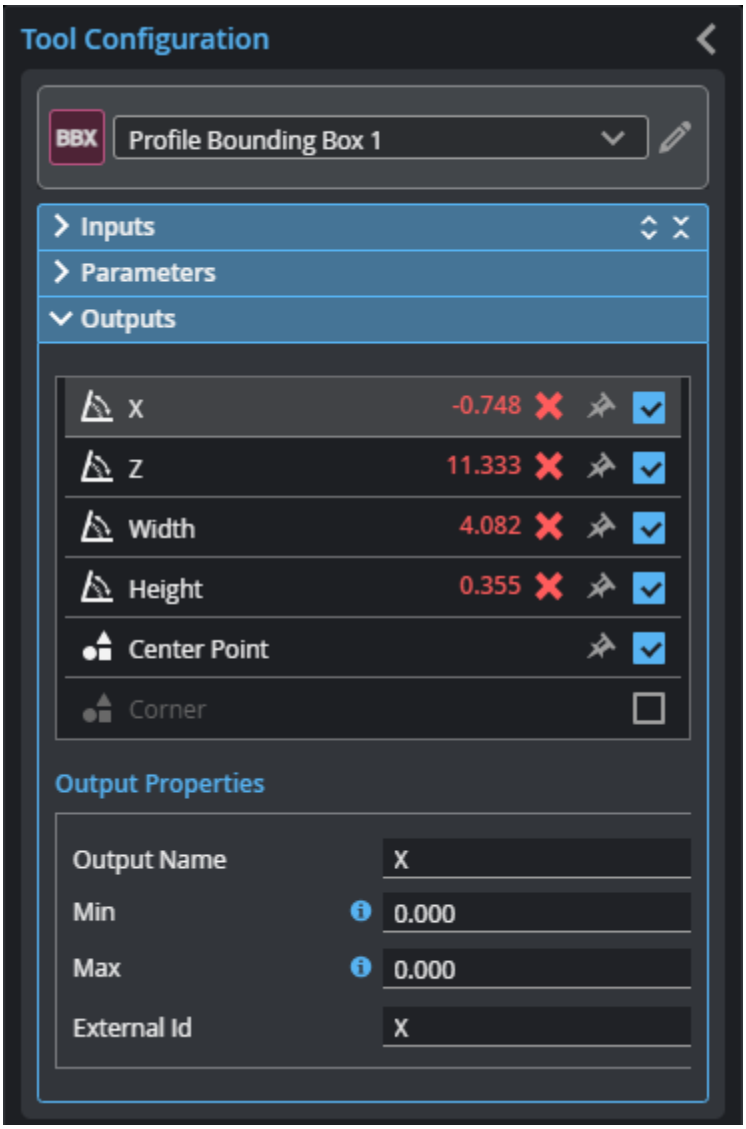


Parameters

Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
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. When enabled, allows rotation of the bounding box. Only displayed if **Use Region** is enabled.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

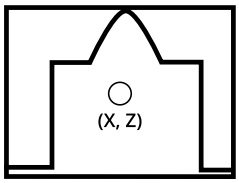
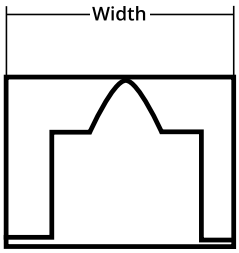
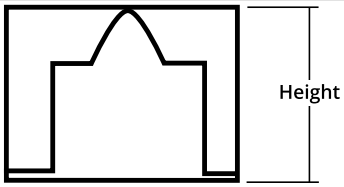


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.	
Angle Determines the angle of the longer side of the bounding box around the Y axis, relative to the X axis. This measurement is only available if the **Rotation** parameter is enabled.	

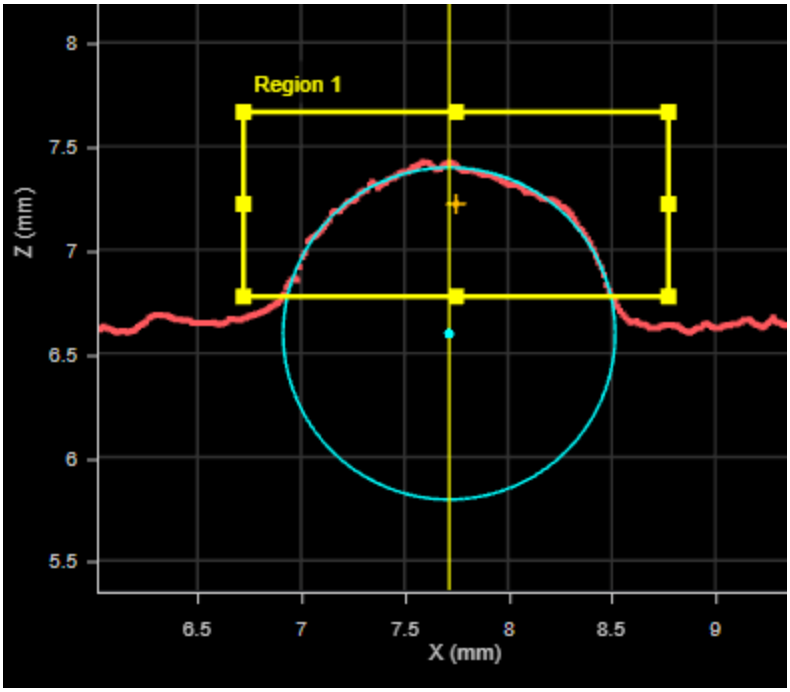
Features

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 227.

Profile Circle

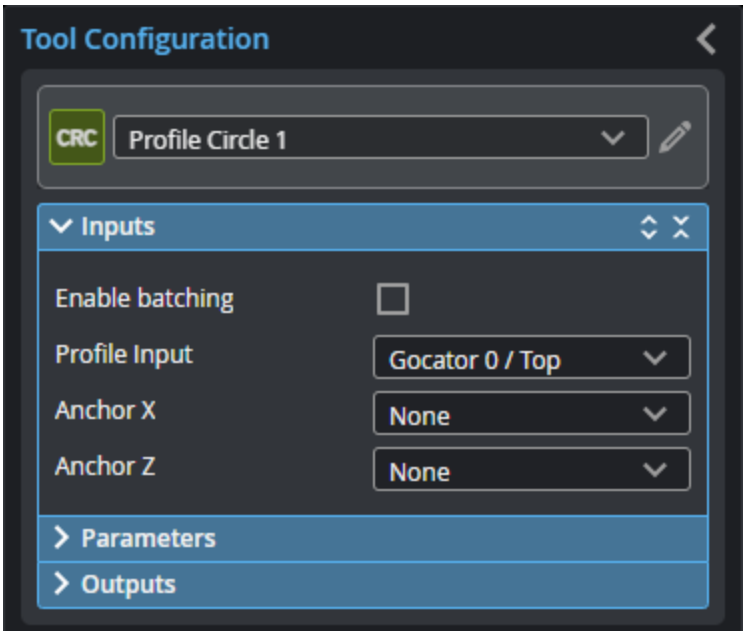
The Circle tool calculates the best-fit circle for a profile and returns measurements of various characteristics of the circle. The tool uses the iterative robust least squares fitting method.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



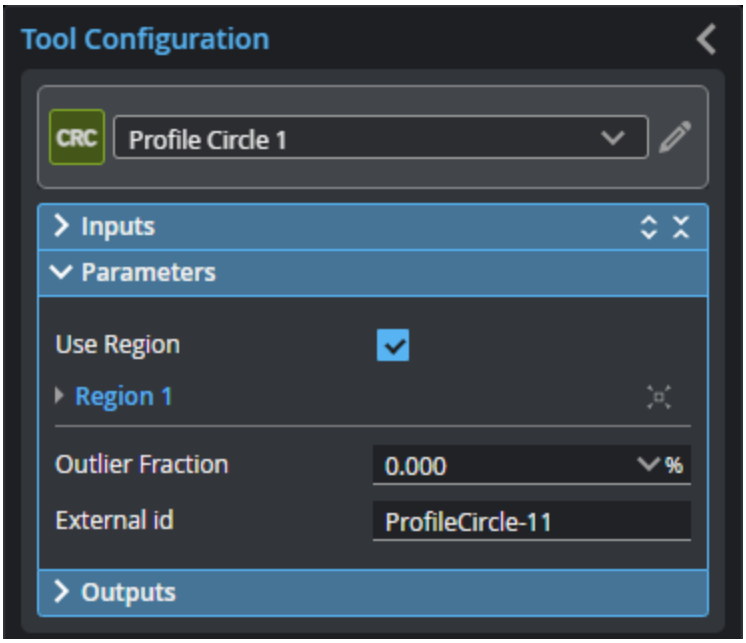
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

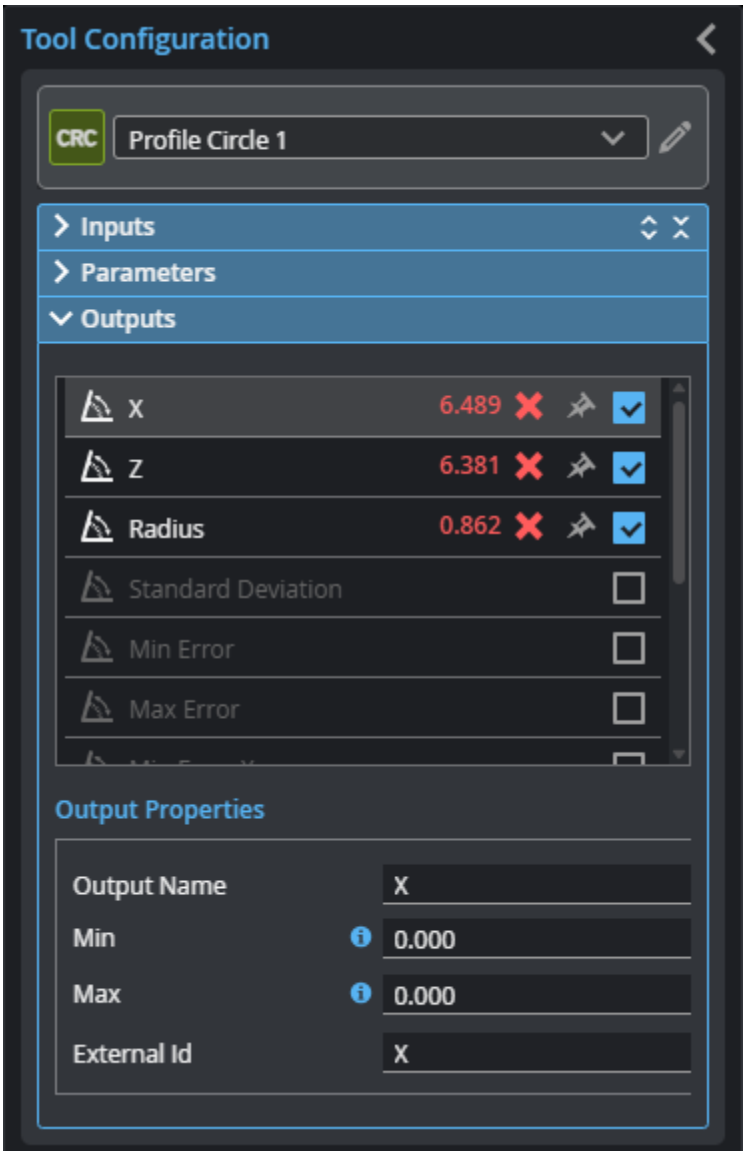


Parameters

Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region {n}	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
Outlier Fraction	The percentage of the profile points that are considered noise. A range from 0% to 80%. For example, suppose a profile contains `m` points. If **Outlier Fraction** is not equal to 0, then `Outlier Fraction*m` points will be considered noises and be removed iteratively.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

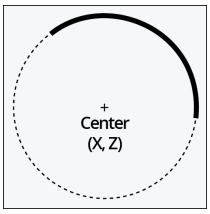
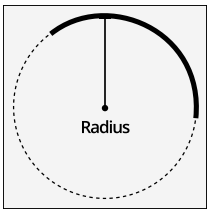


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	Illustration
X Finds the circle center position in the X axis.	
Z Finds the circle center position in the Z axis.	
Radius Measures the radius of the circle.	
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

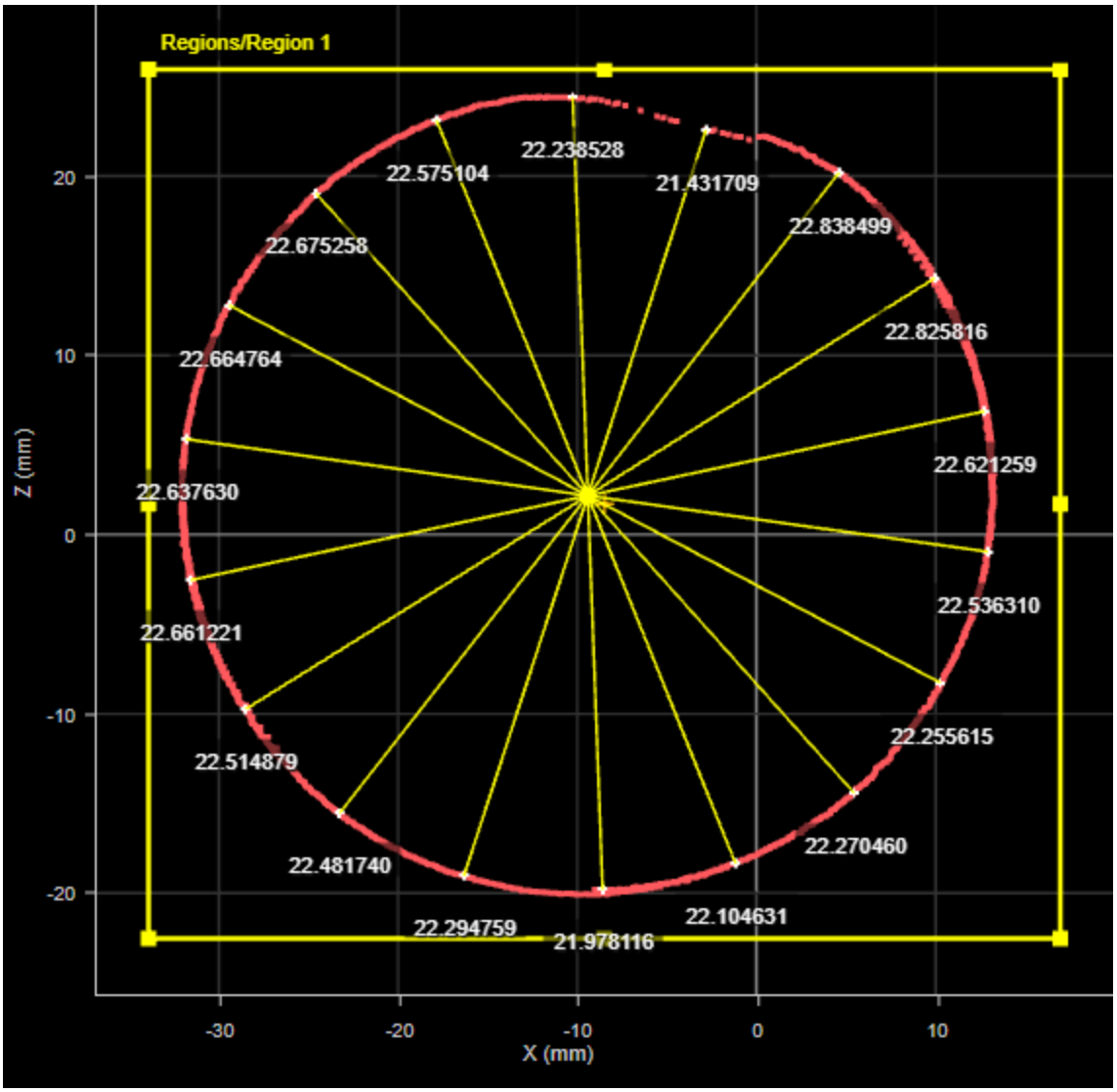
Type	Description
Point	The center point of the fitted circle.
Circle	The fitted circle.

For more information on geometric features, see *Geometric Features* on page 227.

Profile Circle Radii

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. The center point comes from either the center of a bounding box or a point geometric feature from another tool. The tool also provides settings to compensate for missing data and for rough surfaces or noise.

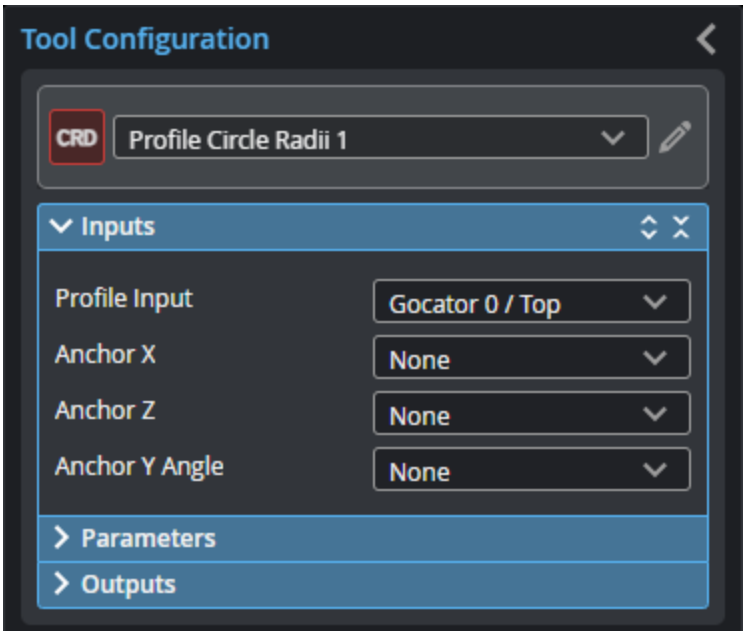
For example, in the following scan of an exhaust pipe by a four-sensor system, the tool is showing a shorter radius measurement at roughly 70 degrees that indicates a dent in the pipe.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



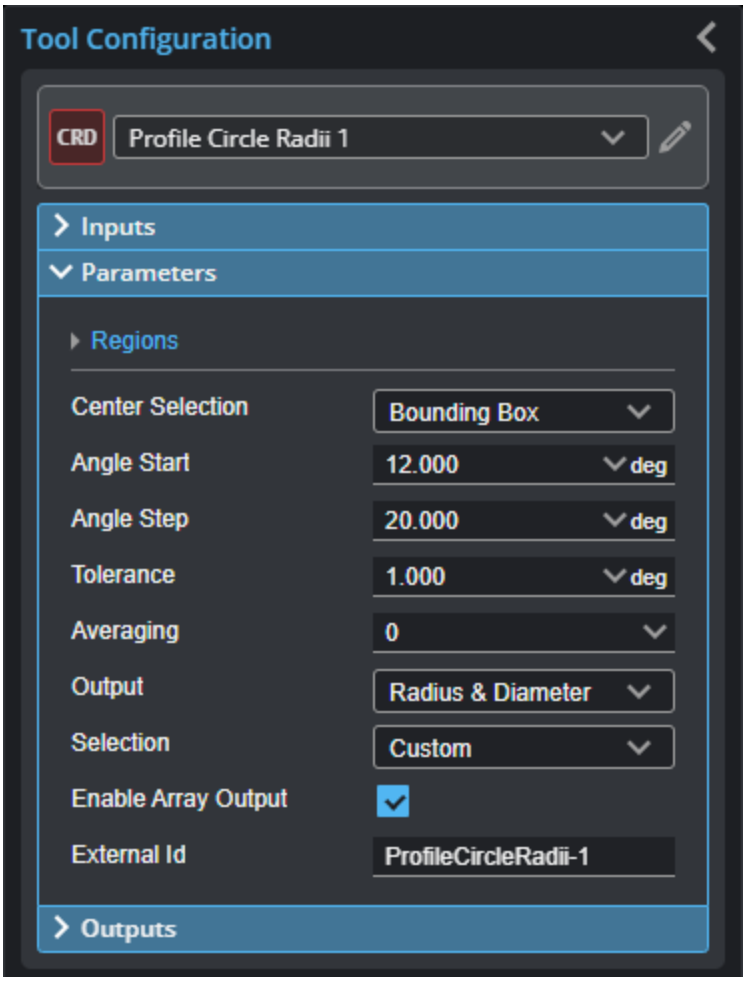
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Profile Input	The data the tool applies measurements to or processes.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Y Angle	The Y angle measurement of another tool that this tool uses as a rotational anchor. Rotational anchors are optional.
Center	The point geometric feature output by 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**.

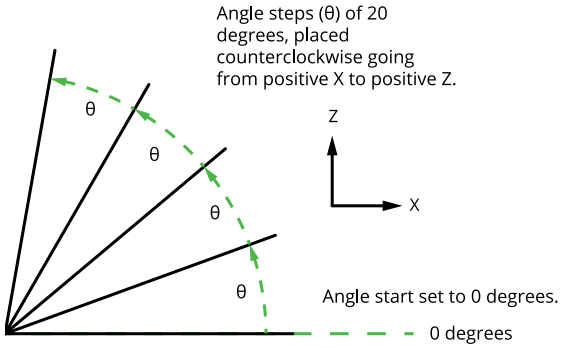
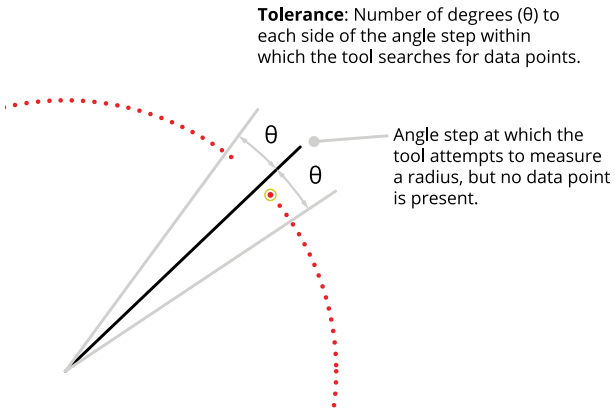
Parameters

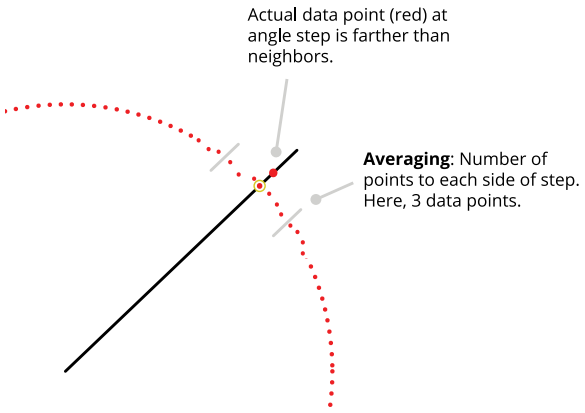
You configure the tool's parameters in the expandable **Parameters** section.



Parameters

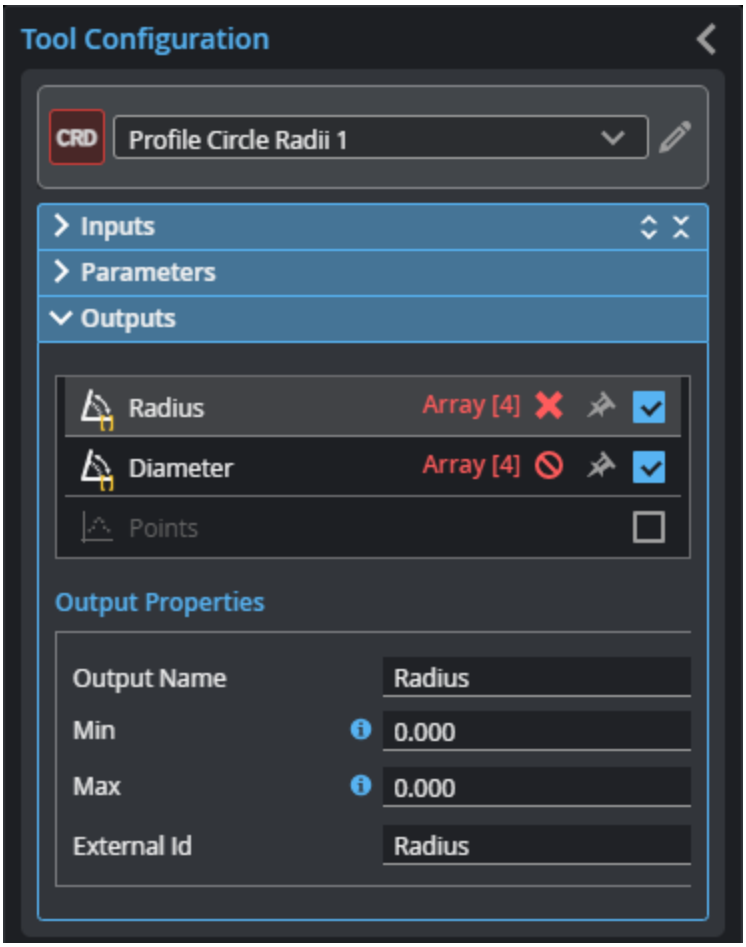
Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
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. If regions are enabled (**Enable** in the **Regions** expander is checked), the tool places a bounding box only around the data in the region. If **Enable** is unchecked, 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.

Parameter	Description
	Feature Input – A point geometric feature provided by another tool, such as the center point from a Circle tool.
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).
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.

Parameter	Description
	 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 the radius, diameter, or both for each step.
Selection	Lets you quickly enable or disable all measurements.
Enable Array Output	When this parameter is checked, the tool outputs arrays containing the radius and diameter measurements. If the parameter is disabled, measurements are output for the radius or diameter at each step.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



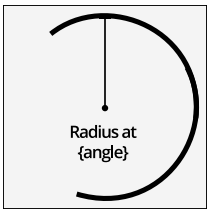
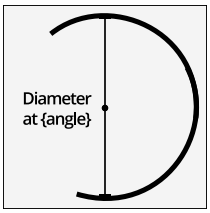
To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

If you check **Enable Array Output** in the tool's parameters, the tool outputs radius and diameter measurements in arrays, called Radius and Diameter, respectively. Otherwise, individual measurements are output, such as "Radius at 90.000," "Radius at 180," and so on.

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.

Profile Closed Area

The Closed Area tool determines the cross-sectional area within a region using point cloud data (that is, with **Enable uniform spacing** disabled in the **Acquire** page > **Scan mode** panel) from a dual- or multi-sensor system.

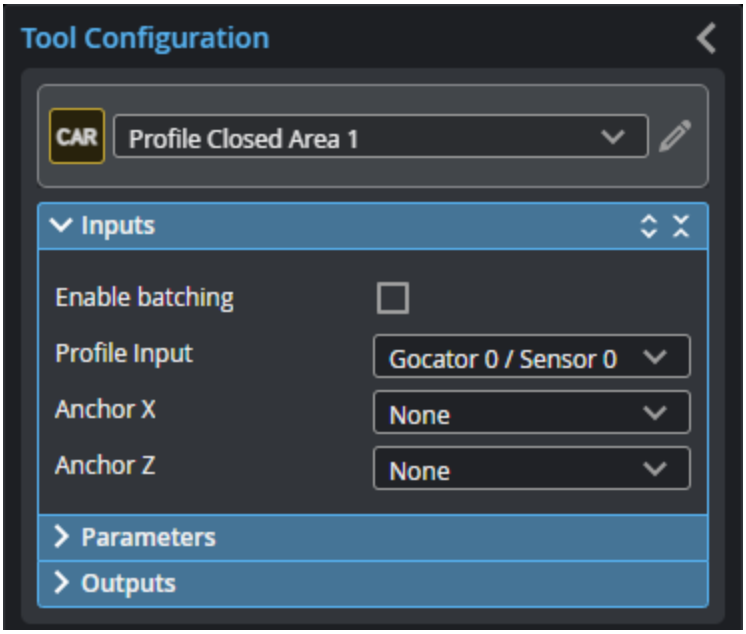
The tool is intended for use with roughly circular shaped profiles, or profiles that do not contain excessive concavity. The tool renders a polygon corresponding to the profile in the data viewer. Use this polygon to decide whether the tool can correctly calculate an acceptable representation of the profile. Minor gaps in the profile are permitted; the size of these gaps is configurable.

Two methods are provided to calculate the area: Sector and Contour. For more information, see *Method* on page 546.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

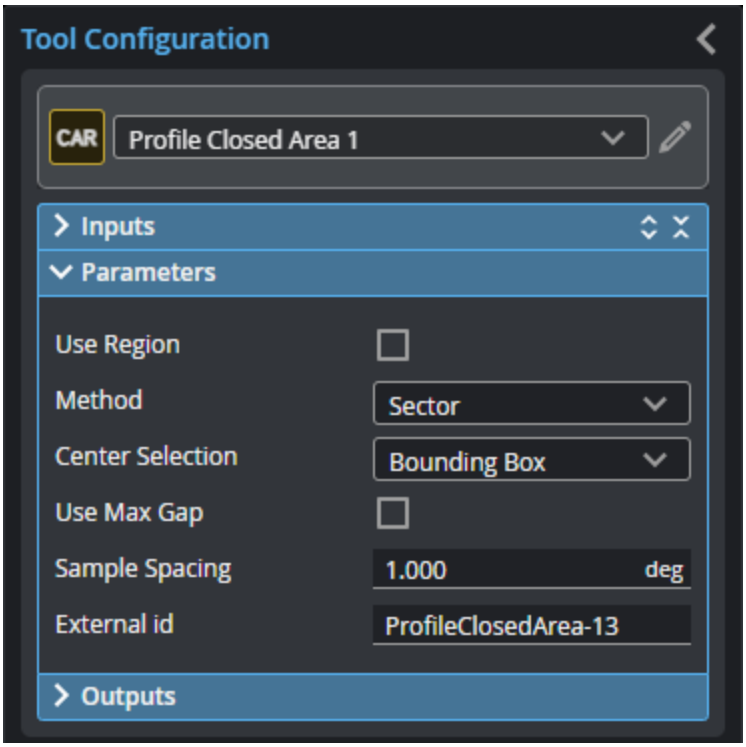
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array,

Name	Description
	other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

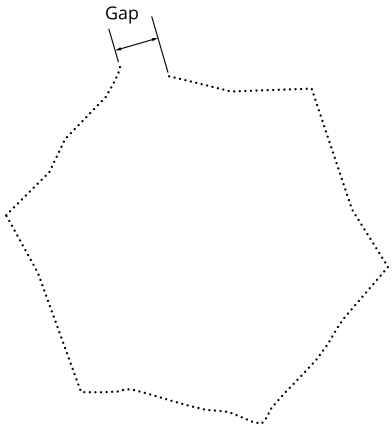
Parameters

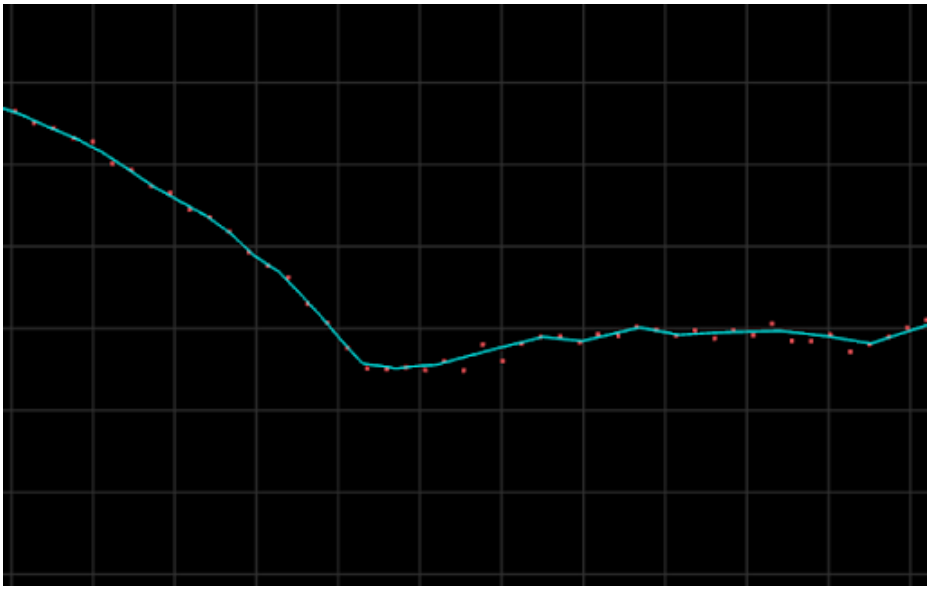
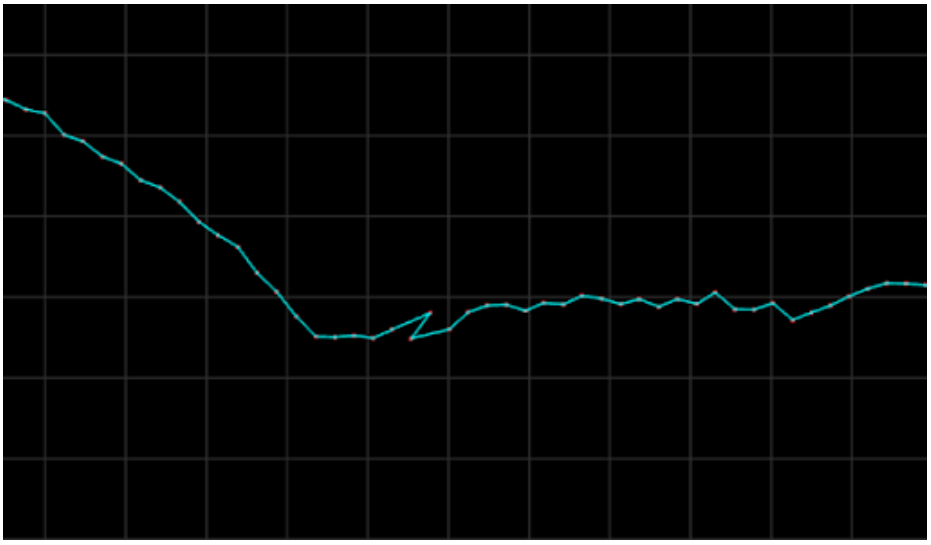
You configure the tool's parameters in the expandable **Parameters** section.



Parameters

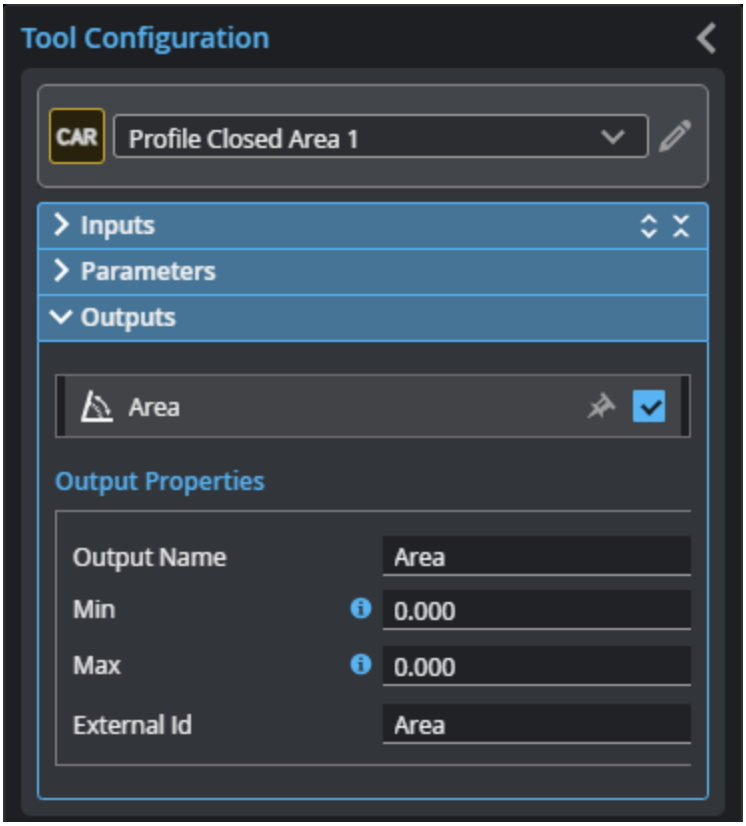
Parameter	Description
Use Region	Indicates whether the tool uses a user-defined region. If this option is not checked, the tool uses data from the entire active area.
Region	The region to which the tool's measurements will apply. For more information, see

Parameter	Description
	Regions on page 214.
Method	One of the following: Sector or Contour. **Sector** The closed profile is separated into several sectors. The origin of the rays that produce the polygon is set in **Center Selection**. The output area is equal to the sum of these sector areas. **Contour** The closed profile is considered a closed contour, and Green's theorem is used to compute its area. Note that this method only supports uniform profiles (uniform spacing is enabled on the **Acquire** page, in the **Scan mode** section). On non-uniform profiles the tool will return incorrect measurements.
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 227.
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. 
Sample Spacing	The angle interval around the center of the profile the tool uses to calculate area.

Parameter	Description
	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.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

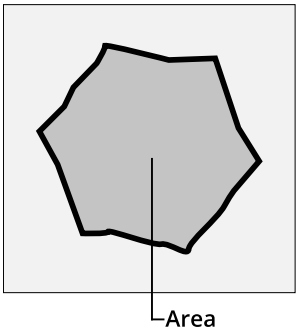


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

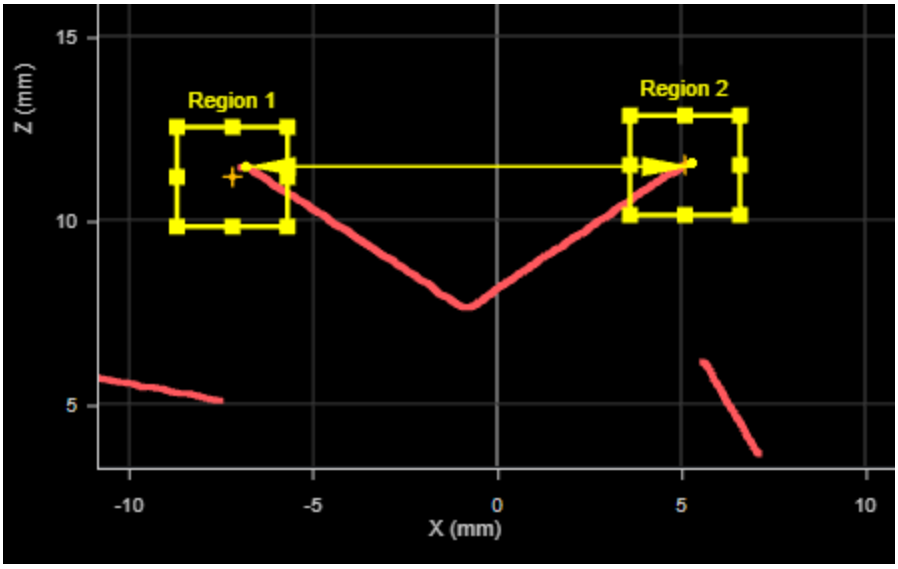
You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Measurement	Illustration
Area Measures the cross-sectional area within a region using data from a dual- or multi-sensor system.	

Profile Dimension

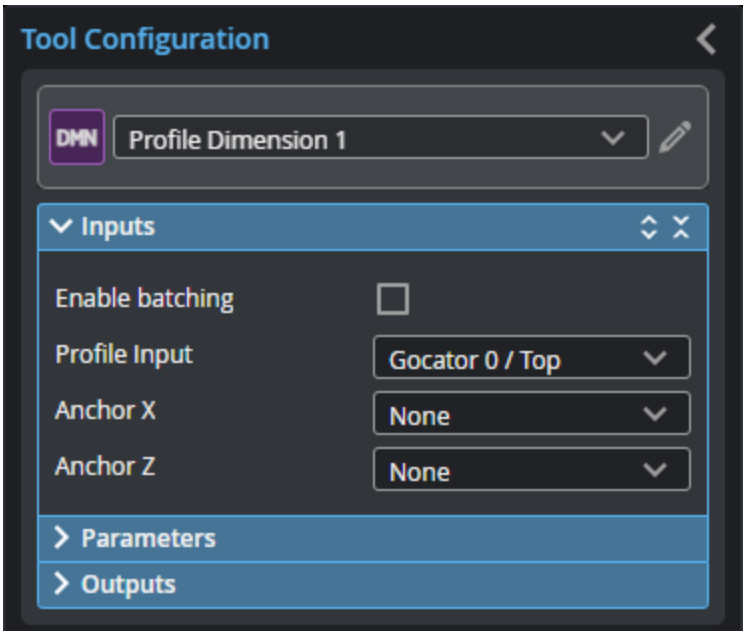
The Dimension tool provides Width, Height, Distance, Center X, and Center Z measurements.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



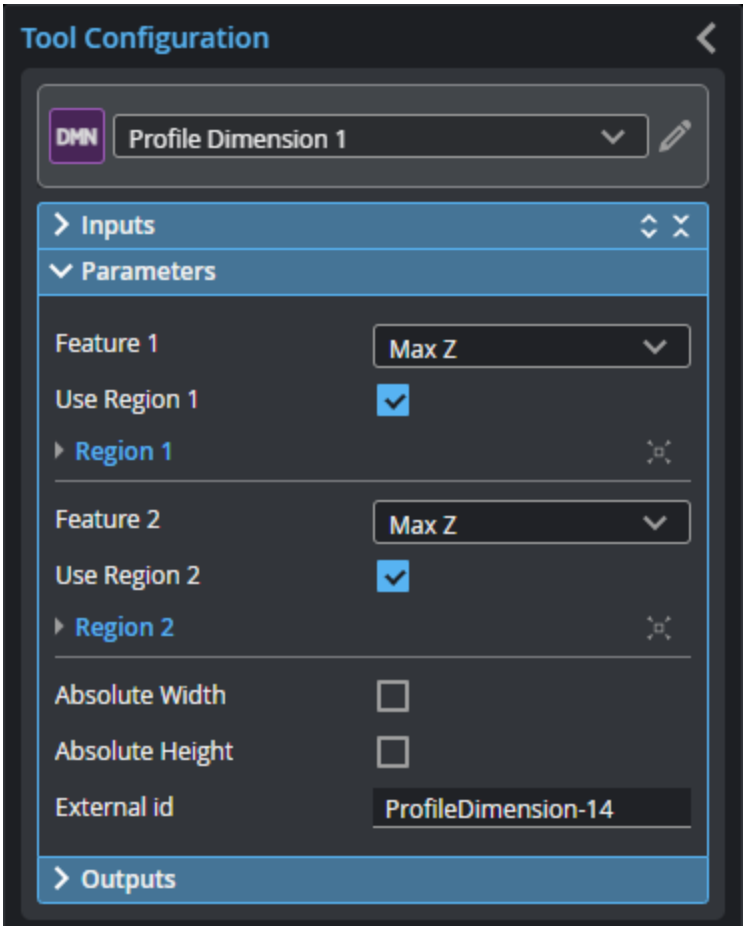
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

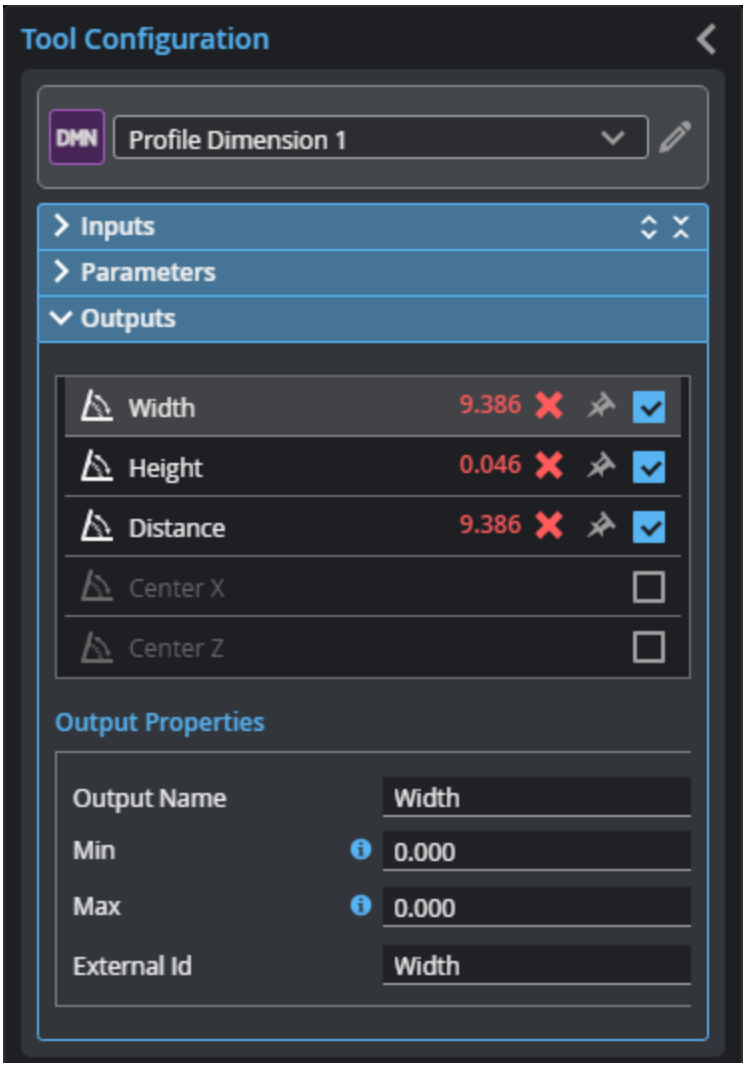


Parameters

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 For more information on feature points, see *Feature Points* on page 224.
Use Region 1 Use Region 2	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region 1 Region 2	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
Absolute Width Absolute Height *(Width and Height measurements only)*	Determines if the result will be expressed as an absolute or a signed value.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

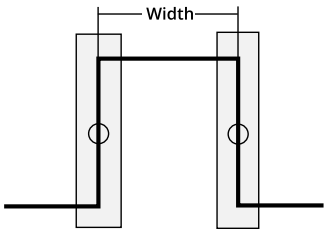
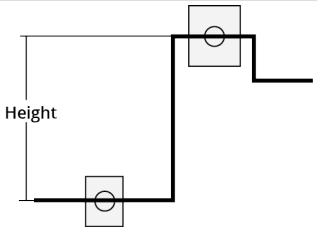
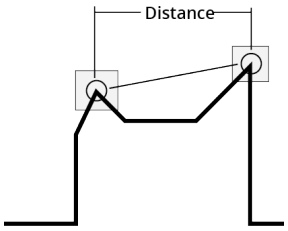
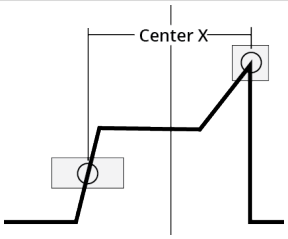
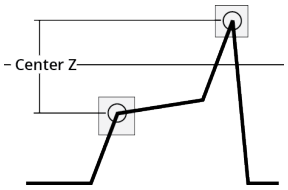
Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

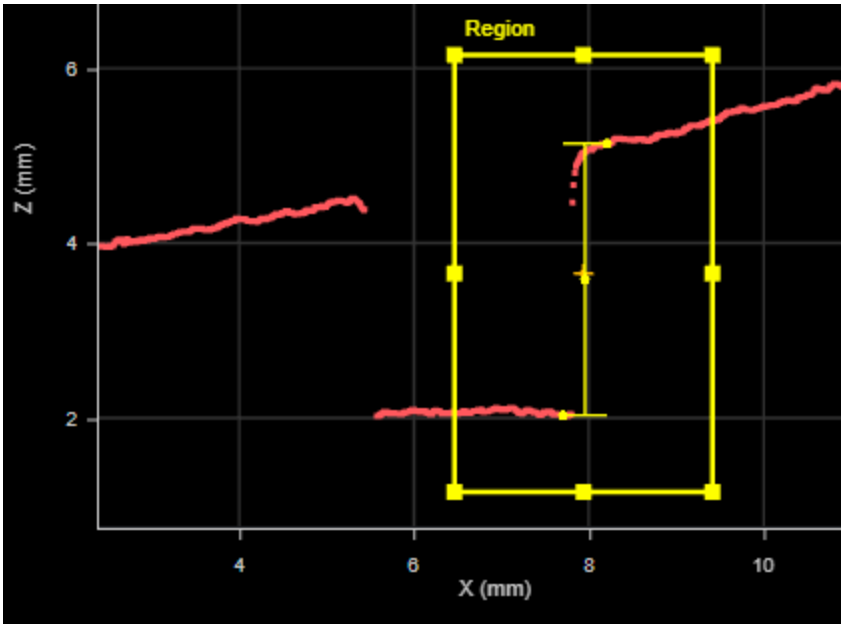
By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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: $Height = Feature\ 2_{Z\ position} - Feature\ 1_{Z\ position}$	
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.	

Profile 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 profile. You can configure the tool to locate a step or a corner (that is, for cases where there is no clear step in the profile but instead a smooth slope), as well as rising or falling steps.



Step height of a step

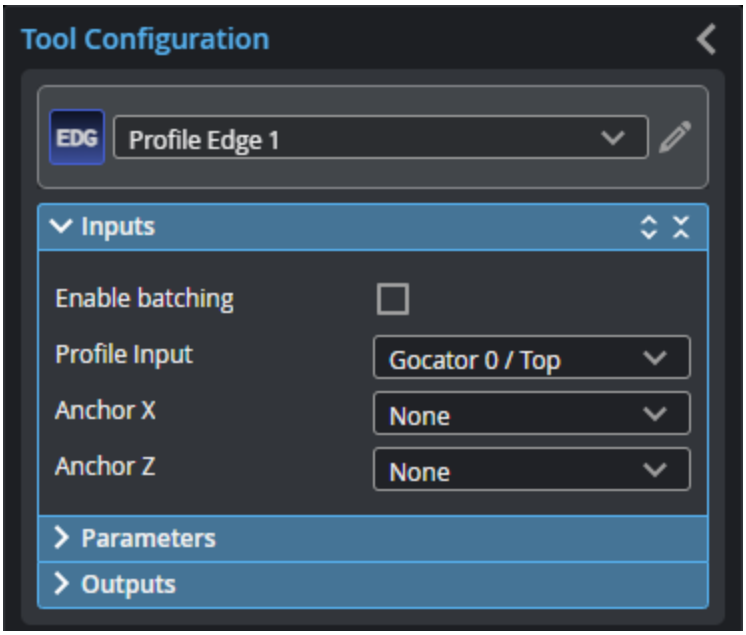
After the tool locates an edge, it returns the position (X and Z) of the edge. For steps, it also returns the step height.

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 Tools* on page 252.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



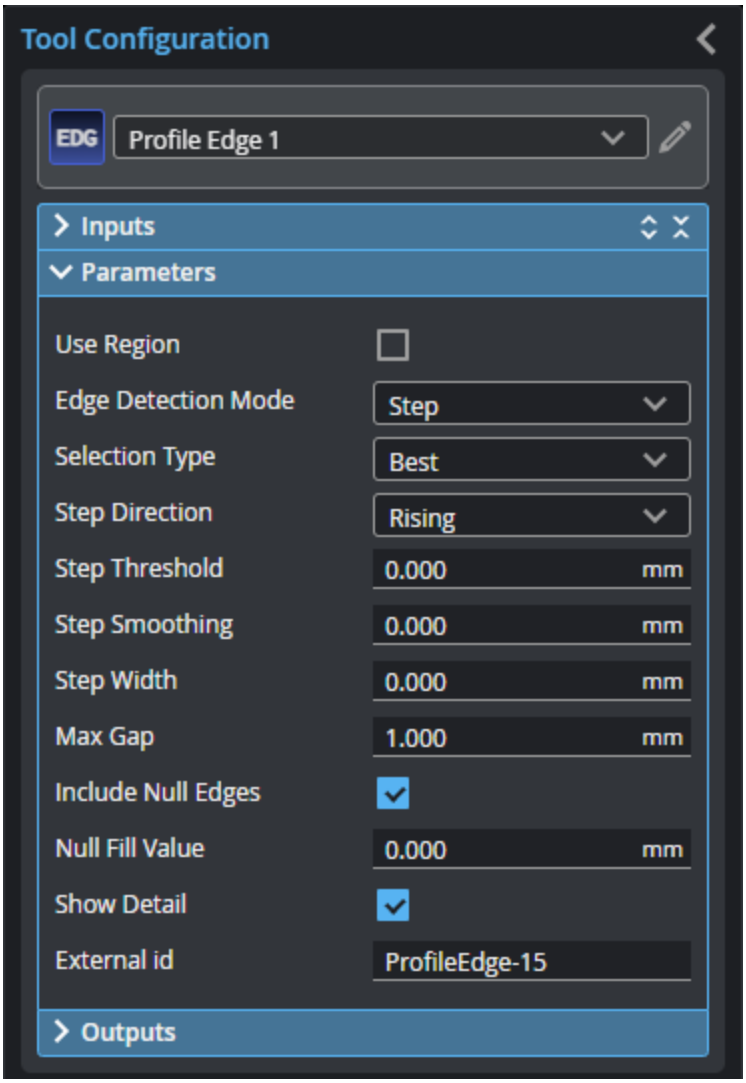
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

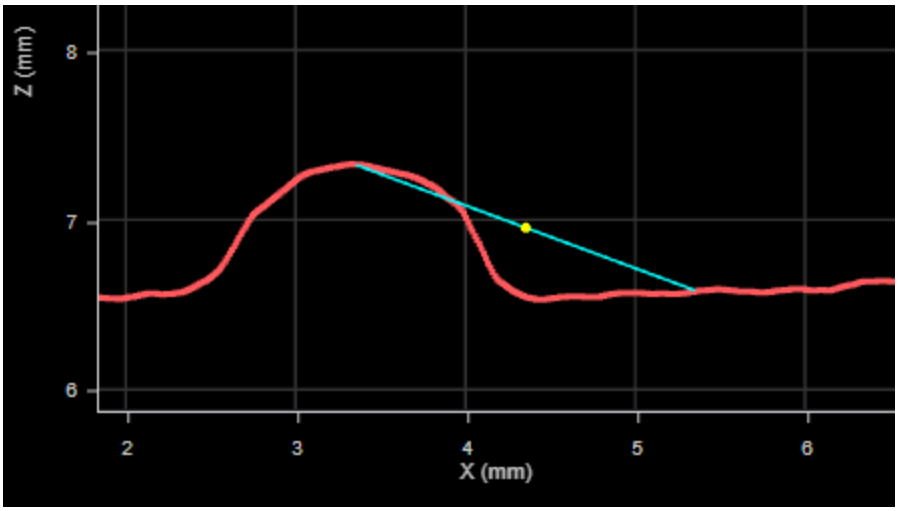
Parameters

You configure the tool's parameters in the expandable **Parameters** section.



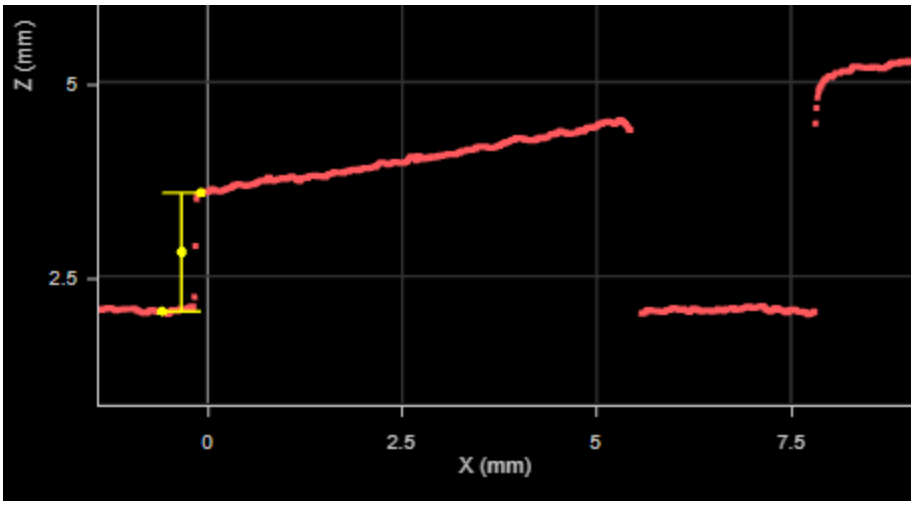
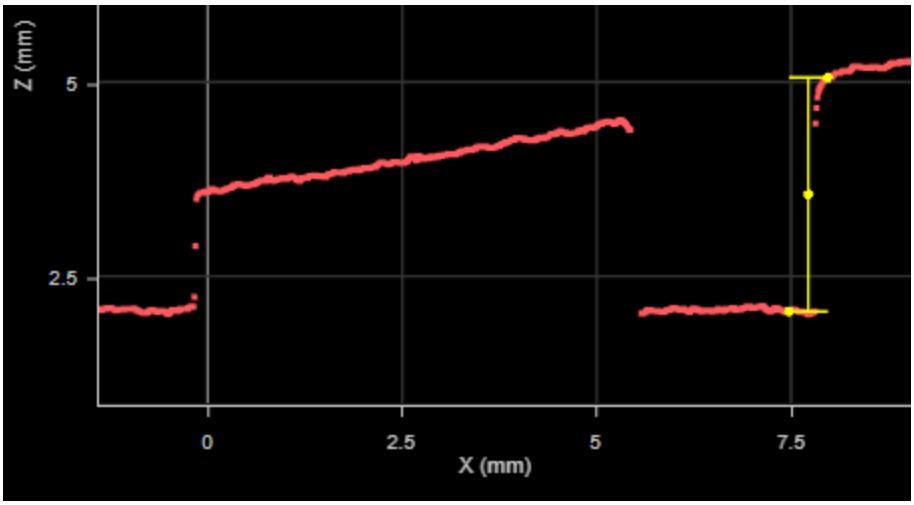
Parameters

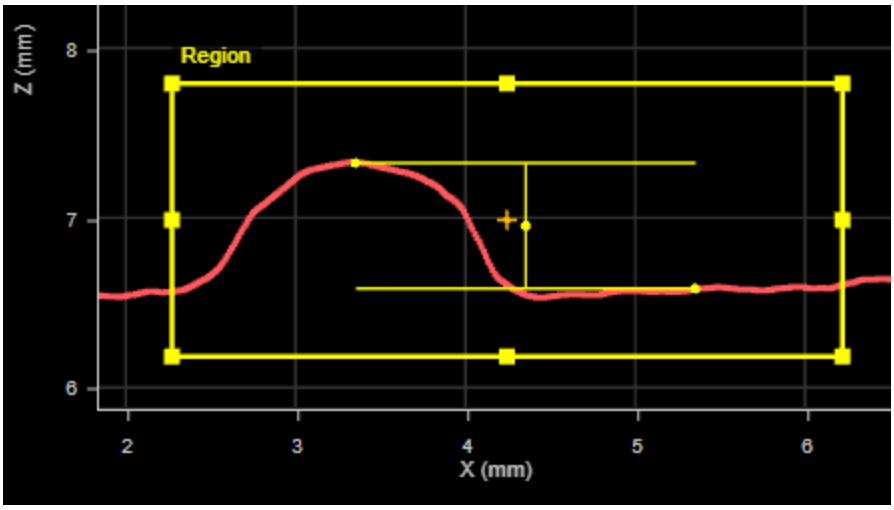
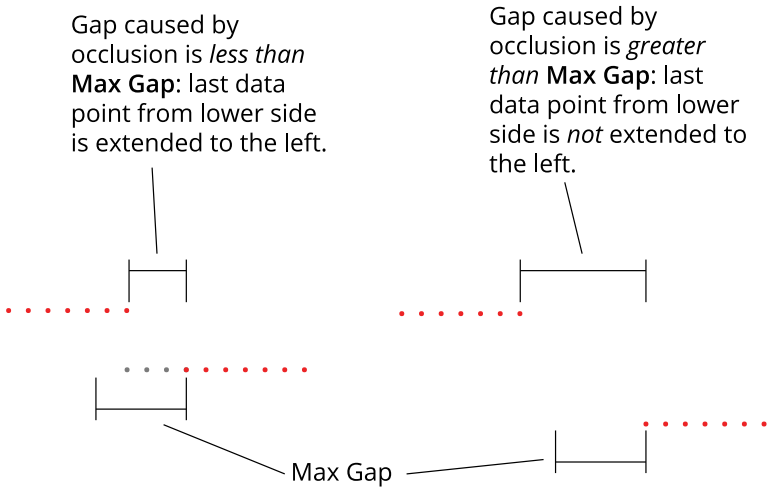
Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
Edge Detection Mode	One of the following: Step or Corner. **Step**: Searches for steps on each path profile. When the edge detection mode is set to Step, you must set several additional parameters. For more information, see *Step Edge Detection Mode Parameters* on the next page. **Corner**: Searches for slopes on each path profile. When the edge detection mode is set to Corner, you must set the **Corner Type** parameter. For more information, see *Corner Edge Detection Mode Parameters* on page 561.

Parameter	Description
Show Detail	When the parameter is enable, the tool displays a line joining the points used to calculate the edge. This can be useful when viewing the edge's center point on a step. Note that enabling this parameter can have an effect on processing. For high frequency scanning, be sure to disable it. 
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Step Edge Detection Mode Parameters

Parameter	Description
Selection Type	Determines which step the tool uses when there are multiple steps in the profile. An edge point is placed on 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.
Step Direction	Determines whether the expected step rises or falls, moving from left to right along the profile. Either **Rising**, **Falling**, or **Rising or Falling**.
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 1.2 (and **Selection Type** set to First), the tool accepts the rising step to the left, with a step of 1.525 mm, because it is the first above the step threshold.

Parameter	Description
	 In the following, when **Step Threshold** is increased to 2.7, the tool excludes the rising step to the left, because it is no longer above the step threshold, and instead uses the step to the right. 
Step Smoothing	The size of the (moving) window along the profile used to calculate averages for the data points used to calculate the step. The setting is useful for averaging out noise. If **Step Smoothing** is set to 0, no averaging is performed.
Step Width	The distance, along a profile, separating the points the tool uses to find steps on a profile. In the following, a step width of 2 mm causes the tool to consider profile points that distance apart as steps. The curved portion of the profile is not used to measure the step.

Parameter	Description
	 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 slope ensures that the tool measures the height difference between the flat regions on either side of the slope. 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 accuracy 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).
Include Null Edges	Indicates whether null points (points where no height value is available, due to

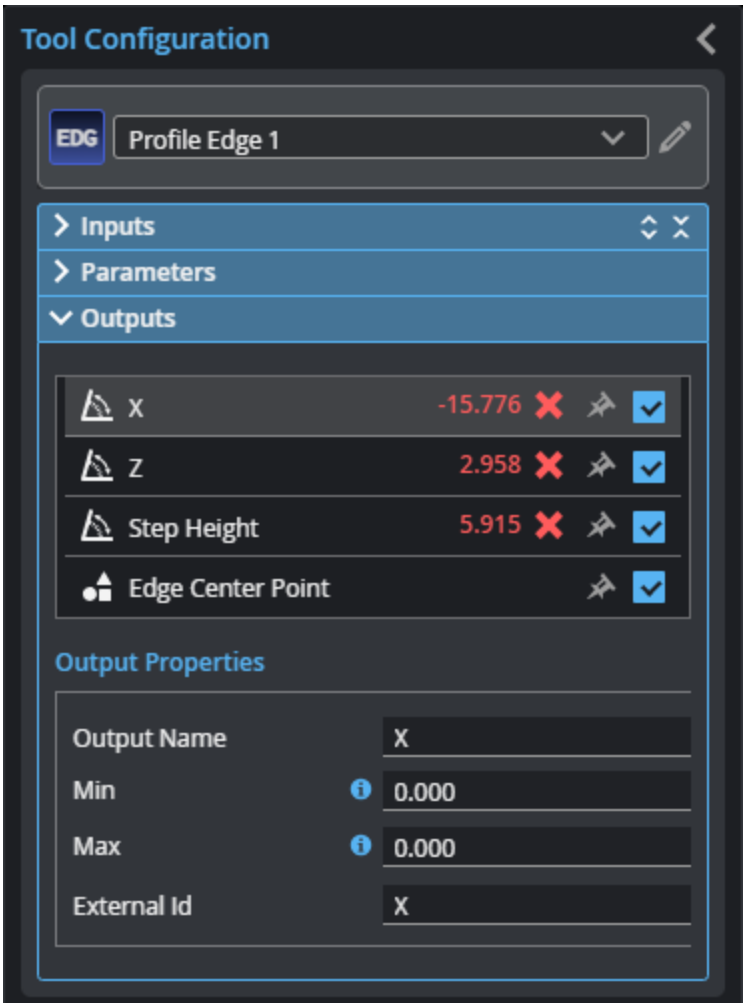
Parameter	Description
	dropouts or regions outside of the measurement range) are filled with the value in **Null Fill Value** as a general “background level.”
	To find an edges next null points, you must use either this option and an appropriate value in **Null Fill Value** or **Max Gap**. Otherwise, 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**.

Corner Edge Detection Mode Parameters

Parameter	Description
Corner Type	Determines the corner type the tool searches for. One of the following: **Best**: Selects the greatest step in the profile data. **Left**: Selects the left-most step in the profile data. **Right**: Selects the right-most step in the profile data. **Top**: Selects the step that is the highest in the profile data. **Bottom**: Selects the step that is lowest in the profile.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement
X **Z** These measurements return the X and Z position of the edge point, respectively. The edge point is located halfway between the upper and lower data points of the step.
Step Height

Measurement

Returns the height of the step on the profile.

Only available if **Edge Detection Mode** is set to Step.

Features

Type	Description
Edge Center Point	The edge point.

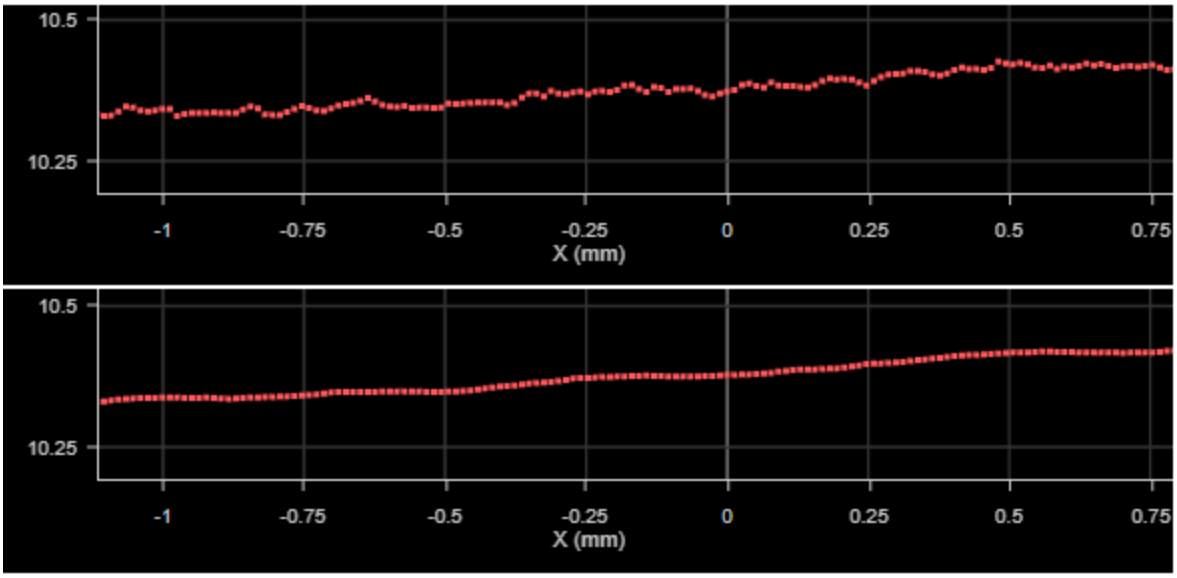


For more information on geometric features, see *Geometric Features* on page 227.

Profile Filter

The Profile Filter tool provides processing filters that you can apply to a uniform profile (not point cloud profiles), letting you 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 Profile tool can use the resulting filtered profile as input.

(The following screenshot is from a G2 sensor.)



Profile with no filtering (top) and with averaging filter (bottom)

For a list of the filters, see *Filters* on page 567.

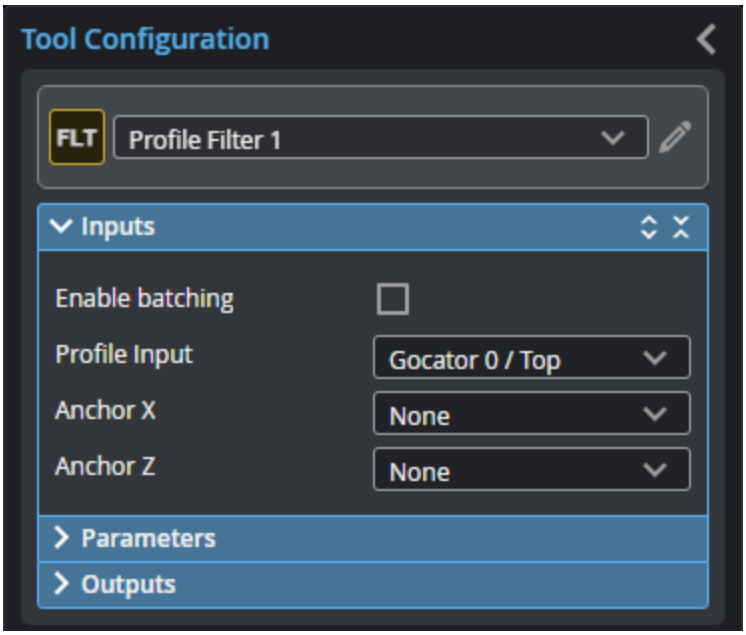
The Filter tool provides no measurements or decisions. It only outputs processed profile data.

> A limited set of filters is also available on the **Acquire** page, in the **Filters** section. These filters let you process scan data without needing to add tools. This can be useful if you are using a sensor mostly as an acquisition device.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



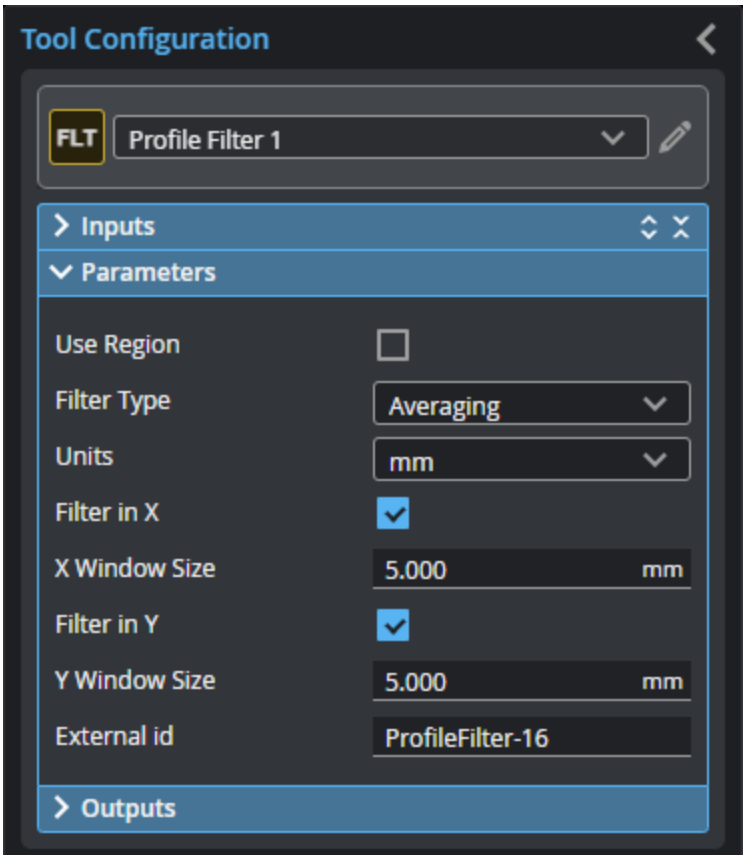
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Number of Regions Region {n}	Lets you set the number of regions, and for each region, the position and dimension. For more information on regions, see *Regions* on page 214.
Filter Type	The type of filter. For more information on the available filters, see *Filters* on the next page.
Units	The units the filter uses for the window or windows: points or distance (mm). Not available with all filters. This parameter is only displayed after enabling **Filter in X** or **Filter in Y**.
Filter in X Filter in Y	When enabled, filters along X and Y, respectively, and displays **X Window Size** or **Y Window Size** parameters you use to set the window size, using the unit set in **Units**. (The gap filling and Gaussian filters only let you filter along the X axis.) The filter window sizes are specified in millimeters (and additionally, in the tools, in data points). To calculate the number of data points that a window covers when the units are millimeters, use the following calculation: • User-specified window size divided by the X spacing interval (that is, the

Parameter	Description
	number of millimeters per point) on the **Spacing** tab in the **Sensor** panel. (For more information on spacing intervals, see *Spacing on X and Y (Uniform Spacing)* on page 140.) • *With the exception of the gap filling filter*, round the result of the division to the nearest integer value. With the gap filling filter, filling is performed within the provided window size. For example, if you set the size of the filter's window to a value between 1.5 mm and 2.49 mm (inclusively), and the X spacing interval is set to 1 mm, the filter covers 2 data points. A filter window size from 2.5 mm to 3.49 mm results in a filter covering 3 data points.
Sigma	The Gaussian curve's sigma value. (Only displayed with the Gaussian filter and when **Filter in X** is enabled.)
Percentile Type	Displayed when **Filter Type** is Percentile. (Be sure to set the **High Percentile** and **Low Percentile** as required.) One of the following: **Point Count**: Removes points based on the count of valid points in the profile. **Point Height Range**: Removes points based on the height range of valid points in the profile.
High Percentile Low Percentile	The range for used when you set **Filter Type** to Percentile. You must also set **Percentile Type**.

The following filters are available in the Profile Filter tool.

Filters

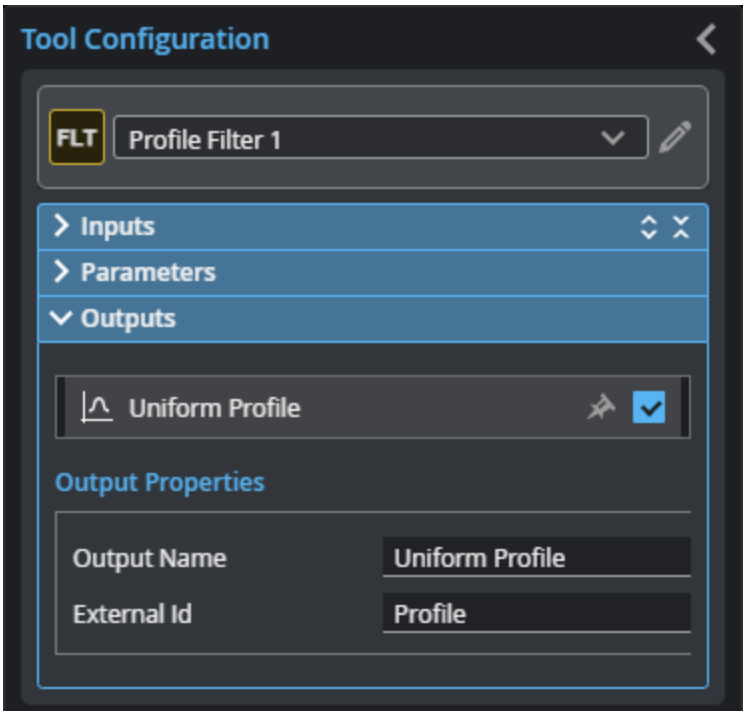
Name	Description
Gap Filling	Fills in missing data using information from the nearest neighbor data points, for example, when data is missing due to occlusions. Gap filling also fills gaps where no data is detected, which can be due to low surface reflectivity, for example dark or specular surface areas, or to actual gaps in the surface. The values in **Filter in X** and **Filter in Y** represent the maximum gaps the sensor will fill. Wider gaps are not filled. 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 the specified window. The sensor can fill gaps along both the X axis and the Y axis. In Profile mode, gap filling is limited to the X axis.
Median	Substitutes the value of a data point with the median calculated within the window or windows set in **X Window Size** or **Y Window Size** around the data point. If the number of valid (non-null) data points in the window is even, the average of the two values in the center is used. If the number of valid points is odd, the value in the center of the sorted list of values is used.

Name	Description
	Missing data points will not be filled with the median value calculated from data points in the neighborhood. With an odd-numbered window size, the output is at the center of the window. With an even window size, the output is 0.5 pixels to the right of the center (that is, using window / 2-1 values from the left, and window / 2 from the right.
Averaging	Substitutes a data point value with the mean value of that data point and its nearest neighbors within the window or windows set in **X Window Size** or **Y Window Size**. 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 neighborhood.
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 Window Size** to 0.2, only points every 0.2 millimeters will be used. The filter generates points starting from the leftmost edge of the scan data, stepping in equal steps away from that side.
Gaussian	A Gaussian filter applied over the specified kernel using the provided sigma. Enable **Filter in X** and set the **Sigma** parameter.
Percentile	Filters points based on the range you define with **High Percentile** and **Low Percentile**, in addition to the type set in **Percentile Type** (see above).

The filters described here that are available in the Filter panel topic use the same algorithms.

Outputs

The tool provides the following output.

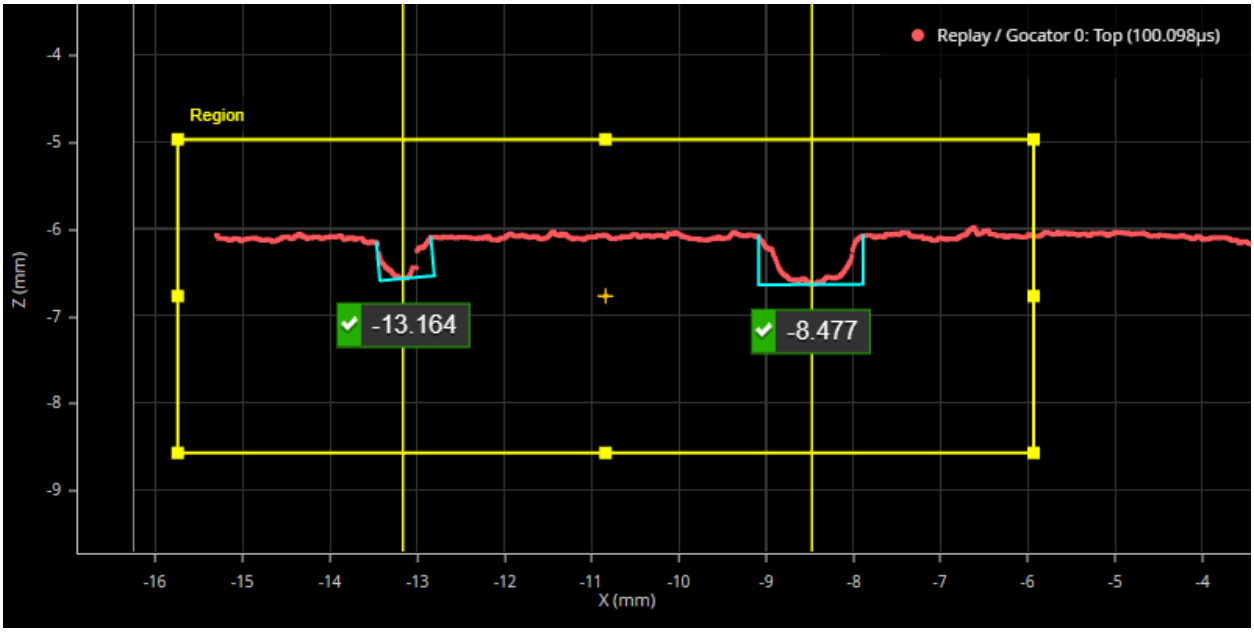


Data

Type	Description
Uniform Profile	The filtered uniform profile.

Profile Groove

The Groove tool provides measurements of V-shape, U-shape, or open-shape grooves.



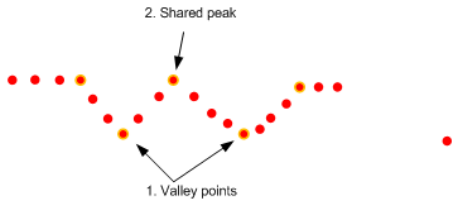
The Groove tool lets you add multiple measurements of the same shape to receive measurements and set decisions for multiple grooves by setting the Groove Count parameter and then configuring each groove's selection type, index, and location (bottom, or left or right corner).

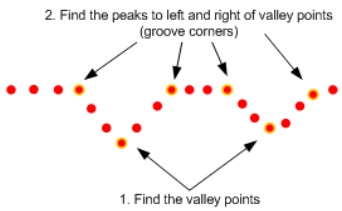
The Groove tool uses a complex feature-locating algorithm to find a groove and then return measurements. For a detailed explanation of the algorithm the tool uses, see *Groove Algorithm* below. The behavior of the algorithm can be adjusted by changing the tool's parameters.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

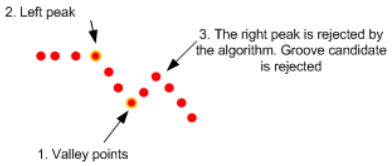
Groove Algorithm

The Groove measurement tool first locates a valley along the profile line. The bottom point of a valley, the valley point, is the first estimation of the position of the groove bottom. For each valley, the algorithm searches for the corner to the left and to the right to find the groove corners. A groove candidate is found when the groove corners are located on the left and right before the next valley is reached. Two groove candidates may share the same corner as shown in the right image below.

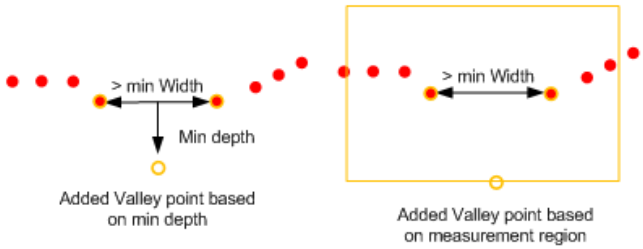




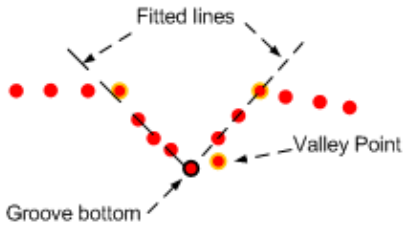
The algorithm derives search parameters from the user settings to prevent noise from triggering false detections. When detecting multiple grooves, an adaptive algorithm is used to ensure that candidate grooves are approximately the same scale.



The valley points of open grooves may not be visible or may fall outside of the measurement region. Voids in the data (regions with no profile data) between pairs of valid points are detected. A valley point is added midway between the pair of valid points. The Z position of the valley point is either the minimum groove depth below the lower of the corners or the bottom edge of the measurement region. The algorithm then proceeds as if to find a U-shaped groove.

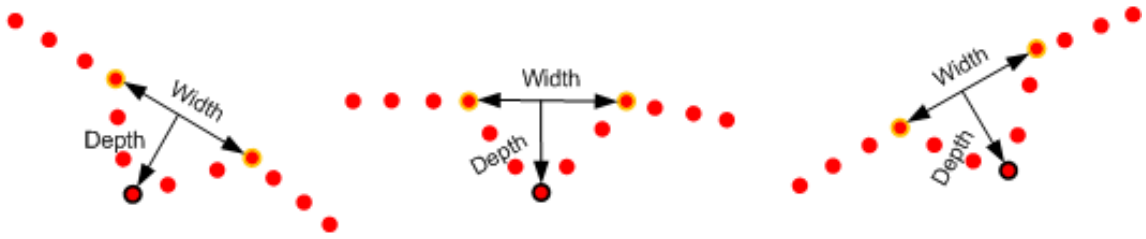


The actual groove bottom is calculated differently for different shapes. For a V-shaped groove, a line is fitted to the sides of the valley points starting from the corners, up to (but not including) the valley point. The groove bottom is the intersection of the left and right lines. Line fitting is used such that an accurate groove bottom can be found even when the real bottom is not visible (i.e., blocked by reflections).



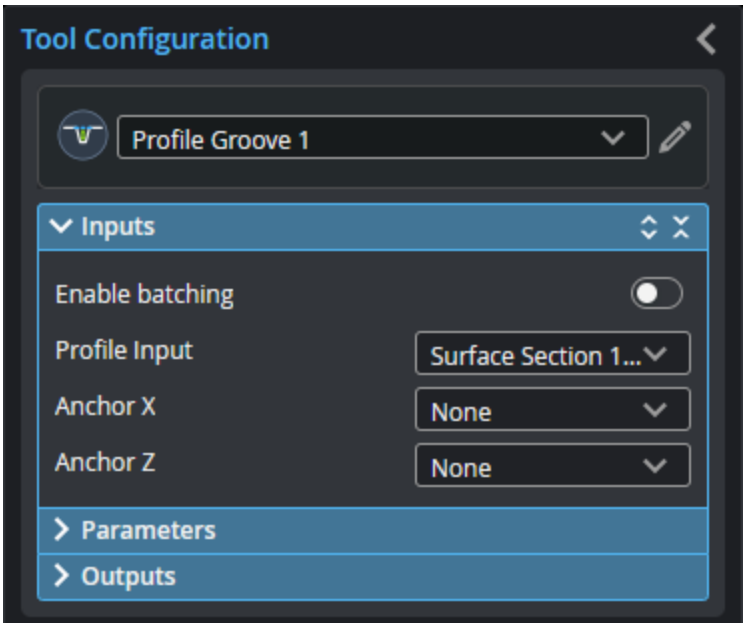
For U-shaped and open groove, the X and Z positions are at the bottom of the groove.

Groove candidates that do not meet the minimum and maximum width and depth settings are rejected. The width and depth measurements are invariant to the groove rotation. The width is the distance between the groove corners and the depth is perpendicular distance of the groove bottom from the groove width.



Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

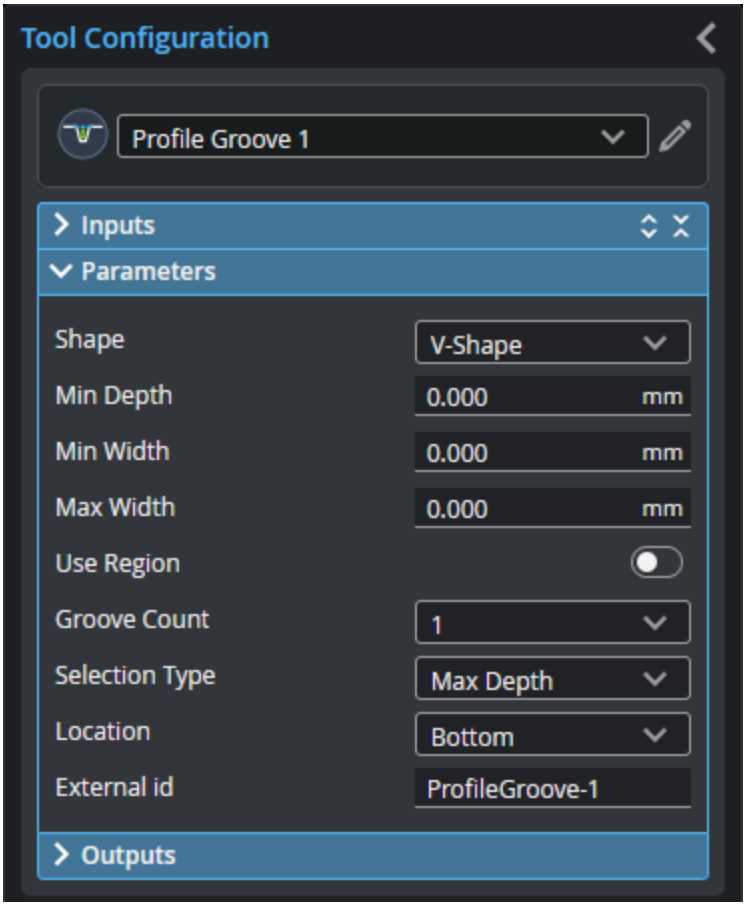
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor.

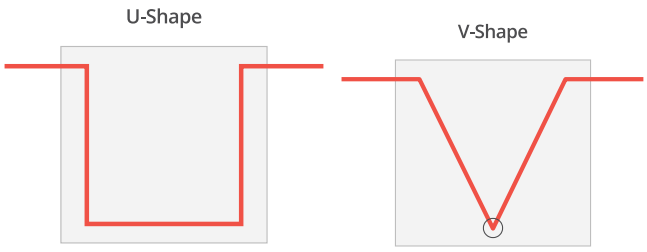
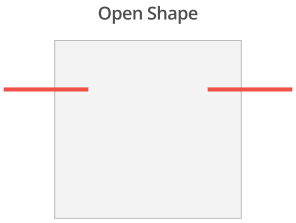
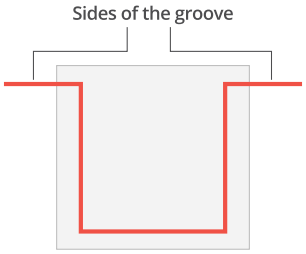
Name	Description
	When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with a maximum of two elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. If **Enable Batching** is *disabled* and the passed array contains more than two elements, GoPxL displays an error. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



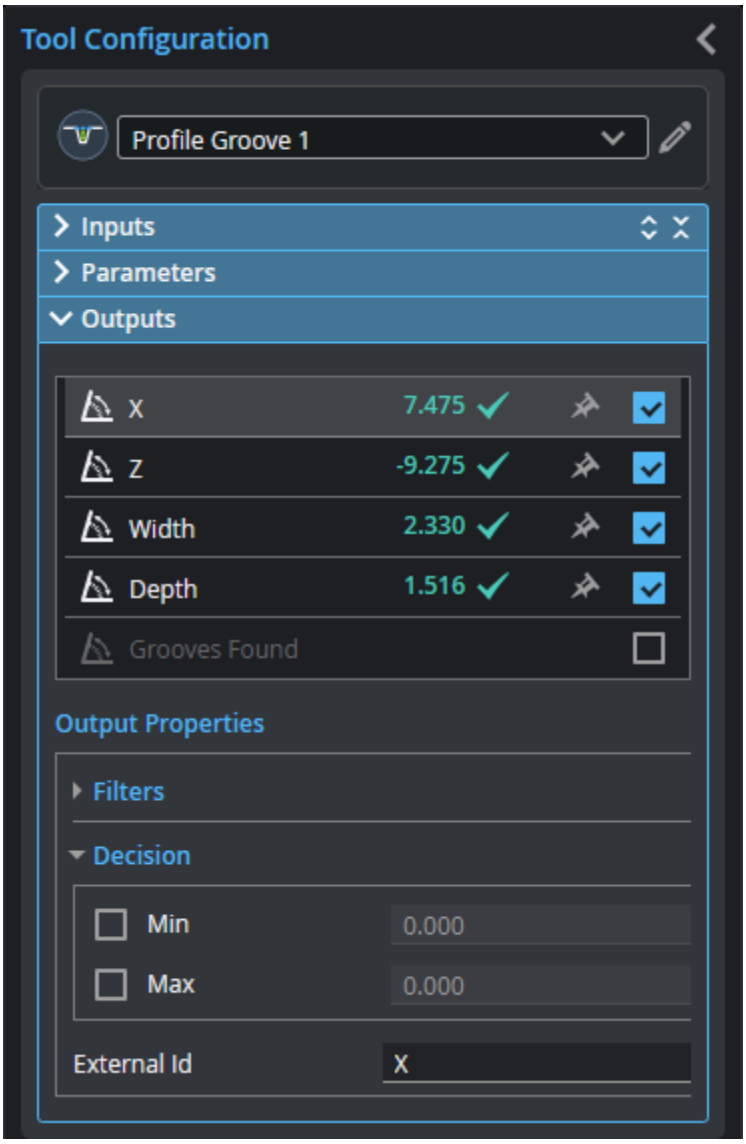
Parameters

Parameter	Description
Shape	One of the following: U-Shape, V-shape, or Open.  
Min Depth	Minimum depth for a groove to be considered valid. When set to 0, any groove depth is accepted.
Min Width	Minimum width for a groove to be considered valid. The width is the distance between the groove corners. When set to 0, any groove width is accepted.
Max Width	Maximum width of a groove to be considered valid. If set to 0, the maximum is the width of the region of interest, or the entire profile if **Use Region** is disabled.
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The measurement region defines the region in which the tool searches for a 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 214.

Parameter	Description
Groove Count	The number of expected grooves you want the tool to return. If the tool finds fewer grooves than the number you specify, it will return Invalid results.
Selection Type {n}	Specifies how a groove is selected. One of the following: **Maximum Depth**: The groove with the maximum depth. **Index from The Left**: A 0-based groove index, counting from left to right. **Index from the Right**: A 0-based groove index, counting from right to left.
Selection Index {n}	The 0-based groove index the tool uses to select a groove when multiple grooves are visible in the data. The direction depends on whether you set **Selection Type** to **Index from the Left** or **Index from the Right**.
Location {n} *(For the X and Z measurements only)*	Specifies the location type to return. One of the following: **Bottom**: The groove bottom. For a U-shape and open-shape groove, the X position is at the center of the bottom 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. For more details, see *Groove Algorithm* on page 570. **Left Corner**: The groove's left corner. **Right Corner**: The groove's right corner.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

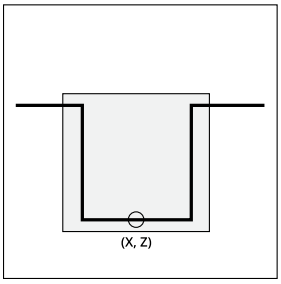
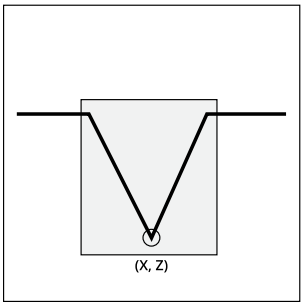
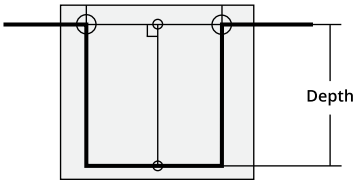
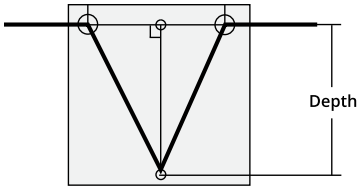
Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

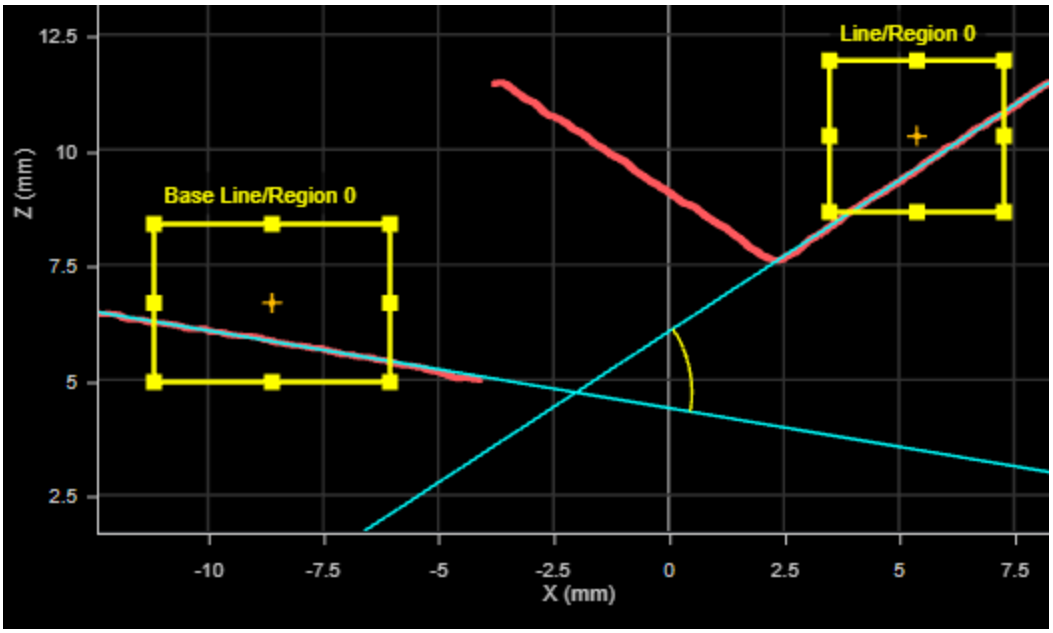
Measurement	Illustration
X, Z Measures the X and Z position, respectively, of the bottom of a groove.	 
Width Measures the width of a groove.	 
Depth Measures the depth of a groove as the maximum perpendicular distance from a line connecting the edge points of the groove.	 
Grooves Found The number of grooves the tool finds.	

Profile 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 one of the following:

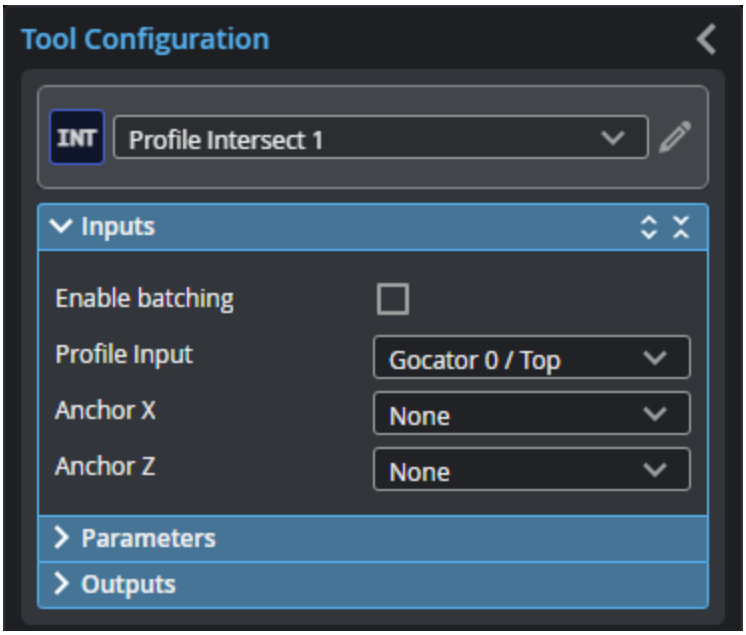
- the X axis (z = 0)
- the Z axis (x = 0)
- a user-defined line



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



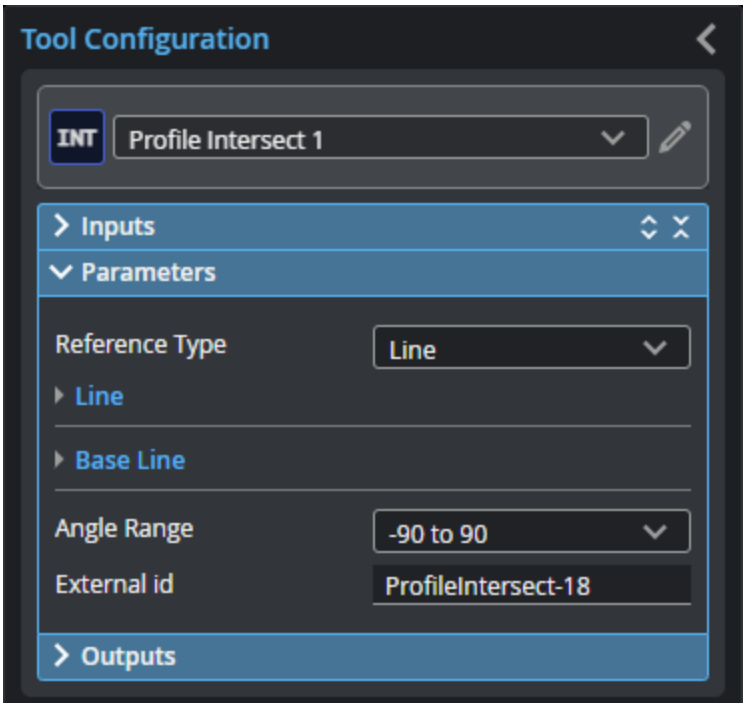
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

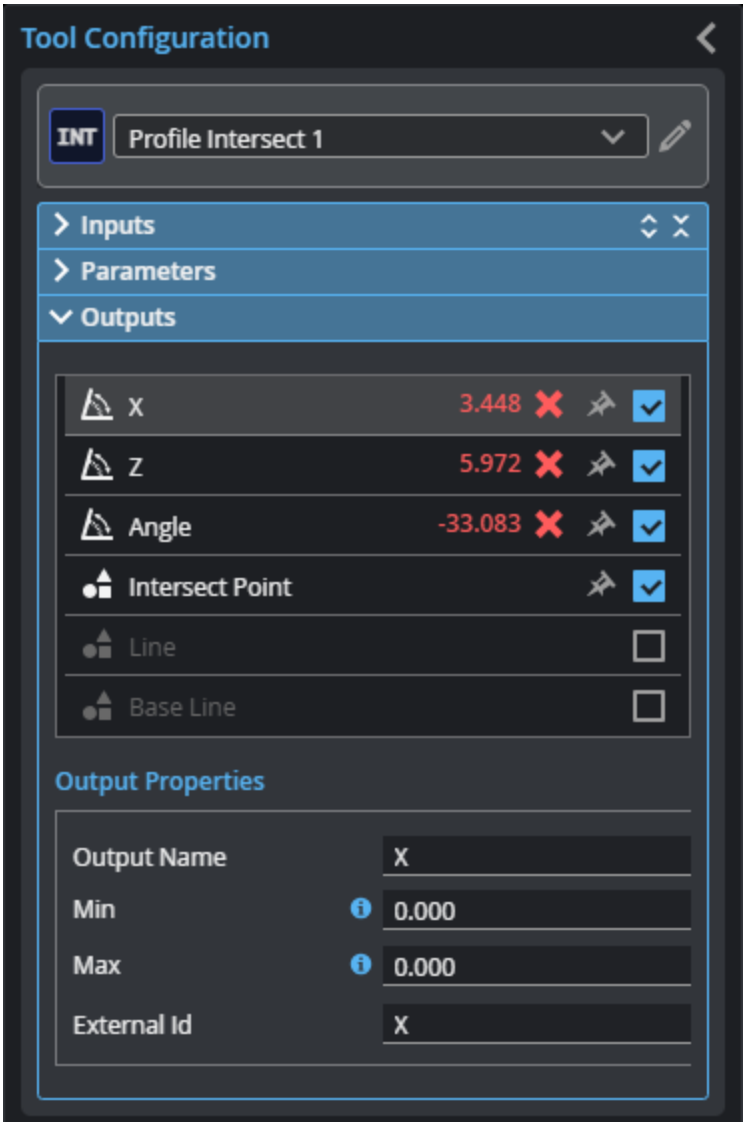


Parameters

Parameter	Description
Reference Type	Determines the type of the reference line. One of the following: X Axis, Z Axis, or Line. **X-Axis**The reference line is set to the X axis. **Z-Axis**: The reference line is set to the Z axis. **Line**: You define the reference line manually using the parameters in the **Base Line** section. You can define the line using one or two regions.
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 and enter the values in the fields. For more information on regions, see *Regions* on page 214. For more information on fit lines, see *Fit Lines* on page 227.
Base Line	Used to define the reference line when the **Reference Type** parameter is set to **Line**. For more information on regions, see *Regions* on page 214. For more information on fit lines, see *Fit Lines* on page 227.
Angle Range	Determines the range returned by the Angle measurement. One of the following: **-90 – 90** **0 – 180**
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

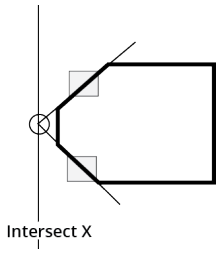
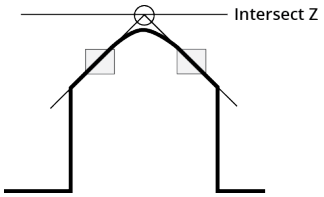
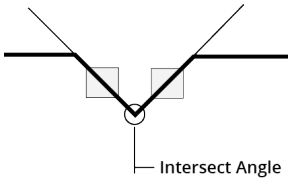


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	Illustration
X Finds the intersection between two fitted lines and measures the X axis position of the intersection point.	
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.	

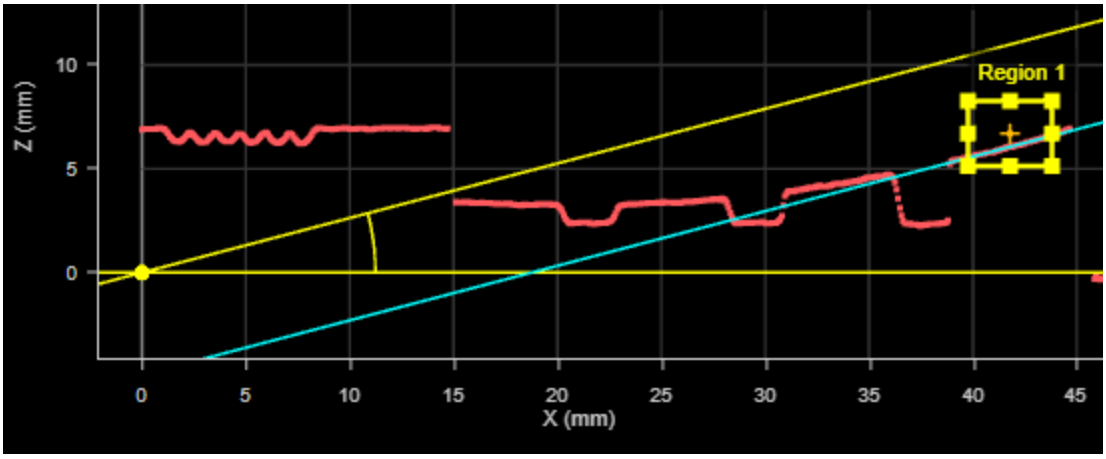
Features

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 227.

Profile Line

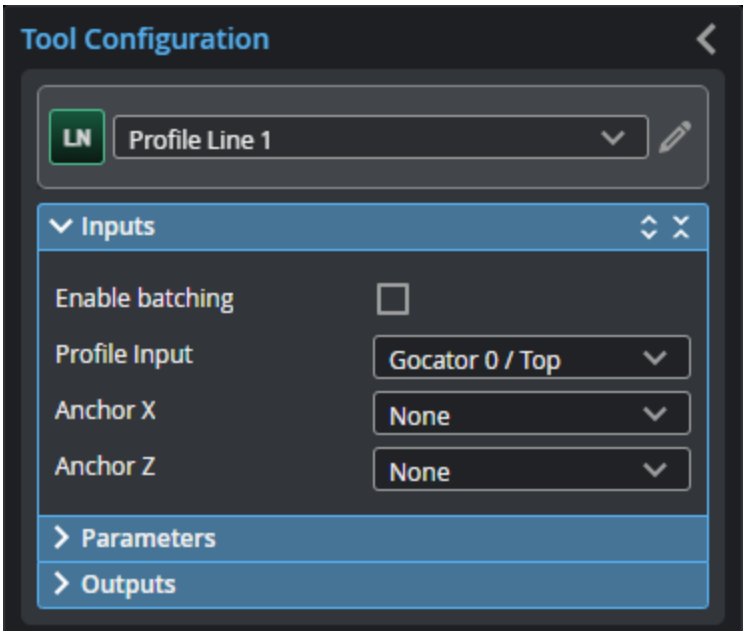
The Line tool fits a line to the profile and measures the deviations from the best-fitted line. The tool also provides two “roughness parameter” measurements: Arithmetic Average and Rz. For a complete list of the measurements the tool provides, see *Measurements* on page 588.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



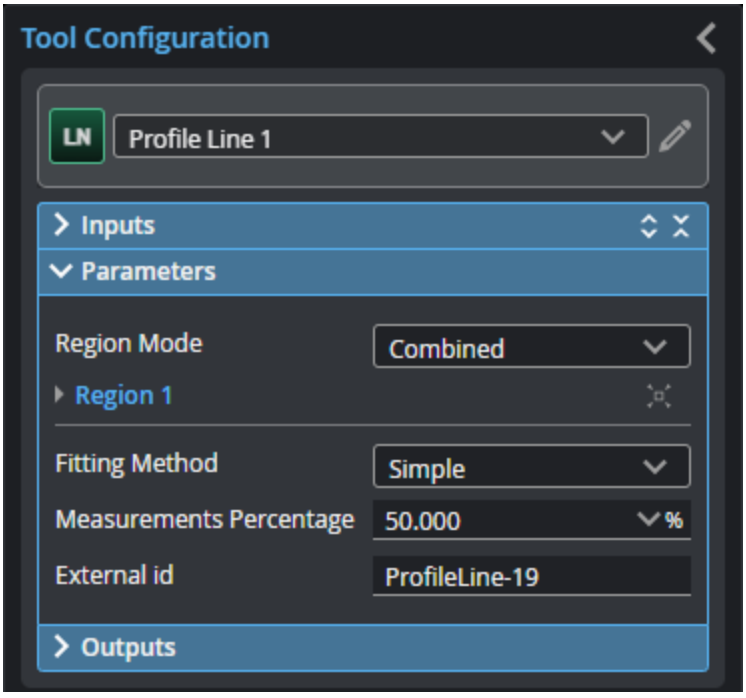
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

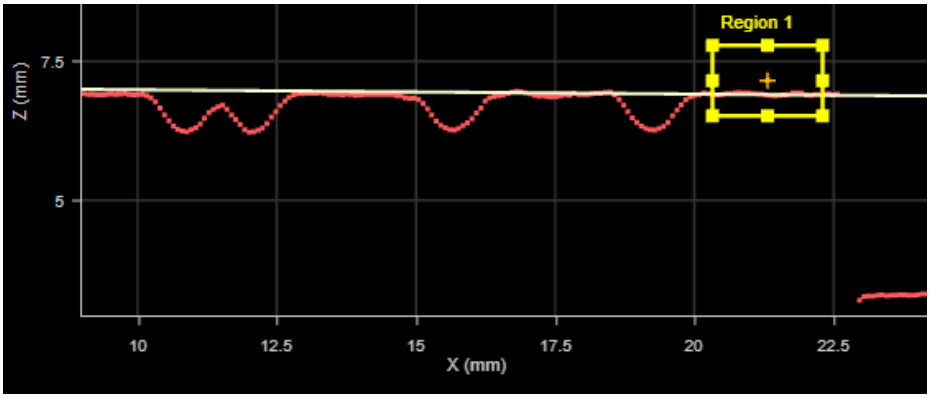
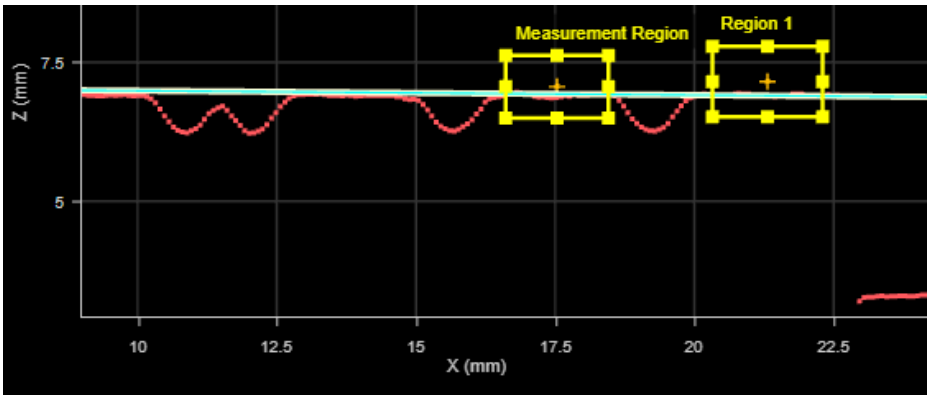
Parameters

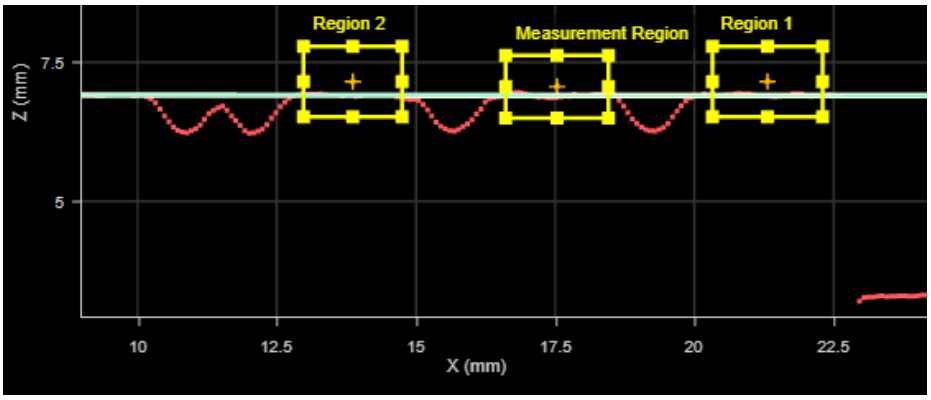
You configure the tool's parameters in the expandable **Parameters** section.



Parameters

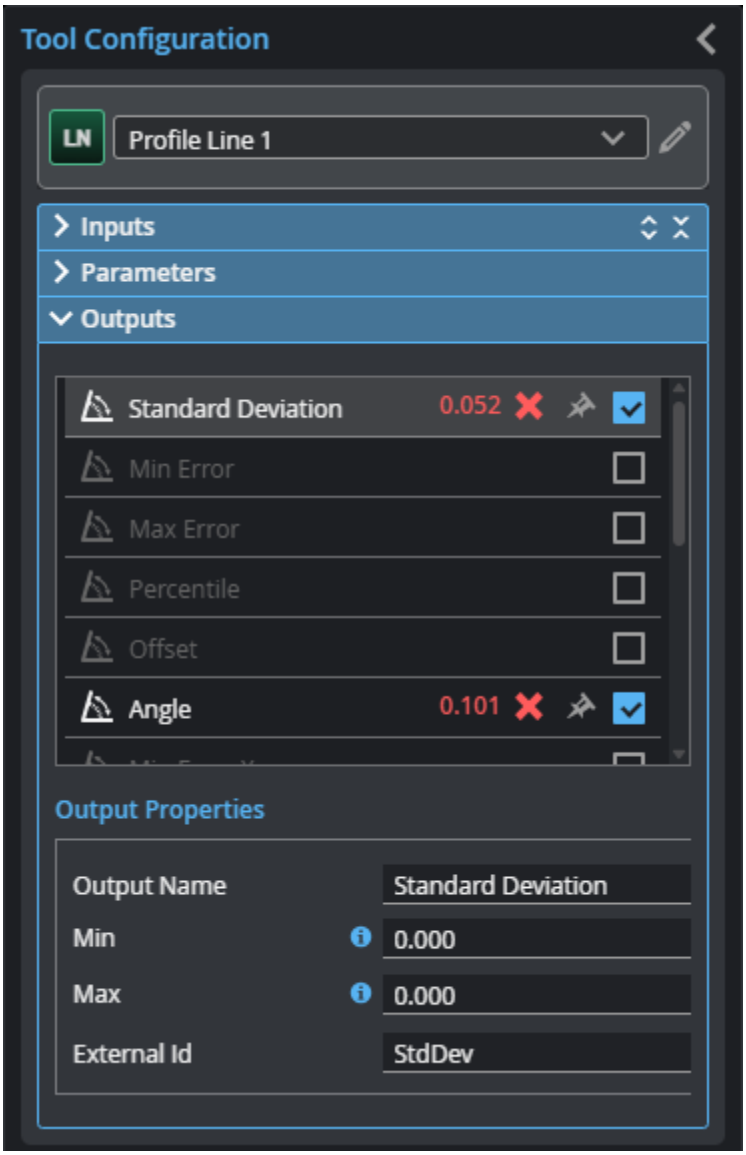
Name	Description
Region Mode	Whether the fitting and measurement regions are used, and whether they are combined or separate. One of the following:

Name	Description
	None The tool uses the entire profile both to fit the line and to perform measurements. **Combined** The tool uses a single, user-defined region to fit the line, in which it also performs measurements.  **Separated** The tool uses one or two regions to fit the line (selected using the **Number of Regions** parameter), and a single, separate region in which it performs measurements. (Offset and Angle measurements are related to the fitted line and are associated with the fitting region, rather than the measurement region.) In the following, the tool 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:

Name	Description
	
Number of Regions	When **Region Mode** is set to Separated, lets you choose the number of regions.
Region {n} Measurement Region	Use these expandable region sections to define the position and size of the fitting and measurement regions precisely. (Use the mouse to position and size them more roughly.) The number of region parameters displayed depends on the mode selected in **Region Mode**.
Fitting Method	Determines how the tool fits the line to the profile. One of the following: **Simple** Uses a less accurate but faster line-fitting method. **Robust** An iterative line-fitting method that removes points and attempts to fit a line until only the fraction set in **Outlier Fraction** of the original profile data points is left. More accurate but takes longer. When **Fitting Method** is set to Robust, set **Outlier Fraction** (see below).
Outlier Fraction	The fraction of outlier points to be removed during line fitting. Adjust this value based on how much noise is present in the profile. By default, 30%. Only displayed when **Method** is set to **Robust**.
Measurements Percentage *(Percentile measurement only)*	The specified percentage of points around the best-fitted line that the Percentile measurement uses.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

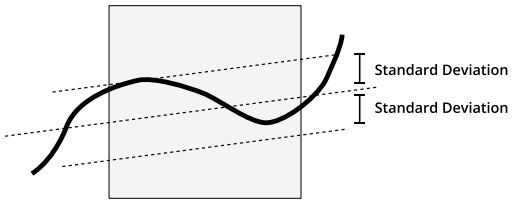
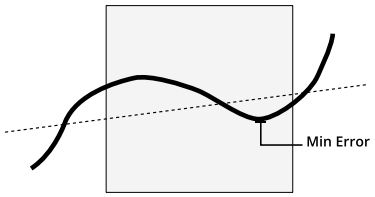
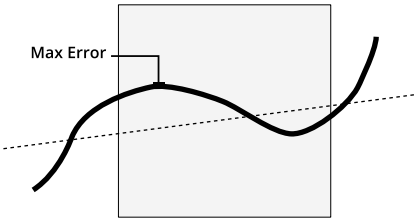
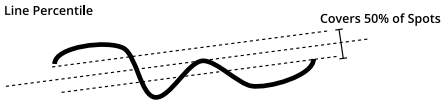
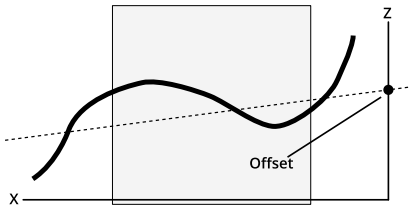
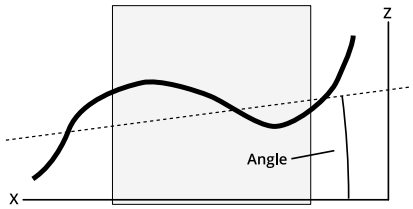
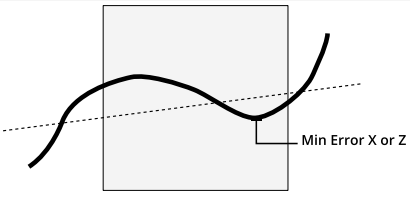
Most tools provide measurements, geometric features, or data as outputs.

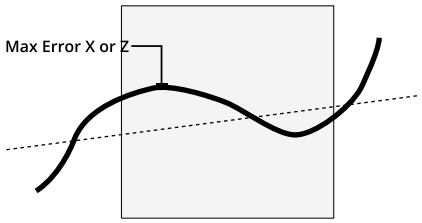
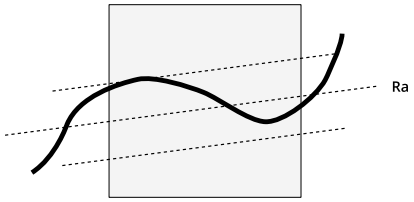
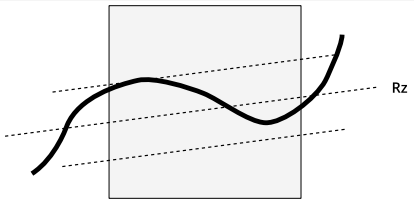


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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).	
Arithmetic Average Returns the roughness average of the profile data. Ra = SUM_N(abs(distance)) / (N) where N is the total number of non-outlier points, and the distance is from each point to the fitted line.	
Rz Returns the maximum height of the profile data. Rz = abs(smallest distance) + abs(biggest distance) The distance is from each non-outlier point to the fitted line.	

Features

Type	Description
Line	The fitted line.
Min Error Point	The point of minimum error.
Max Error Point	The point of maximum error.

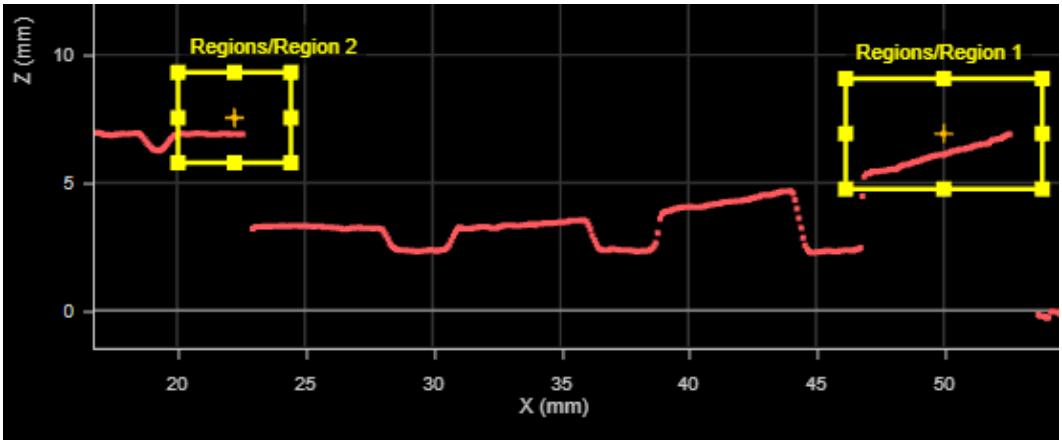
For more information on geometric features, see *Geometric Features* on page 227.

Profile Mask

The Profile Mask tool lets you define up to 16 regions to extract data from a profile. Each region's size, position, and shape (circular, elliptical, and rectangular) can be individually configured, and regions can overlap. The tool can also exclude inner data of circular and elliptical regions, letting you avoid measuring noise or unwanted areas of profile data. Extracted data is output in a single profile.

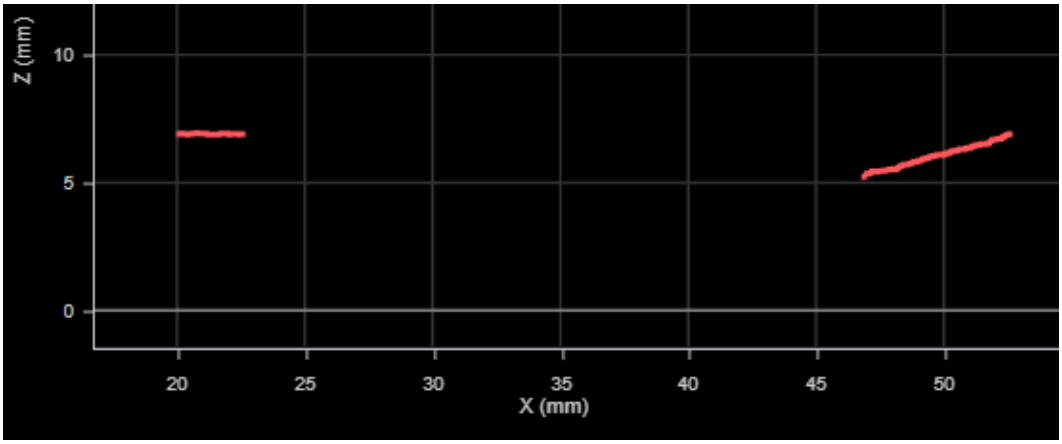
The resulting profile can then be further processed or measured by other tools.

For example, given the following scan data:



Two mask regions defined on a profile (original profile, all data included)

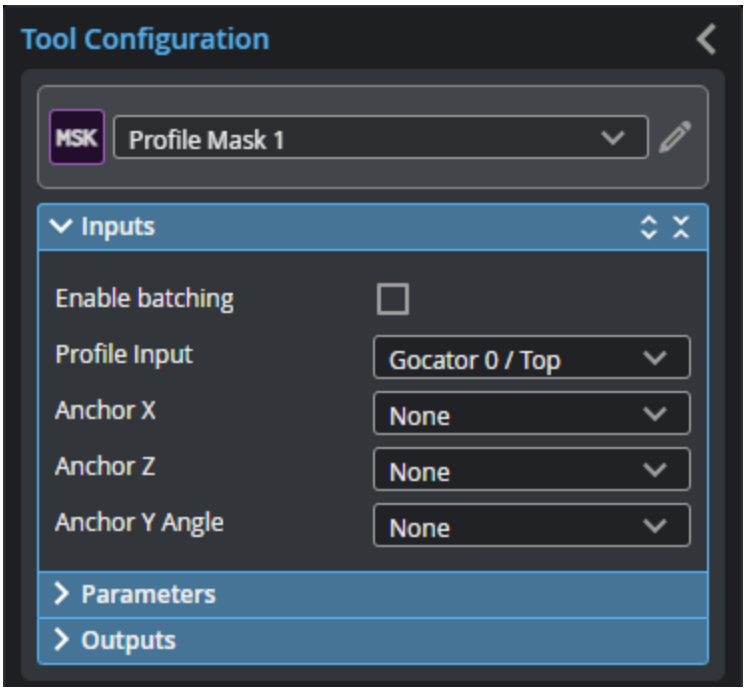
The image below shows the extracted data. The extracted profile data can then be further processed by other tools, or measurements can be applied to the surface data.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



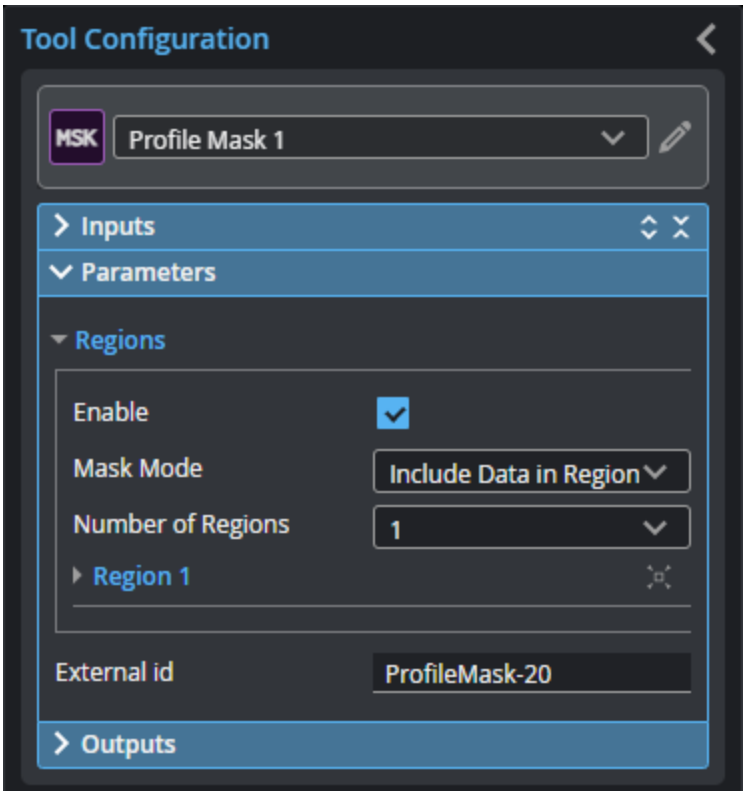
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Y Angle	The Y angle measurement of another tool that this tool uses as a rotational anchor. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

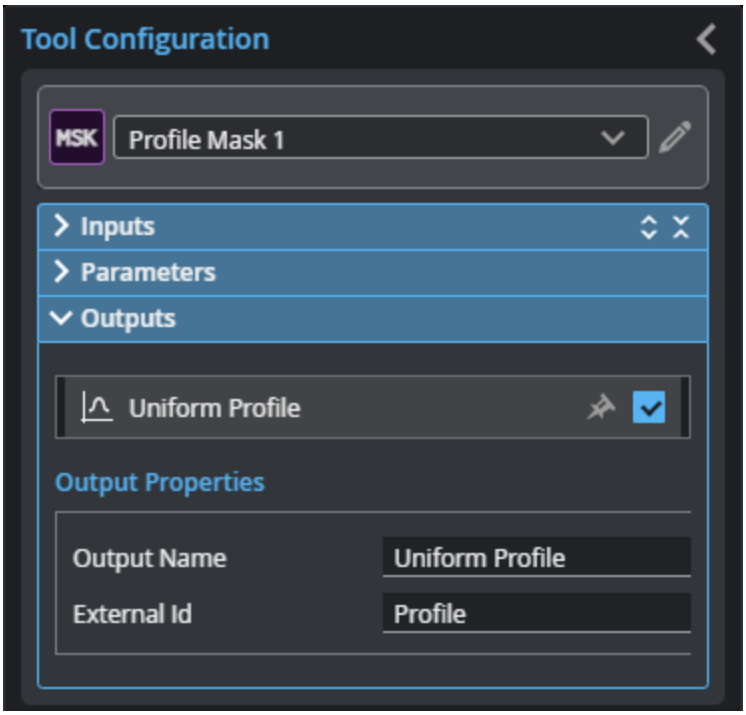


Parameters

Parameter	Description
Regions	When expanded, displays region-related settings. For details on the regions and their settings, see *Flexible Regions* on page 217.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Data

Type	Description
Uniform Profile Point Cloud Profile	The profile containing the extracted region or regions.

Profile Panel

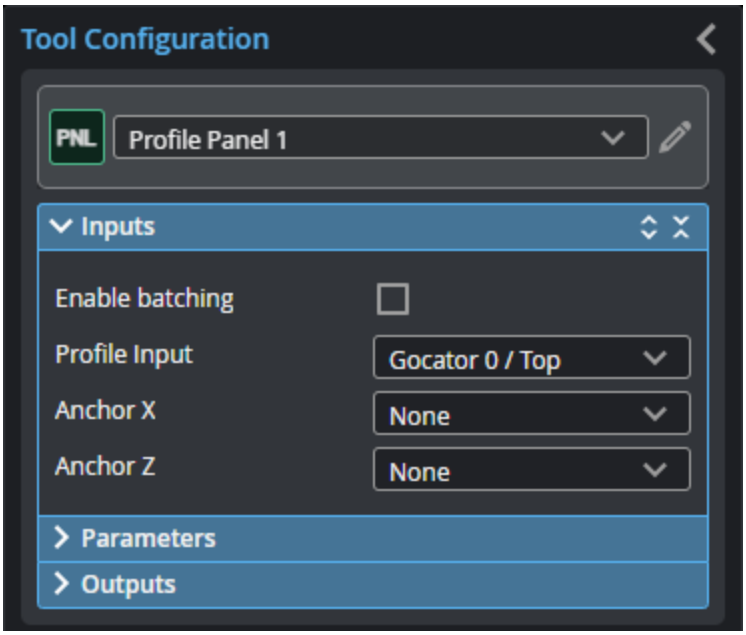
The Panel tool provides Gap and Flush measurements.

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 tool's parameters. For a detailed explanation of the algorithm, see *Gap and Flush Algorithms* on page 601.

> You must make sure that there are enough data points to define the edge in the profile, by properly setting up exposure, etc. If not, the algorithm will not function.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs



> To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

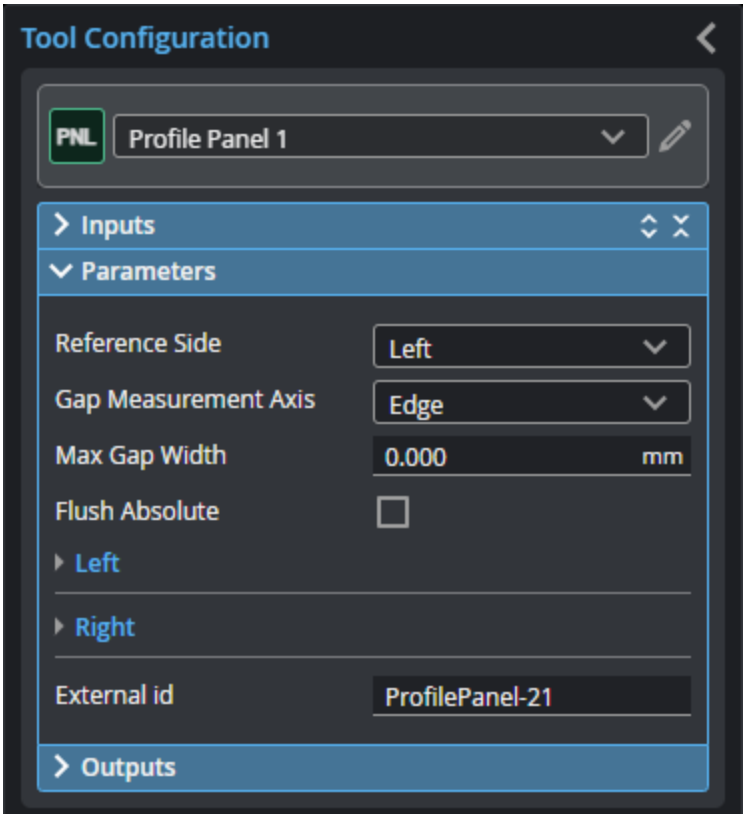
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with a maximum of two elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole.

Name	Description
	For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. If **Enable Batching** is *disabled* and the passed array contains more than two elements, GoPxL displays an error. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

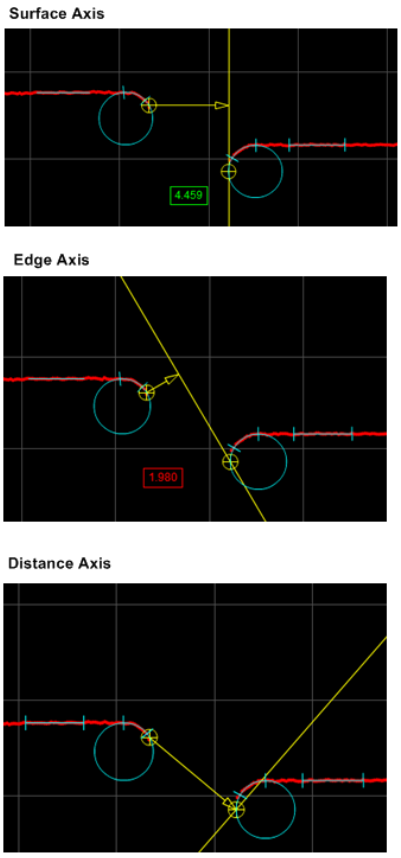
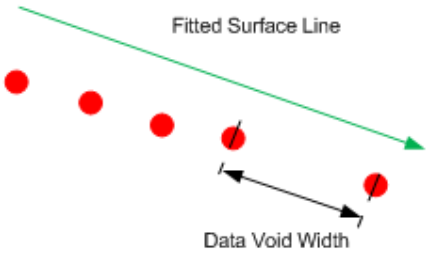
Parameters

You configure the tool's parameters in the expandable **Parameters** section.

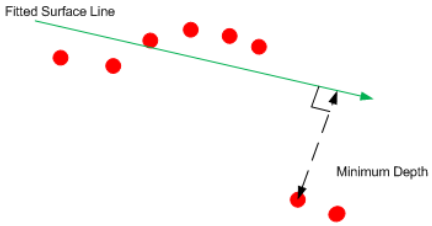
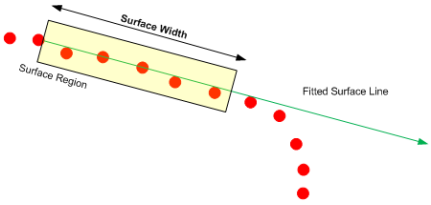
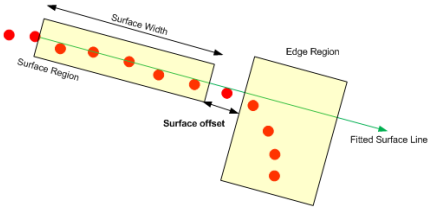
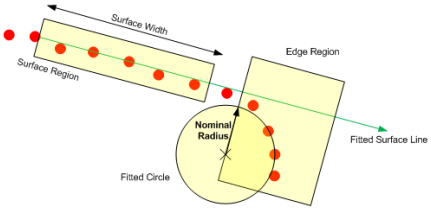


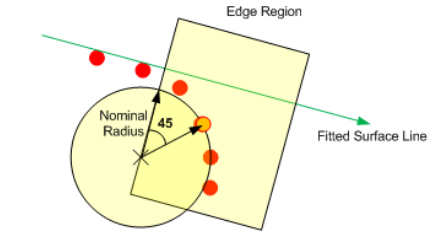
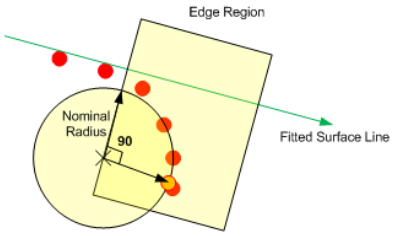
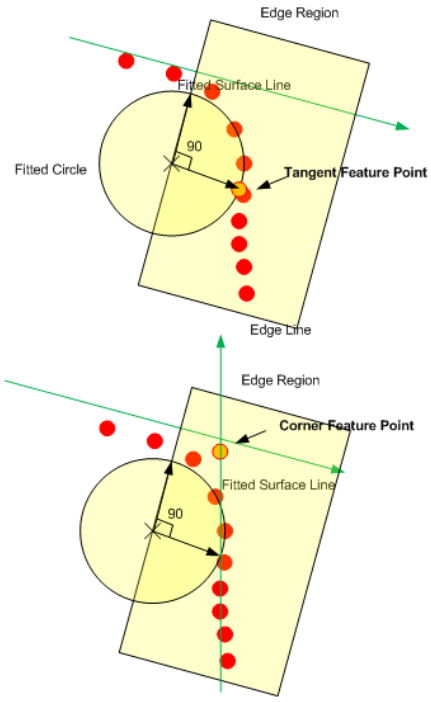
Parameters

Parameter	Description
Reference Side	Defines the side used to calculate the measurement axis (see below) rounded corner.
Gap 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.

Parameter	Description
	Edge: In the direction perpendicular to the edge of the reference surface. **Distance**: The Cartesian distance between the two feature locations. 
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. 
Flush Absolute *Flush measurement only*	When enabled, the Flash measurement returns an absolute value instead of a signed value.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

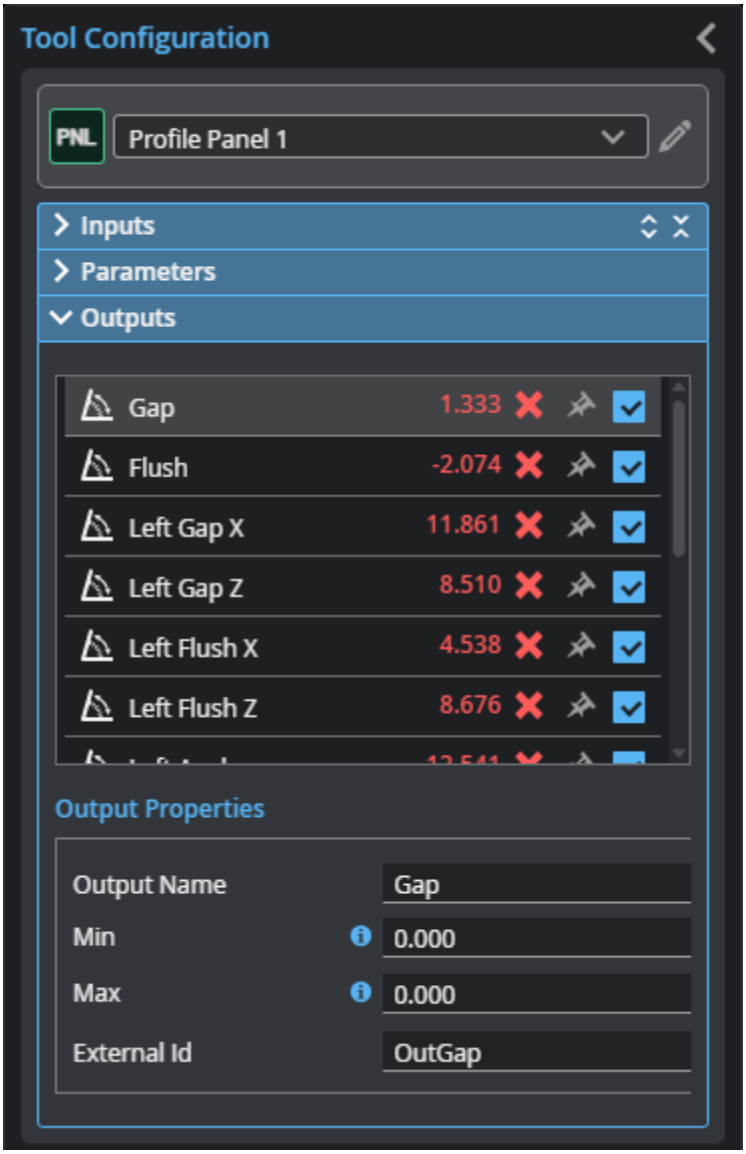
Left and Right Parameters

Parameter	Description
Min Depth	Defines the minimum depth before an opening can 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. Make sure the surface width does not go beyond the edge of the region of interest (that is, with **Region** enabled). 
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**. Make sure the surface offset does not move the surface width (see above) beyond the end of the region of interest (that is, with **Region** enabled).
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 the tools uses to calculate the feature point. The

Parameter	Description
	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 an edge line formed by interpolating the points at and after the tangent within the edge region.

Outputs

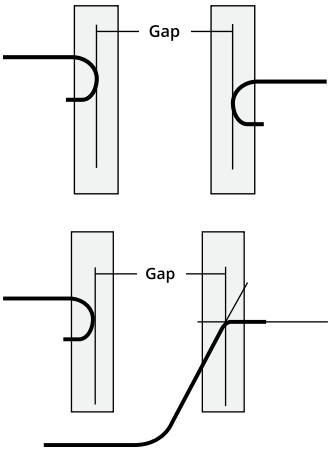
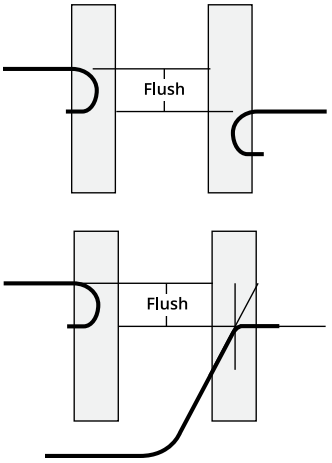
Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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 **Left Gap Z** The X and Z position of the edge feature on the left side used to measure the gap.	
Left Flush X **Left Flush Z** Returns the X and Z position of the feature on the left side used to measure flushness.	
Left Angle **Right Angle** The angle of the left and right side surface relative to the X axis.	
Right Gap X **Right Gap Z**	

Measurement	Illustration
Returns the X and Z position of the edge feature on the right side used to measure the gap.	
Right Flush X **Right Flush Z** Returns the X and Z position of the feature on the right side used to measure flushness.	

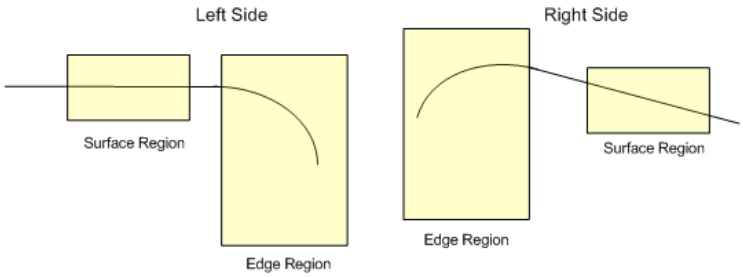
Data

Type	Description
Left Gap Right Gap	Point geometric features representing the point defined by Left Gap X and Left Gap Z, and Right Gap X and Right Gap Z, respectively. Use a side's **Edge Angle** parameter to choose where on the fitted circle the points fall.

Gap and Flush Algorithms

The Panel measurement tool uses the same algorithm to find a feature using either the Gap or the Flush measurement. The Round Corner tool uses the same algorithm, but applies it only to the left or the right; you must choose the side in the tool.

This algorithm first searches for two regions on a side: a surface region and an edge region. (See the tables below for the parameters used by the algorithm.)

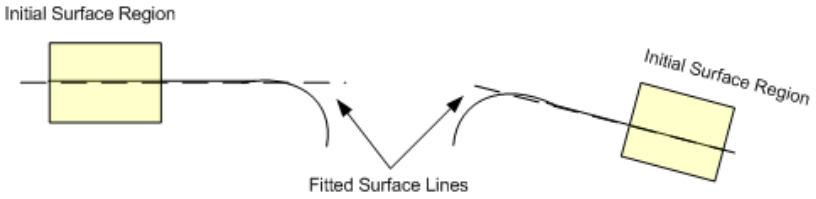


After the algorithm finds the regions, it places a feature point on the surface region based on a set of parameters. You can control the measurement regions, which contain the surface and the edge regions, for the left and the right side. A measurement region also defines the region in which the measurement tool will search for the feature points. Feature points are located on a side using the following algorithm.

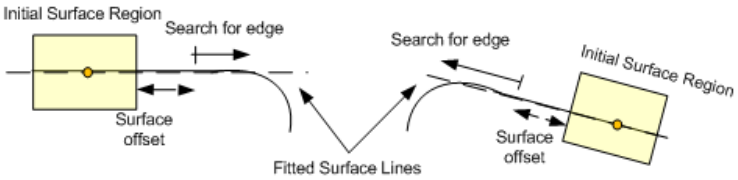
1. On the left side, search from left to right to find a surface region with data that covers at least the value specified in the **Surface Width** setting. For the right side, do the same, searching from right to left.



2. If a surface region is found, fit a line, called the surface line, using the data within the area.

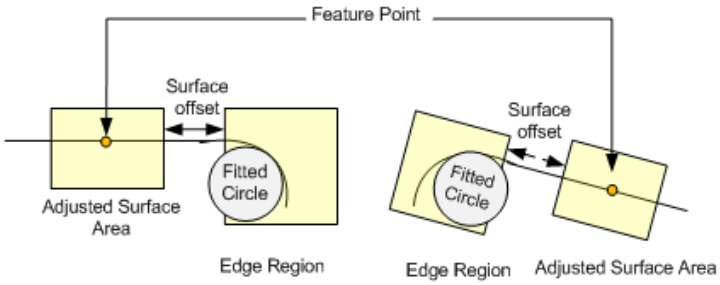


3. Search for a valid edge region that is located at least the distance specified in the **Surface Offset** setting from the end of the surface region. If a surface region is not found, move along the search direction and repeat step 1.



A valid edge region is detected when an edge matches the value in the **Nominal Radius** setting or when the depth exceeds the value in the **Min Depth** setting.

4. If a valid edge region is detected, a model fit is applied to the surface and edge regions to accurately determine the region positions and feature point locations. The model fit takes into account the **Surface Width**, **Surface Offset**, **Edge Angle** and the **Edge Type** parameters.

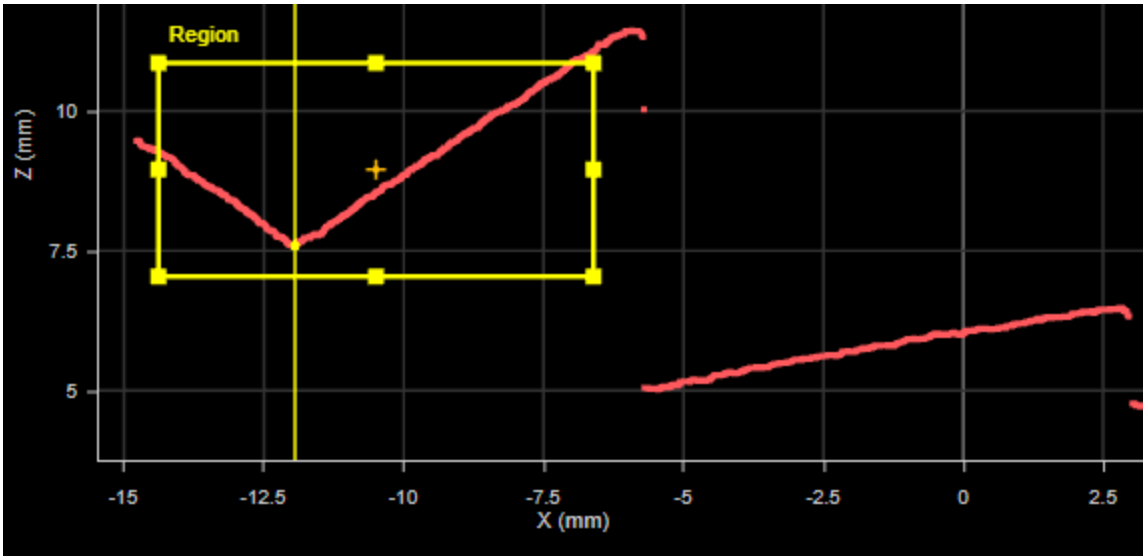


Profile Part Detection

This tool is not used on G3 sensors.

Profile Position

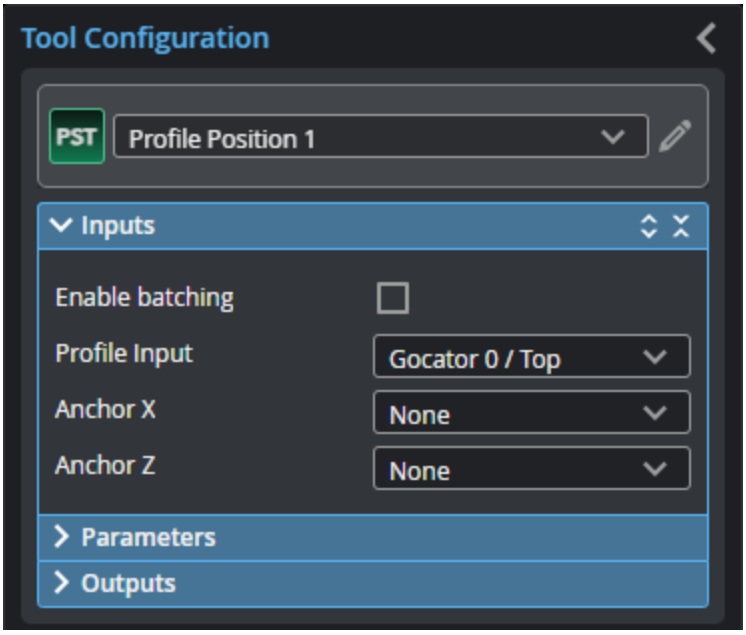
The Position tool finds the X or Z axis position of a feature point. The feature type must be specified (see below).



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool

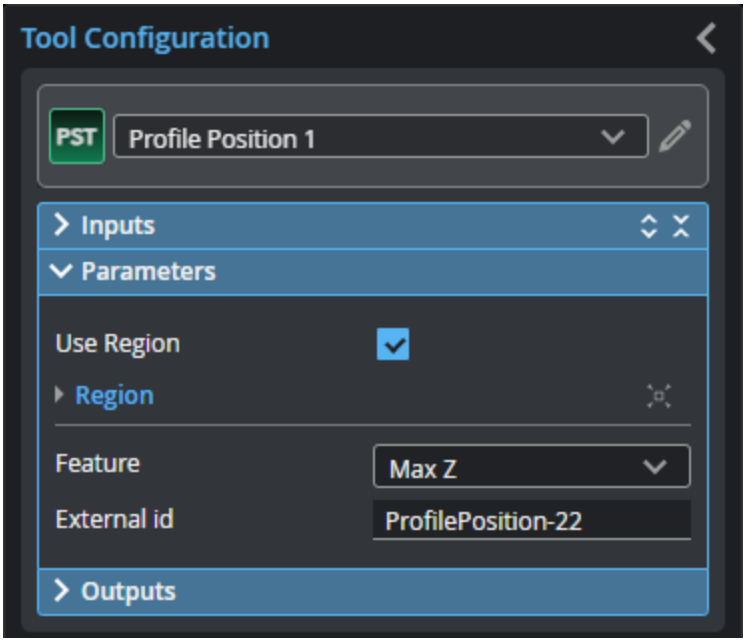
providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

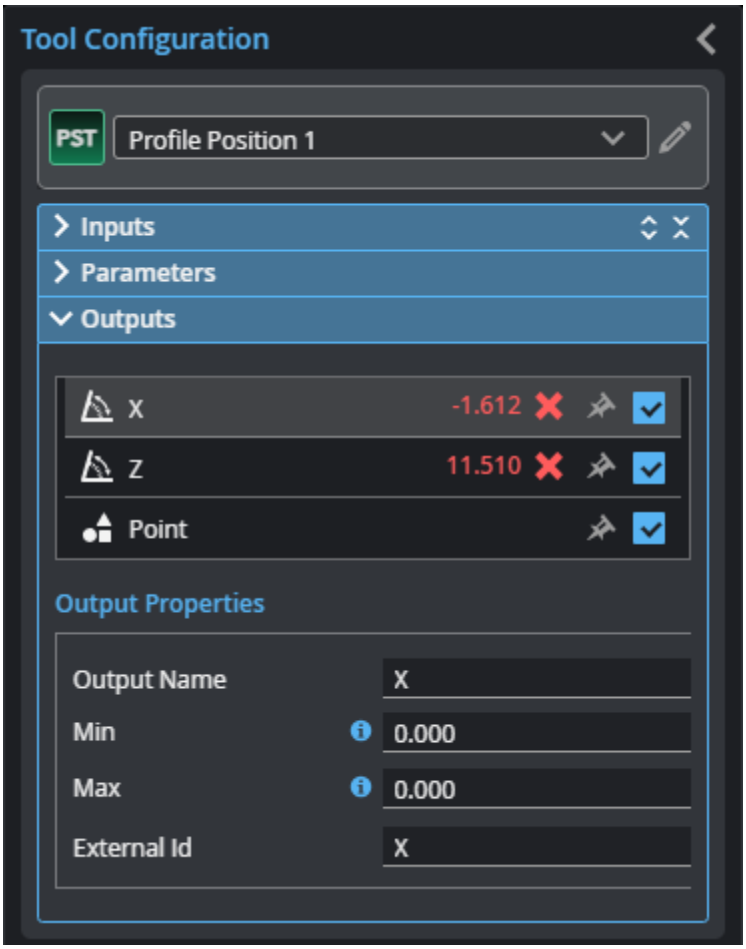


Parameters

Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
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 For more information on feature points, see *Feature Points* on page 224. To set the region of a feature, adjust it graphically in the data viewer, or expand the feature in the **Parameters** section and enter the values in the fields. For more information on regions, see *Regions* on page 214.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

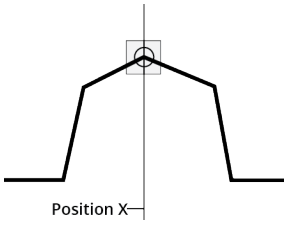


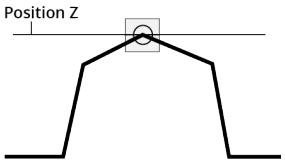
To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	Illustration
X Finds the position of a feature on the X axis.	

Measurement	Illustration
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 227.

Profile Roughness

The Profile Roughness tool calculates multiple measurements of profile roughness according to different industry standards, including the following:

ISO 21920

ISO 4287

ASME B46.1

JIS B0601

ISO 13565

ISO 12085

VDA 2006

VDA 2007

Parameters and measurements differ with standards. Choose the standard appropriate for your application.

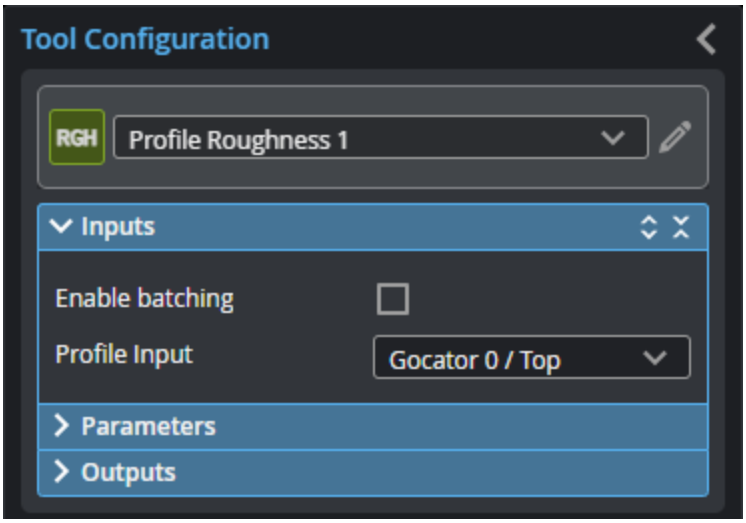
> There cannot be a large invalid point range for Profile roughness. Use the region to restrict the valid range in the tool for the calculation.

> Because the accuracy of a roughness measurement is limited by the X resolution of a sensor, this tool is typically only used on confocal sensors, whose high resolution is better suited for roughness applications.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

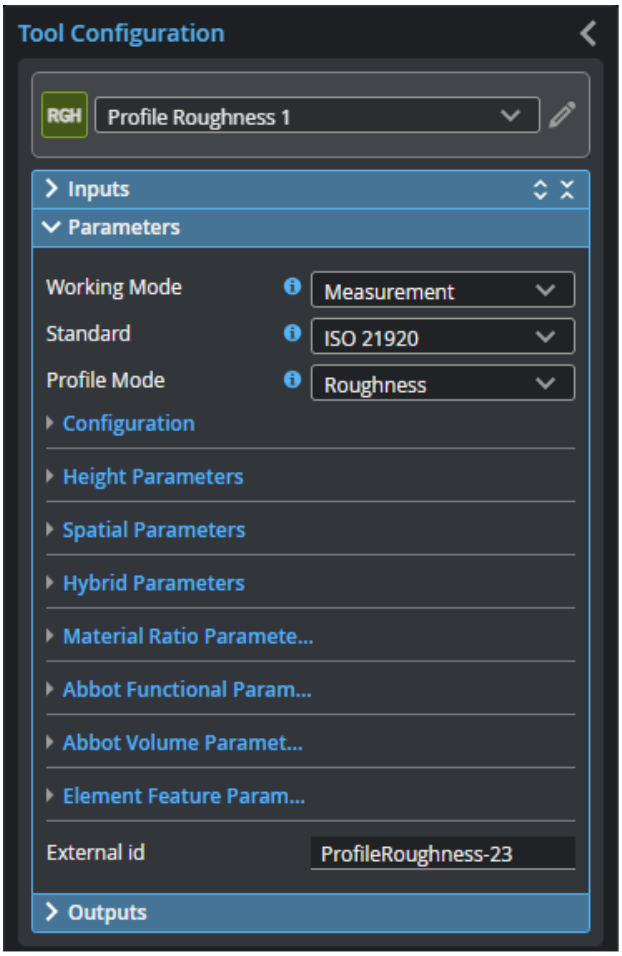
Inputs

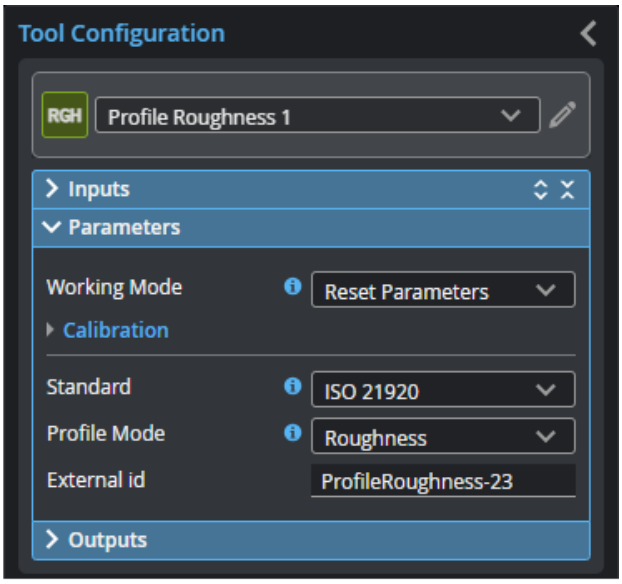
Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes.

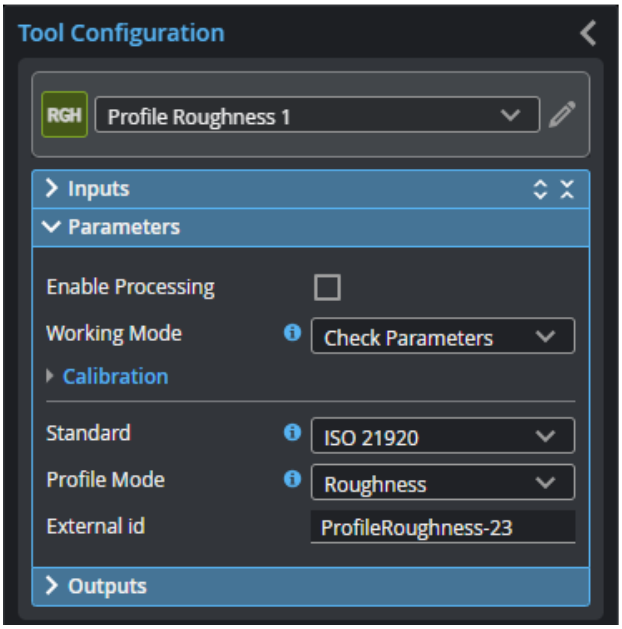
Parameters

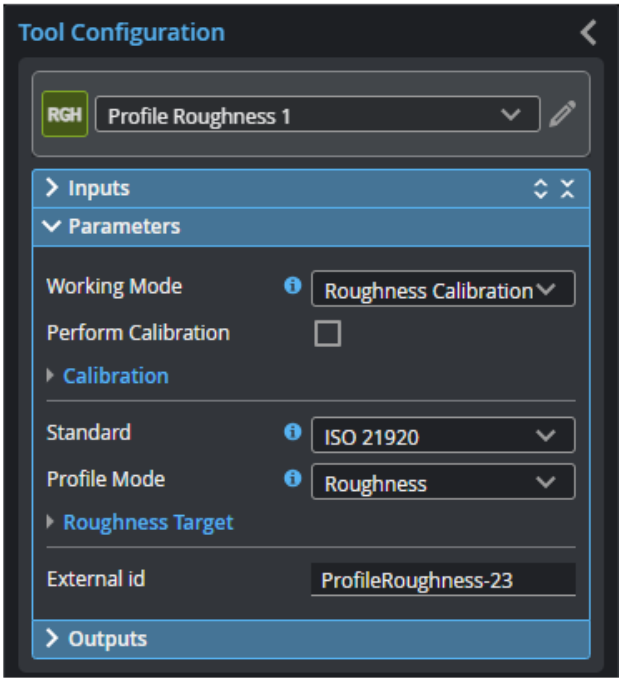
You configure the tool's parameters in the expandable **Parameters** section.

Note that the tool displays different sets of parameters based on what you set **Working Mode** to.


Tool Configuration
RGH
Profile Roughness 1
Inputs
Parameters
Working Mode
Measurement
Standard
ISO 21920
Profile Mode
Roughness
Configuration
Height Parameters
Spatial Parameters
Hybrid Parameters
Material Ratio Paramete...
Abbot Functional Param...
Abbot Volume Paramet...
Element Feature Param...
External id
ProfileRoughness-23
Outputs

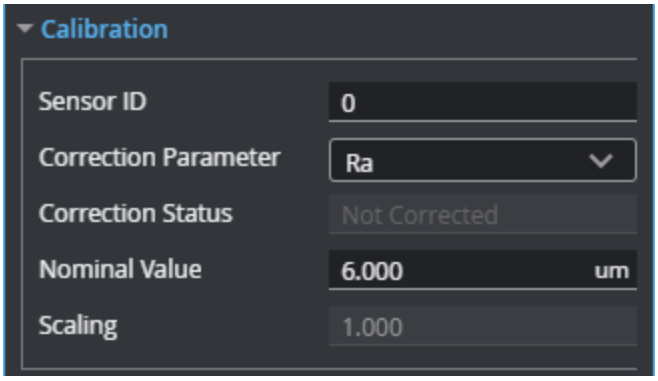

Tool Configuration
RGH
Profile Roughness 1
Inputs
Parameters
Working Mode
Reset Parameters
Calibration
Standard
ISO 21920
Profile Mode
Roughness
External id
ProfileRoughness-23
Outputs

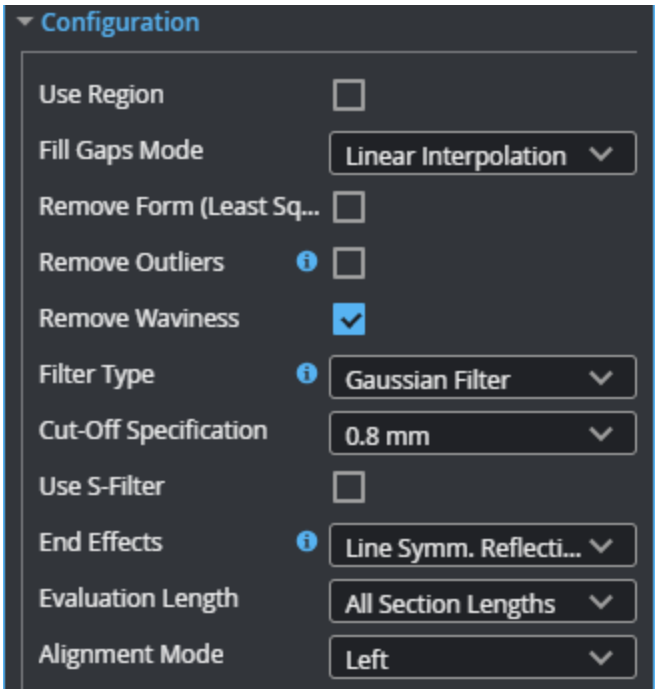

Tool Configuration
RGH
Profile Roughness 1
Inputs
Parameters
Enable Processing
Working Mode
Check Parameters
Calibration
Standard
ISO 21920
Profile Mode
Roughness
External id
ProfileRoughness-23
Outputs


Tool Configuration
RGH
Profile Roughness 1
Inputs
Parameters
Working Mode
Roughness Calibration
Perform Calibration
Calibration
Standard
ISO 21920
Profile Mode
Roughness
Roughness Target
External id
ProfileRoughness-23
Outputs

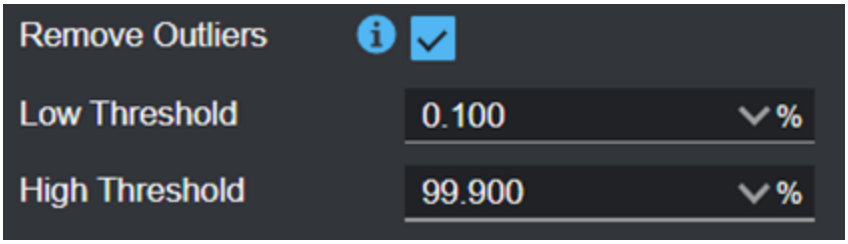
Parameters

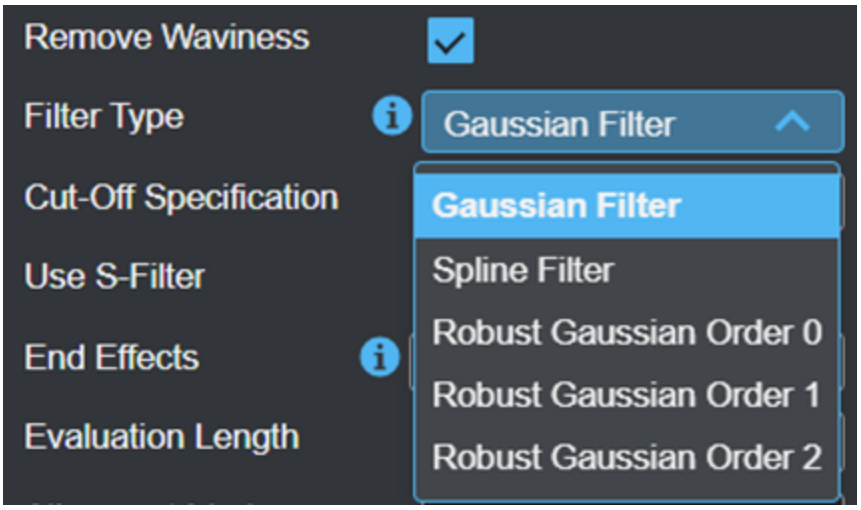
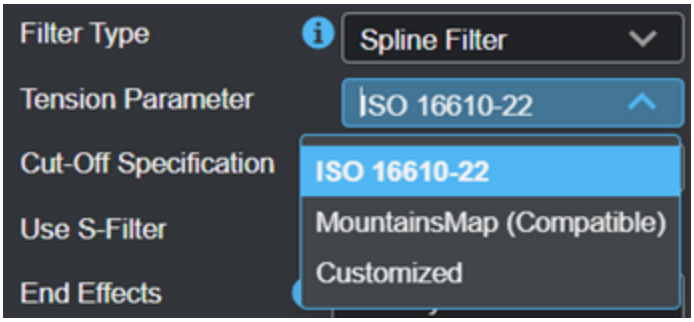
Parameter	Description
Enable Processing	The calculation starts only when this is checked. Only displayed when **Working Mode** is set to Check Parameters or Roughness Calibration. (Processing happens automatically in other modes.)
Working Mode	Support the following four working modes: • Measurement • Check Parameters • Reset Parameters • Roughness Calibration Before starting the calibration, make sure to remove the old calibration.
Perform Calibration	Performs the calibration. Only displayed when **Working Mode** is set to Roughness Calibration.
Standard	The standard the tool uses. Some standards change how options in **Evaluation Length** behave. For some standards, you should follow the recommendations below. **ISO 21920** Set **All Section Lengths** and **All Sections (no Averaging)**. These parameters behave the same with this standard. **ISO 4287** Typically, you will set **Evaluation Length** to "n Section Lengths", and set **Section Count** to 5. You can also choose "All Section Length" with this standard. **ASME B46.1** Typically, you will set **Evaluation Length** to "n Section Lengths", and set **Section Count** to 5. If you use a different **Evaluation Length** option, note that "All Section Lengths" and "All Sections (no Averaging)" are treated the same with this standard. **JIS B0601** According to the original definition, “All Section Lengths” should be used. **ISO 13565** “Total Profile Length” should be used. **ISO 12085** The same as definitions in ISO 4287, or using “Total Profile Length”.

Parameter	Description
	VDA 2006 “All Section Lengths” and "All Sections (no Averaging)" are the same. **VDA 2007** “Total Profile Length” should be used.
Profile Mode	In the standards, many parameters are also defined in relation to Primary or Waviness Profile. Select the profile mode after setting a standard.
Configuration	Shows when **Working Mode** is set to "Measurement". See *Configuration* on the next page for details.
Calibration	Shows when **Working Mode** is set to any mode other than "Measurement” 
Roughness Target	Shows when **Working Mode** is “Roughness Calibration”. The parameters are the same as the **Configuration** group.
Height Parameters	Shows different parameters depending on what you select in **Standard**. See *Height Parameters* on page 619.
Spatial Parameters	See *Spatial Parameters* on page 621 for details.
Hybrid Parameters	See *Hybrid Parameters* on page 621 for details.
Material Ratio Parameters	See *Material Ratio Parameters* on page 622 for details.
Abbot Functional Parameters	See *Abbot Functional Parameters* on page 622 for details.
Abbot Volume Parameters	See *Profile Roughness* on page 609 for details.
Element Feature Parameters	See *Element Feature Parameters* on page 624 for details.
Motif Parameters	See *Motif Parameters* on page 625 for details.

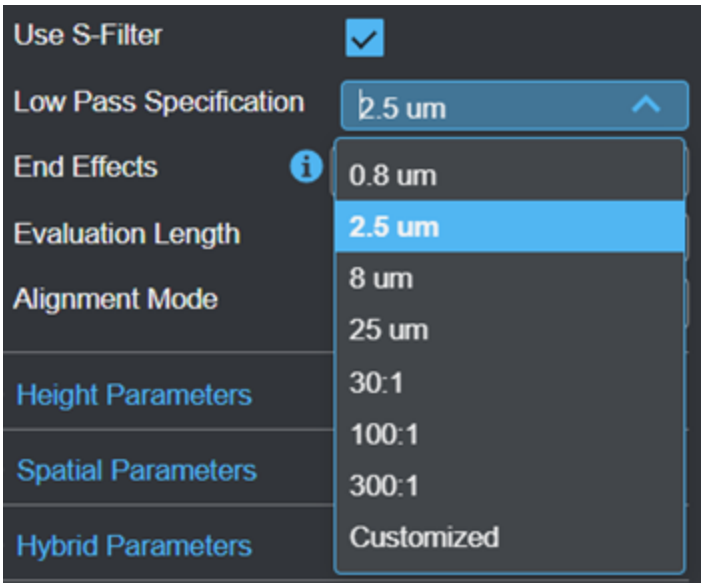
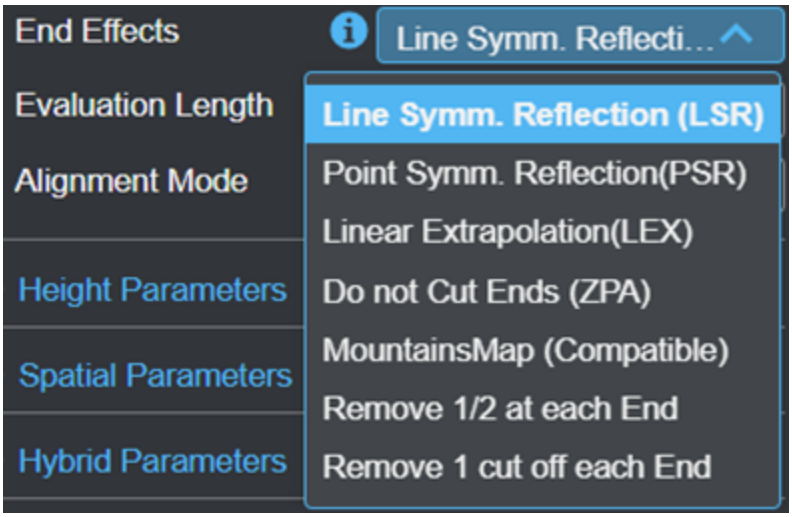


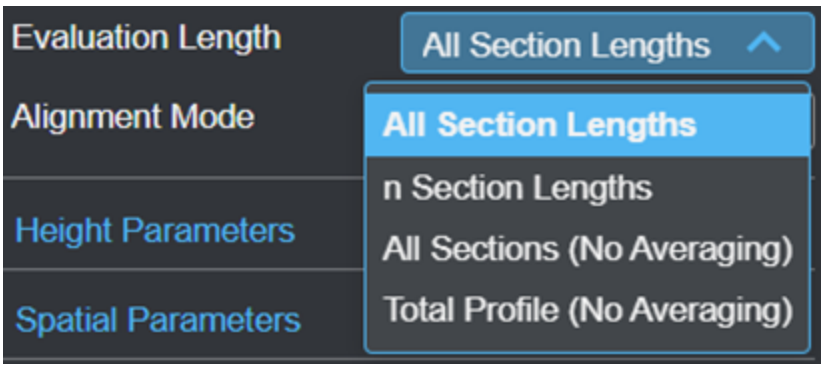
Configuration

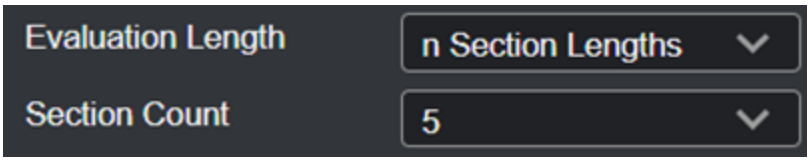
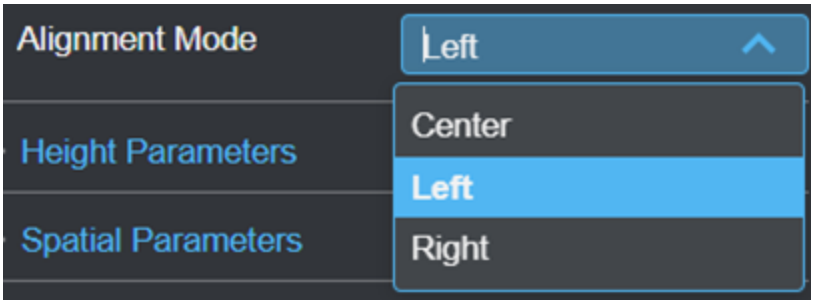
Parameter	Description
Use Region Region	Check to set region size.
Fill Gaps Mode	Choose **Linear Interpolation** to fill gaps in the profile but keep the profile as natural as possible.
Remove Form (Least Squares)	When checked, displays a drop-down containing one of the following: • Linear Alignment • Polynomial Subtraction: Show “Polynomial Order (<21)” for edit if selected • Circular Unwrapping • Circle With Nominal Radius: Show “Nominal Radius” for edit if selected.
Remove Outliers	Check to remove outliers from the profile and set Low Threshold and High Threshold. The outliers are defined according to the Sigma Rule in terms of the given percentage multiplied by 1.5. 
Remove Waviness	Check to remove the waviness of the profile. Select the filter in **Filter Type**.

Parameter	Description
	 The tool provides the following filters according to the ISO 16610 standards series, which allow different filter kernels having different shapes. Choose the filter depending on your application and its sensitivity to the curvature of the waviness. **Gaussian Filter**: According to ISO 16610-21, ISO 16610-28 If selected, set choose the appropriate option in **End Effects** (see *End Effects* on page 617). **Spline Filter**: According to ISO 16610-22, ISO 16610-28. If selected, set **Tension Parameter** to control how tightly the spline curve fits through the data points.  **Robust Gaussian Order 0**: According to ISO 16610-31, ISO 16610-28. Using Savitzky-Golay filter coefficients: $\alpha = \sqrt{ln(2)/\pi}$ Retaining the zeros moment means to reduce end effects.

Parameter	Description
	Robust Gaussian Order 1: According to ISO 16610-31, ISO 16610-28. Using Savitzky-Golay filter coefficients: $\alpha = \sqrt{ln(2)/\pi}$ Retaining the first moment means to reduce end effects. **Robust Gaussian Order 2**: According to ISO 16610-31, ISO 16610-28. Using Savitzky-Golay filter coefficients $\gamma = \sqrt{(-1 - W(-1/(2e))/\pi}$  The cut-off is selected depending on the workpiece surface either according to the valley spacing, or the expected roughness values. At the same time the total evaluation length and the corresponding total profile length are defined according to standards. Deviations are necessary if the workpiece does not allow the required total profile length. If the actual possible total profile length on the workpiece surface is not enough for lt, the number of sampling lengths is reduced accordingly and specified in the drawing. If the actual available total profile length is less than a sampling length, the total height of profile Pt of the primary profile is evaluated instead of Rt or Rz.
Use S-Filter	Check to use a low pass filter with the cut-off wavelength defined in **Low Pass Specification**.

Parameter	Description
	
End Effects	Shows when **Remove Waviness** is checked. The end effect is the unintended changes in filtration response in the both end sections of an open profile, and the end effect correction methods are according to standard ISO 16610-28.  **Line Symm. Reflection (LSR)** A measured profile is extended by horizontal reflection on the left hand and right hand, respectively. **Point Symm. Reflection (PSR)** A measured profile is extended by horizontal reflection in conjunction with vertical reflection on the left and right hand respectively. Both reflection lines shall

Parameter	Description
	intersect at the respective end point of the profile. **Linear Extrapolation (LEX)** In the case of linear extrapolation, a least-squares line is fitted to the profile within the left and right end effect regions. **Do not Cut Ends (ZPA)** Zero padding is a simple method for retaining the total length after filtering the profile. Profile z(x) is padded with zeros over the total length. **MountainsMap (Compatible)** Tries to make filtering compatible with MMP "Manage End Effect". This setting only applies to Guassian filters. **Remove 1/2 at each end** No end effects treatment required as the critical regions are not involved in the calculation. **Remove 1 cut off each end** No end effects treatment required as the critical regions are not involved in the calculation.
Evaluation Length	The evaluation length. One of the following:  **All Section Lengths** The maximum available sections will be used. **n section lengths** Set the number of sections in **Section Count**. The effective section number is always limited by the available length.

Parameter	Description
	 **All Sections (no Averaging)** All parameters as long as the definition allows are calculated in the entire evaluation length. **Total Profile (no averaging)** All parameters as long as the definition allows are calculated in the total profile length.
Alignment Mode	 Note: The alignment in MountainsMap is left-justified.

Height Parameters

Parameter	Description	Standard
Ra	**Roughness average** Ra is the arithmetic average of the absolute values of the roughness profile ordinates, in µm.	ISO 21920 ISO 4287 ASME B46.1 JIS B0601 VDA 2006 ISO 13565-2
Rq	**RMS roughness** Rq is the root mean square average of the roughness profile ordinates, in µm.	ISO 4287 ISO 21920 JIS B0601
Rz	**Mean roughness depth** Rz is the arithmetic mean value of the single roughness depths Rzi of consecutive sampling lengths, in µm.	ISO 21920 JIS B0601
Rz1max	**Maximum height** Rz1max is the greatest Rz value from the n sampling lengths lr, in	ISO 4287

Parameter	Description	Standard
	µm.	
Rz(JIS)	**The ten point height** RzJIS is defined on each sampling length as being the sum of the 5 highest peaks and the 5 deepest holes, divided by 5. The result is expressed in the length unit of the Z-axis, in µm.	JIS B0601
Rz(n)	Maximum height inside the nth section length, in µm.	ISO 4287 JIS B0601
Rmax	**Maximum roughness depth** Rmax is the largest single roughness depth within the evaluation length, in µm.	
Rt	**Total height of profile** Rt is the distance between the highest peak and the deepest valley of the profile of the total evaluation length ln, in µm.	ISO 21920 ISO 4287 JIS B0601
Rsk	**Skewness** Rsk is a measure of the asymmetry of the amplitude density curve. A negative skewness value indicates a surface with good bearing properties. (No unit.)	ISO 21920 ISO 4287 JIS B0601
Rku	**Kurtosis** Rku is a measure of the peakedness of the amplitude density curve. (No unit.)	ISO 21920 ISO 4287 JIS B0601
Rp Rpm	Rp is the height of the highest profile peak of the roughness profile within one sampling length. According to ASME, the Rp mean value (average calculated over the evaluation length) is called Rpm. Both in µm.	ISO 21920
Rv	Rv is the depth of the deepest profile valley of the roughness profile within one sampling length, in µm.	ISO 21920
Rz	The sum of Rp + Rv is the single roughness depth Rzi, in µm.	JIS B0601
Rzx	**Maximum height** (peak-to-valley), within a section length, in µm.	ISO 21920
Rc	**Mean height of profile elements** A horizontal discrimination of 1% of the sampling length and a vertical discrimination of 10% of Rz are applied to avoid taking into account segments with outlying values due to noise. In µm.	ISO 4287 JIS B0601
Rp1max	**Maximum local profile peak height** Rp1max is the Maximum of Rpi values which are calculated within a sampling length, in µm.	ISO 4287 JIS B0601
Rv1max	**Maximum local profile valley depth** Rv1max is the Maximum of Rvi values which are calculated within a sampling length, in µm.	ISO 4287 JIS B0601

Parameter	Description	Standard
Rz1max	**Maximum local height of the profile** Rz1max is the Maximum of Rzi values which are calculated within a sampling length, in µm.	
Rpt	Maximum hill height, in µm.	ISO 21920
Rvt	Maximum dale depth, in µm.	ISO 21920

Spatial Parameters

Parameter	Description	Standard
Ral	Autocorrelation length Ral is the horizontal distance of the autocorrelation function (tx) which has the fastest decay to a specified value s, with 0 < s < 1. The default value for s in the software is 0.2. This parameter expresses the content in wavelength of the surface. A high value indicates that the surface has mainly high wavelengths (low frequencies). In mm.	ISO 21920
Rsw	Dominant spatial wavelength Rsw is the wavelength corresponding to the largest value of the FFT. In mm.	ISO 21920
WDsm	Mean horizontal values determined from the amplitude spectrum Size of the profile elements (mean period length of the dominant waviness). In mm.	VDA 2007
WDc	Average of the heights of the profile elements within the evaluation length, in µm.	VDA 2007
WDt	Vertical difference between the highest and lowest point of the WD profile within the evaluation length, in µm.	VDA 2007

Hybrid Parameters

Parameter	Description	Standard
Rdq	**Root mean square gradient** Rdq is the root mean square value of the ordinate slopes dZ/dX, within the sampling length, in degrees.	ISO 21920
Rda	Arithmetic mean absolute gradient, in degrees.	ISO 21920
Rdt	Maximum absolute gradient, in degrees.	ISO 21920
Rdl	**Developed length** Rdl is the Length measured following the profile curve, in mm.	ISO 21920
Rdr	**Developed length ratio** Rdr is the Ratio between the length measured following the profile curve ("real profile"), and the profile length, expressed as a percentage in excess of 100%.	ISO 21920

Material Ratio Parameters

Parameter	Description	Standard
Rmr	**Material ratio** Rmr indicates what ratio the totaled length in the material has assumed relative to the evaluation length (in %). The comparison is made in the specified section height c and the total evaluation length ln. The material ratio curve indicates the material ratio as a function of the section height.	ISO 21920 ISO 4287 JIS B0601
Rmr (Rz/4)	**Automatic relative material ratio** Rmr (Rz/4) is the Material ratio calculated at cut level c = 0.25 Rz, relative to a specified reference level, as a percentage.	ISO 4287 JIS B0601
Rmc	**Inverse material ratio** Rmc is the height distance between highest level and the cut level c corresponding to a given material ratio, in µm.	ISO 21920
Rdc	**Profile section height difference** Rdc is the Profile section height difference or bearing height. Gives the vertical distance between two cut levels given by their bearing ratio, in µm.	ISO 4287 ISO 21920 JIS B0601

Abbot Functional Parameters

Parameter	Description	Standard
Rk	**Core roughness height** Rk is the effective roughness of the profile. It is the height of the roughness profile, without taking into account the elevated hills or the very deep dales, in µm.	ISO 13565-2 ISO 21920
Mr1	**Upper material ratio** Mr1 is the smallest material ratio at the limits of the roughness core area, as a percentage.	ISO 13565-2 ISO 21920
Mr2	**Lower material ratio** Mr2 is the highest material ratio at the limits of the roughness core area, as a percentage.	ISO 13565-2 ISO 21920
Rpk	**Reduced peak height** Rpk is the mean height of the peaks protruding from the roughness core profile, in µm.	ISO 13565-2 ISO 21920
Rpkx Rpk*	**Maximum peak height** Rpkx is the roughness height of the hills, before area correction, in µm.	ISO 13565-2 ISO 21920
Rvk	**Reduced valley depth** Rvk is the mean depth of the valleys protruding from the roughness core profile, in µm.	ISO 13565-2 ISO 21920

Parameter	Description	Standard
Rvkx Rvk*	**Maximum dale depth** Rvkx is the roughness depth of the dales, before area correction, in µm.	ISO 13565-2 ISO 21920
A1	**Upper area** A1 is calculated as the area of the triangle equivalent to the peaks, $\mu m^2/mm$.	ISO 13565-2
A2	**Lower area** A2 is calculated as the area of the triangle equivalent to the holes, $\mu m^2/mm$.	ISO 13565-2
CV	**Crevice volume** Equivalent to A2, but expressed in specific units ($\mu m^3/\mu m^2$). Note: 10 $\mu m^3/\mu m^2$ = 1 mm^3/cm^2, so the CV value is 10 times the A2 value.	
APH	**Peak height** APH is the maximum height of the protruding peaks above the roughness core profile, in µm.	ISO 13565-2
AVH	**Valley depth** AVH is the maximum depth of the profile valleys below the roughness core profile, in µm.	ISO 13565-2
Rvk/Rk	**Valley-core depth ratio** Rvk/Rk is the ratio of the reduced valley depth Rvk over the core roughness depth Rk. (No unit.)	ISO 13565-2
Rvq	**Valley root-mean square roughness** Rvq is the slope of the linear regression calculated on the dale zone, in µm.	ISO 13565-3 ISO 21920
Rpq	**Plateau root-mean square roughness** Rpq is the slope of the linear regression calculated on the hill zone, in µm.	ISO 13565-3 ISO 21920
Rmq	**Material ratio** Rmq is the Intersection of both regression lines. Gives information about the depth of the intersection between fine surface finish (plateau zone) and rough surface finish (valley zone), as a percentage.	ISO 13565-3 ISO 21920

Abbot Volume Parameters

Parameter	Description	Standard
Rvm	**Material volume** Rvm is the volume of material at a material ratio p (as a percentage), in $\mu m^3/mm^2$.	ISO 21920
Rvv	**Void volume**	ISO 21920

Parameter	Description	Standard
	Rvv is the volume of voids at a material ratio p (as a percentage), in $\mu m^3/mm^2$.	
Rvmp	**Hill material volume** Rvmp is the volume of material in the hills, between 0% material ratio and a material ratio p (as a percentage), calculated in the zone above c1, in $\mu m^3/mm^2$.	ISO 21920
Rvmc	**Core material volume** Rvmc is the volume of material in the core, between two material ratios p and q (as a percentage), calculated in the zone between c1 and c2, in $\mu m^3/mm^2$.	ISO 21920
Rvvc	**Core void volume** Rvvc is the volume of void in the core, between two material ratios p and q (as a percentage), calculated in the zone between c1 and c2, in $\mu m^3/mm^2$.	ISO 21920
Rvvv	**Dale void volume** Rvvv is the volume of void in the dales, between a material ratio p (as a percentage) and 100% material ratio, calculated in the zone below c2, in $\mu m^3/mm^2$.	ISO 21920

Element Feature Parameters

Parameter	Description	Standard
Rsm	**Mean width of the profile elements** Rsm gives information about the spacing of motifs (length of the period), in mm.	ISO 4287 ISO 21920 JIS B0601
Rsmx	Maximum profile element width, in mm.	ISO 21920
Rsmq	**Standard deviation of the profile element widths** Rsmq is the root mean square deviation of widths, in mm.	ISO 21920
Rc	Mean height of the profile elements, in µm.	ISO 21920
Rcx	Maximum height of the profile elements, in µm.	ISO 21920
Rcq	**Standard deviation of the profile element heights** Rcq is the root mean square deviation of heights, in µm.	ISO 21920
RPc	**Peak count parameter** RPc corresponds to the number of mean widths of profile elements (XSm) inside a given length (default length 10 mm). (No unit.)	ISO 4287 ISO 21920 JIS B0601

The principle of the Motif standard consists of looking for local peaks and valleys in the primary profile, and associating one valley with the closest preceding and following peaks in order to create a Motif. Several iterative combinations of two Motifs each assure that the most important Motifs, the

width of which fall below the limit A, are considered. If not otherwise specified, the default value is A = 0.5 mm. The limit A has a similar function as the cut-off in the Gaussian filtering.

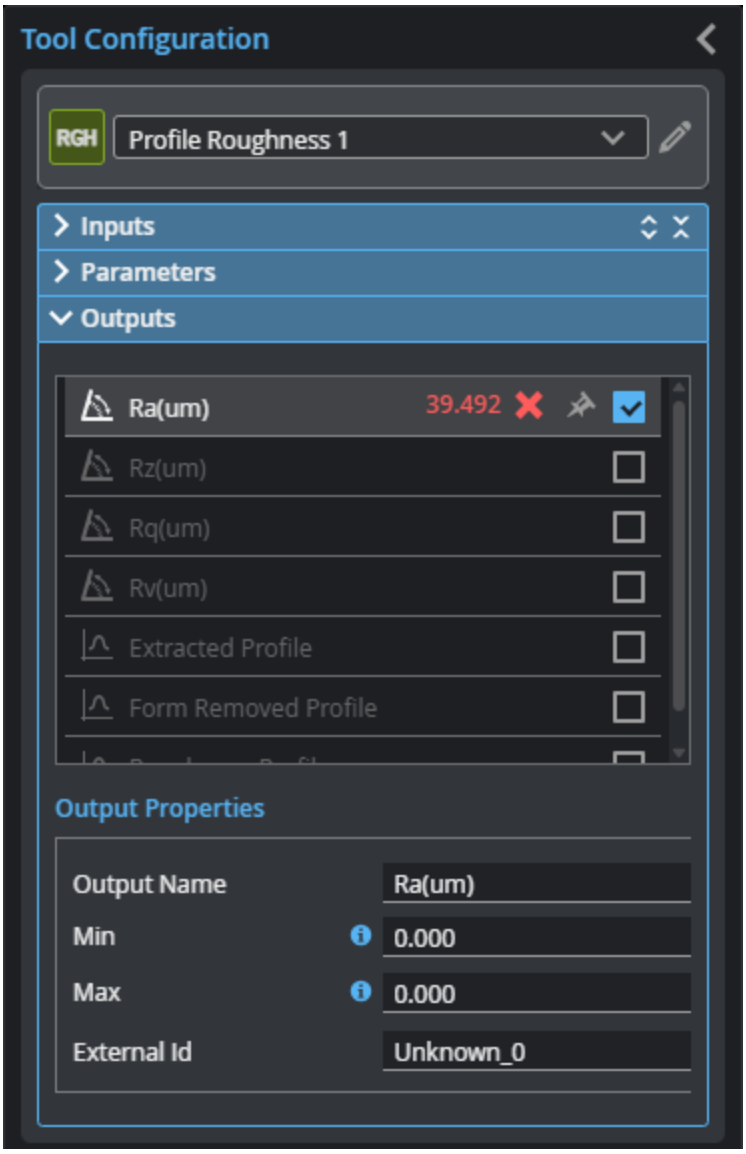
Motif Parameters

Parameter	Description	Standard
R	**Mean depth of roughness Motifs** R is the arithmetic mean value of the depths Hj of the roughness Motifs within the evaluation length, in µm.	ISO 12085
AR	**Mean spacing of roughness Motifs** AR is the arithmetic mean value of the lengths ARi of the roughness Motifs within the evaluation length, in mm.	ISO 12085
Rx	**Maximum depth of the roughness motifs** Rx is the deepest depth Hj within the evaluation length, in µm.	ISO 12085
Rt	Total height of the profile, in µm.	ISO 12085
Kr	Mean slope of the roughness motifs. (No unit.)	ISO 12085
Nr	**Number of roughness motifs** If the number of roughness motifs is less than 3, only the Rx and the Pt can be computed. (No unit.)	ISO 12085
SR	**Standard deviation of the depths of the roughness motifs** SR computes the dispersion of the Ri values, in µm.	ISO 12085
SAR	**Standard deviation of the spacings of the roughness motifs** SAR computes the dispersion of the ARi values, in mm.	ISO 12085
W	**Mean depth of waviness Motifs** W is the arithmetic mean value of the depths Hj of the roughnes Motifs within the evaluation length, in µm.	ISO 12085
AW	**Mean spacing of waviness Motifs** AW is the arithmetic mean value of the lengths ARi of the roughness Motifs within the evaluation length, in mm.	ISO 12085
Wx	**Maximum depth of the waviness motifs** Wx is the biggest of all Wi within the evaluation length, in µm.	ISO 12085
Wte	Total depth of waviness, in µm.	ISO 12085
Kw	Mean slope of the waviness motifs. (No unit.)	ISO 12085
Nw	**Number of waviness motifs** If the number of roughness motifs is less than 3, only the Wx and the Wte can be computed. (No unit.)	ISO 12085
SW	**Standard deviation of the depths of the waviness motifs** SW calculates the dispersion of the Wi values, in µm.	ISO 12085
SAW	**Standard deviation of the spacings of the waviness motifs** SAW calculates the dispersion of the AWi values, in mm.	ISO 12085

Parameter	Description	Standard
Trc	**Micro Geometric material ratio** Trc is the bearing fraction at a given depth c of the envelope leveled profile, as a percentage.	ISO 12085
HTrc	Micro geometric criterion, in µm.	
Rke	Kernel roughness of the envelope leveled profile, in µm.	
Rpke	Reduced height of the envelope leveled profile, in µm.	
Rvke	Reduced depth of the envelope leveled profile, in µm.	

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

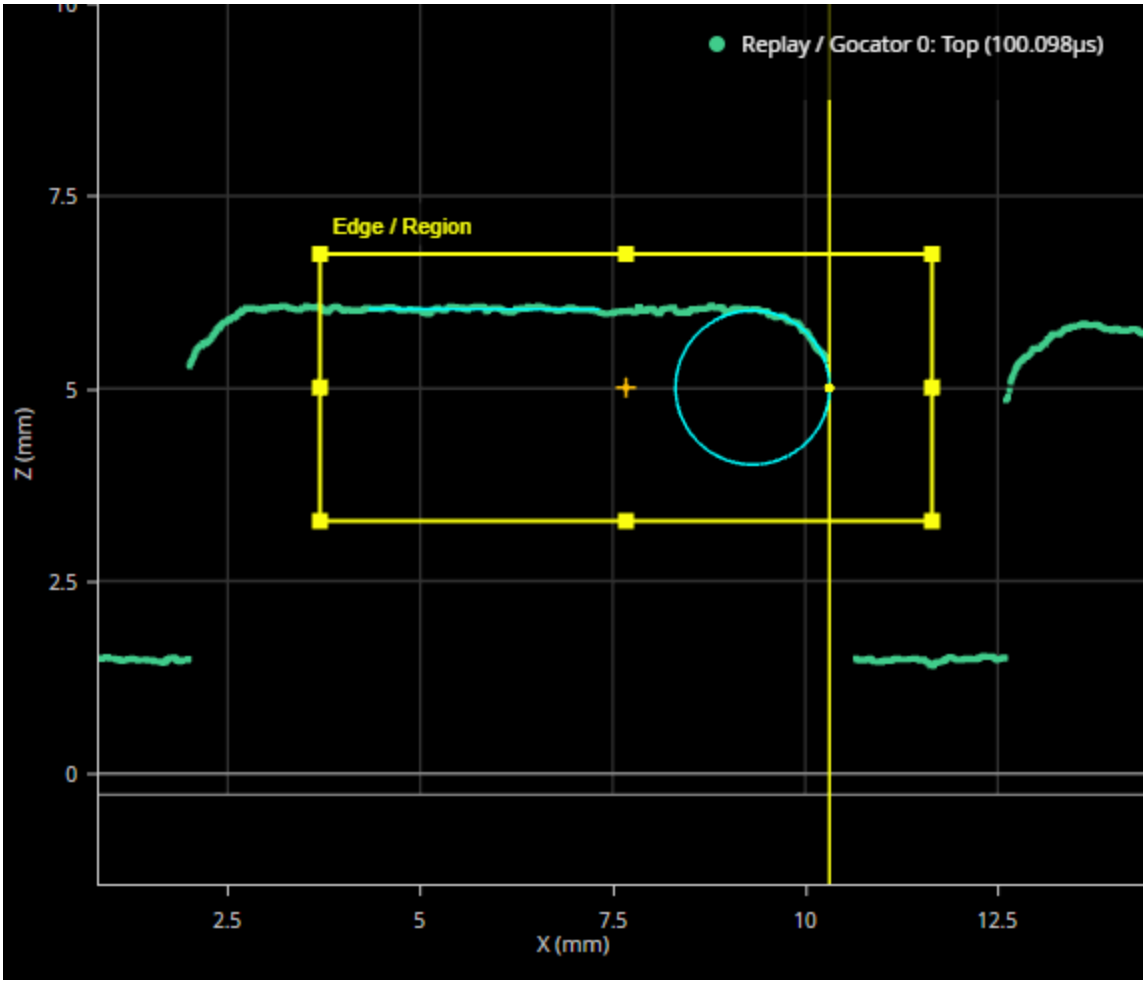
Measurement
Ra Arithmetic average of the absolute values of the roughness profile ordinates. For the other measurements, see the parameter details above.

Data

Type	Description
Extracted Profile	Extraction of the measured profile with the defined region.
Form Removed Profile	Obtained after removal of the nominal form (F-operation).
Roughness Profile	Obtained after filtering including a (high-pass) L-filter (λc) rejecting long wavelengths. When presenting the roughness profile, the mean line is the zero line.

Profile 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.



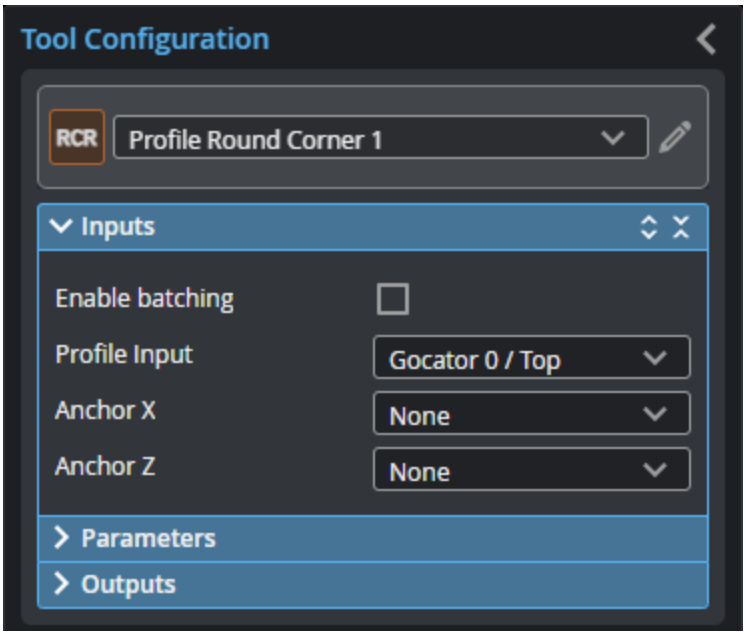
For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

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 tool's parameters. For a detailed explanation of the algorithm, see *Gap and Flush Algorithms* on page 601.

> You must make sure that there are enough data points to define the edge (proper exposure, etc.). If not, the algorithm will not function.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



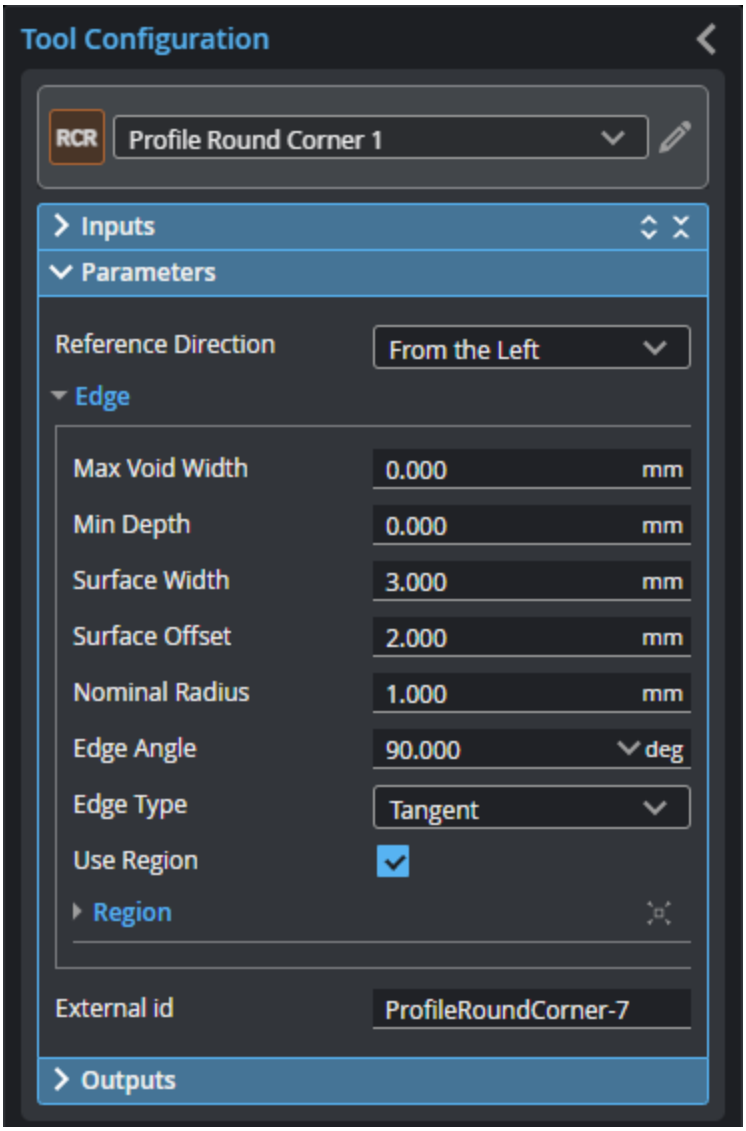
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with a maximum of two elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. If **Enable Batching** is *disabled* and the passed array contains more than two elements, GoPxL displays an error. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

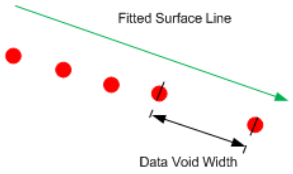
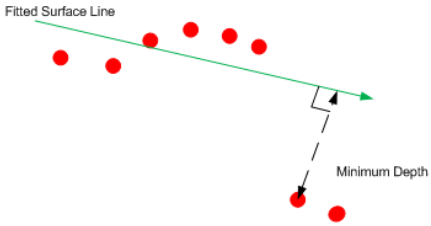
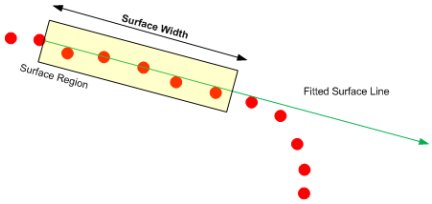
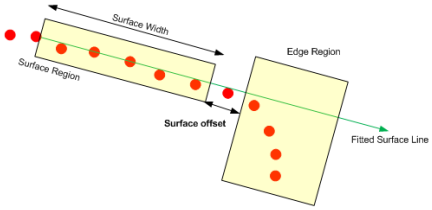
Parameters

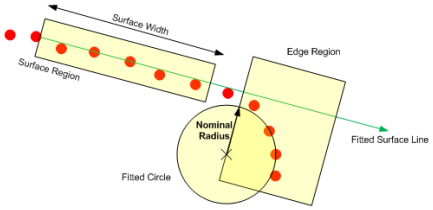
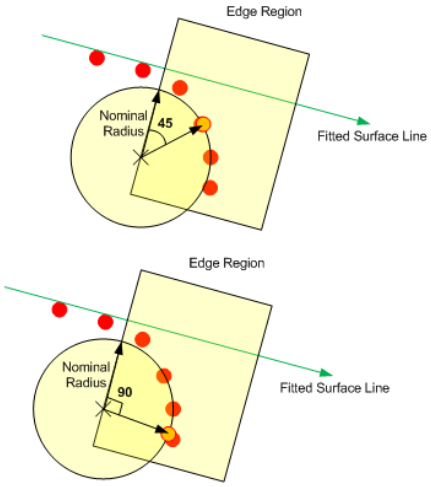
You configure the tool's parameters in the expandable **Parameters** section.

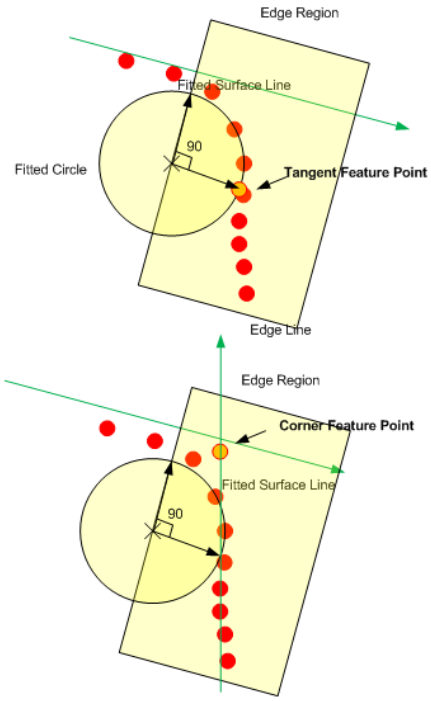


Parameters

Parameter	Description
Reference Direction	Defines the side used to calculate the rounded corner.
Edge	When expanded, displays the parameters for defining the edge (see below).
Max Void Width	The maximum allowed width of missing data caused by occlusion or data dropout. A larger value prevents the algorithm from registering a section of missing data as an edge. Setting the value to 0 causes the algorithm to try to detect an edge in every missing data section.

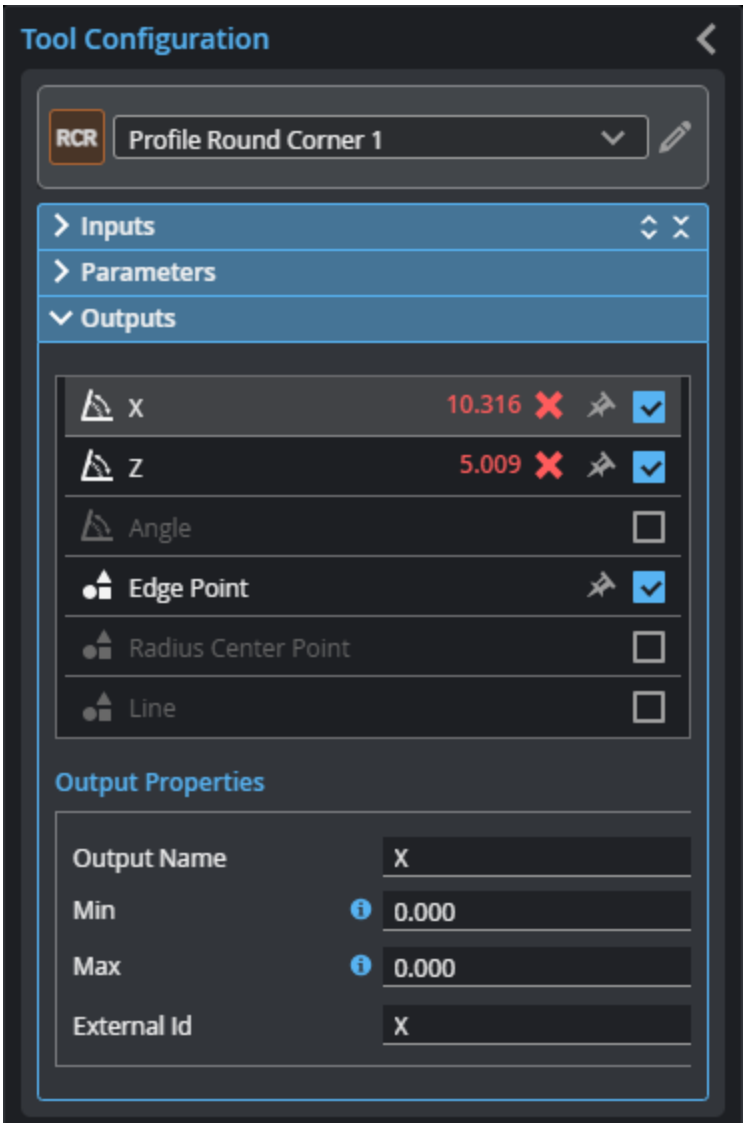
Parameter	Description
	
Min Depth	Defines the minimum depth before an opening could be considered to have a potential edge. The depth is the perpendicular distance from the fitted surface line. 
Surface Width	The width of the surface area in which data is used to form the fitted surface line. This value 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. The algorithm searches for a start position in which the remaining data most resemble a circle of the specified nominal radius.

Parameter	Description
	
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.

Parameter	Description
	
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

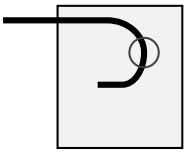
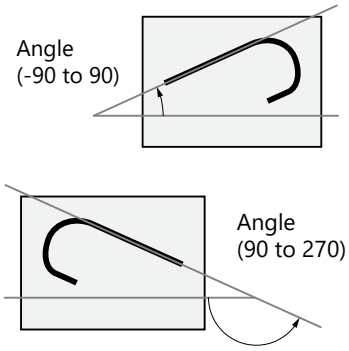


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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**, above), with respect to the X axis. Left edge angles are from -90 to 90. Right edge angles are from 90 to 270.	

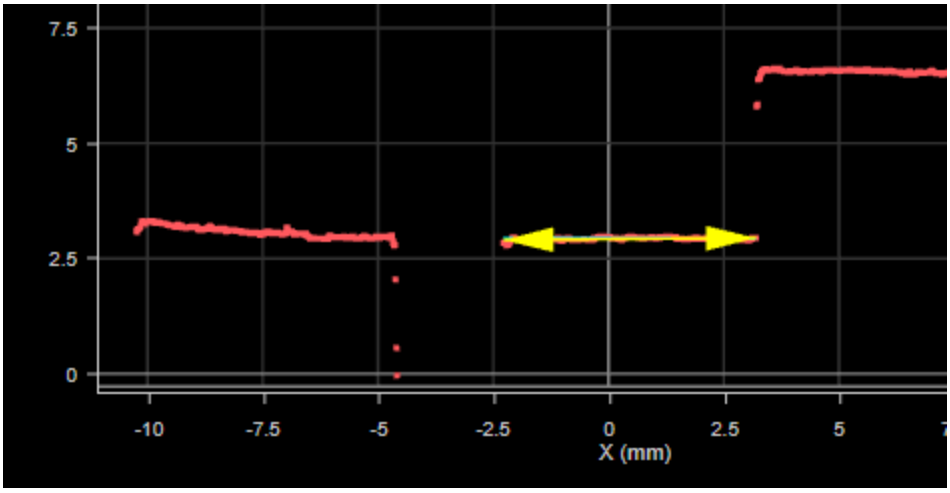
Features

Type	Description
Edge Point	The position of the edge.
Radius Center Point	The center of the radius.
Line	The fitted surface line.

For more information on geometric features, see *Geometric Features* on page 227.

Profile Strip

The Strip tool returns various measurements related to a strip, such as its X and Z positions, width, and height.



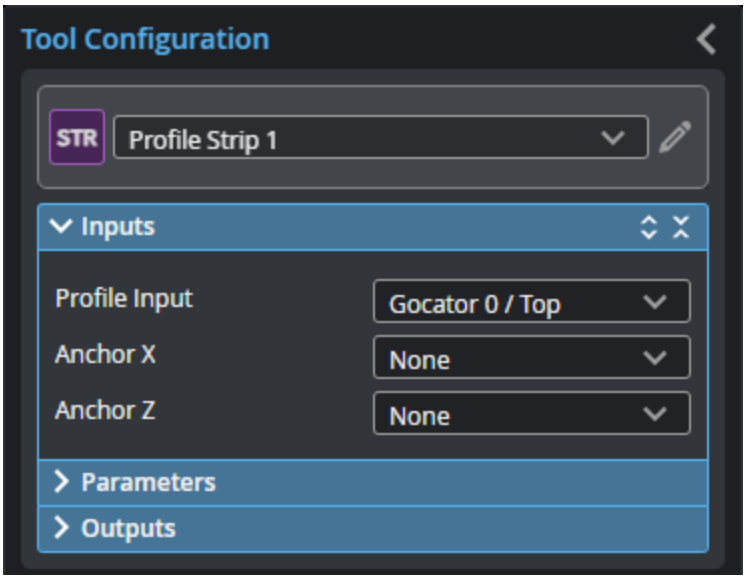
The tool lets you optionally set an index to return the measurements of a specific strip when more than one is visible in the region of interest. If you need to return multiple strips from a profile, add a Profile Strip tool for each strip, configure it to represent the desired strip, and set its **Selection Index** parameter to the desired strip (0-based index).

The Strip tool uses a complex feature-locating algorithm to find a strip and then return measurements. For a detailed explanation of the algorithm, see *Strip Algorithm* on page 643. The behavior of the algorithm can be adjusted by changing the tool's parameters.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



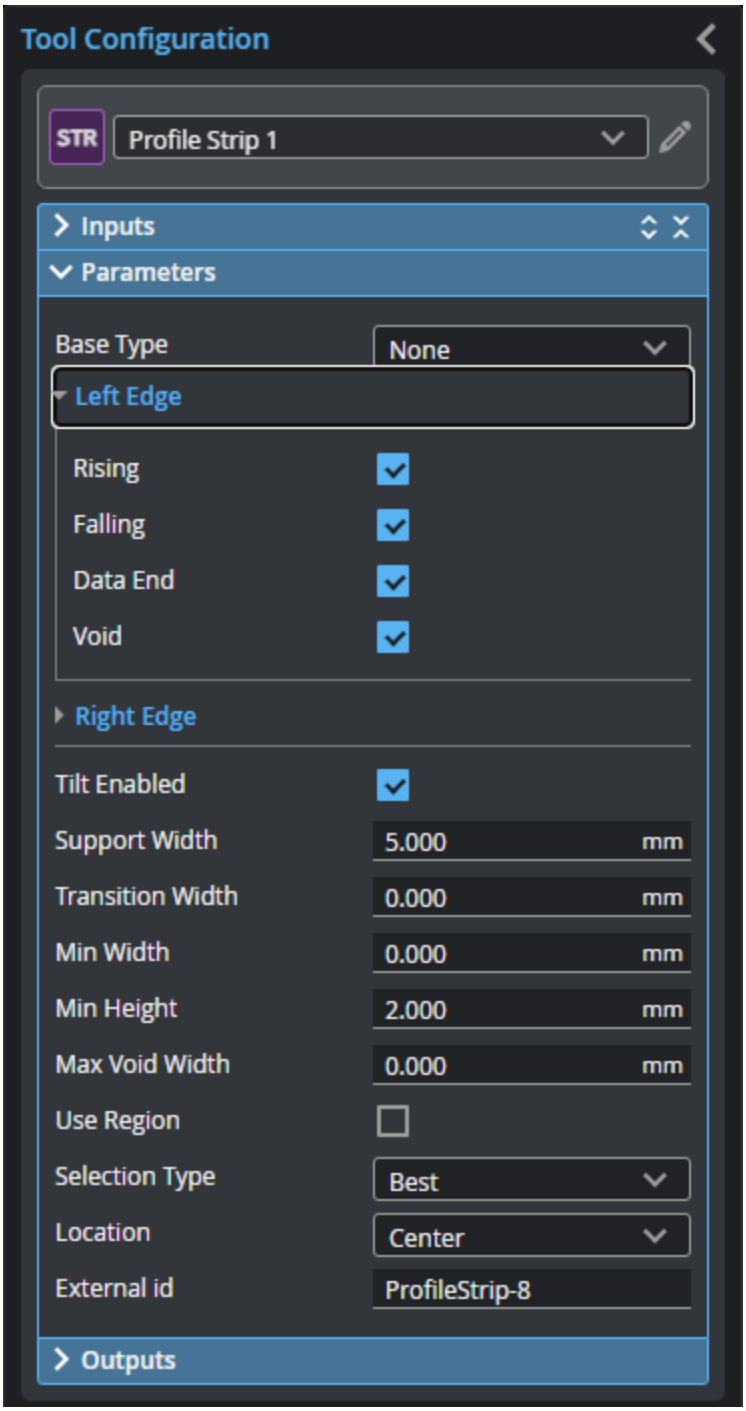
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Profile Input	The data the tool applies measurements to or processes.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

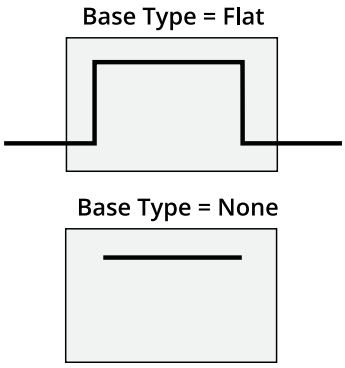
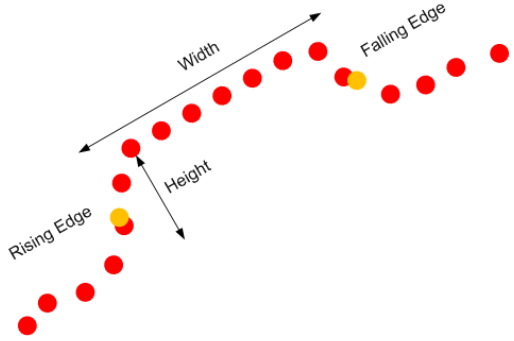
Parameters

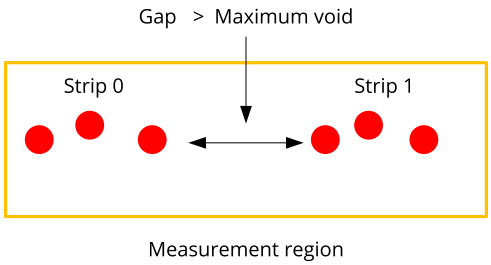
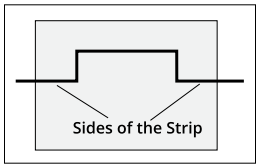
You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Base Type	Affects detection of rising and falling edges.

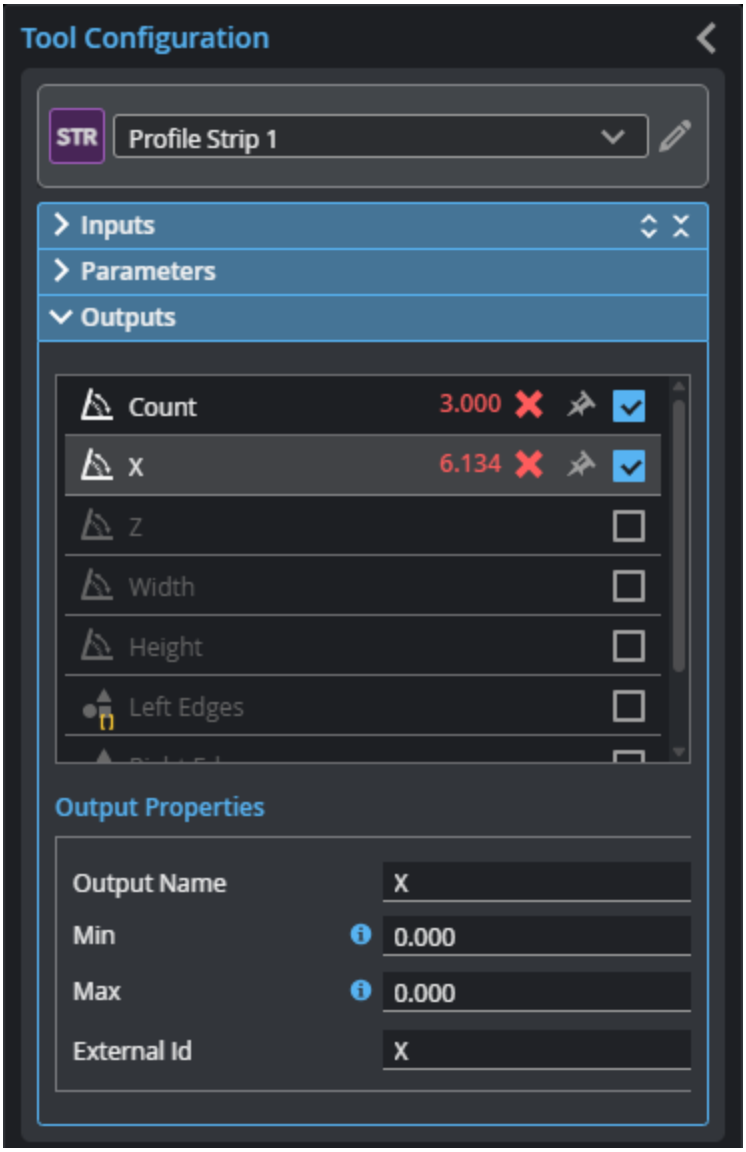
Parameter	Description
	 When **Base Type** is set to **Flat**, both strip (raised area) and base support regions are needed. When set to **None**, only a point that deviates from a smooth strip support region is needed to find a rising or falling edge.
Left Edge Right Edge	Specifies the features that 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. For the definitions of these conditions, see *Strip Start and Terminate Conditions* on page 643.
Tilt Enabled	Enables and 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, corrected by 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. For information on how this parameter is used by different base types, see *Strip Step Edge Definitions* on page 645.
Transition Width	Specifies the nominal width needed to make the transition from the base to the

Parameter	Description
	strip. For information on how this parameter is used by different base types, see *Strip Step Edge Definitions* on page 645.
Min Width	Specifies the minimum width for a strip to be considered valid.
Min Height	Specifies the minimum deviation from the strip base. For information on how this parameter is used by different base types, see *Strip Step Edge Definitions* on page 645.
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, use a gap filling filter in a Profile Filter tool to fill the gaps; for more information, see *Profile Filter* on page 564.
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
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 214.
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.
Selection 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. Only displayed when **Selection Type** is set to an "Index"

Parameter	Description
	option.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

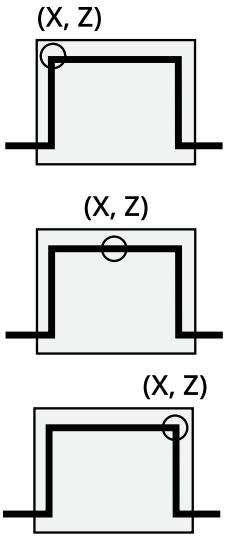
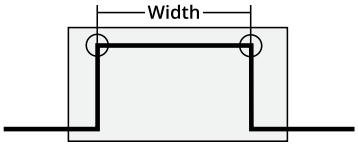
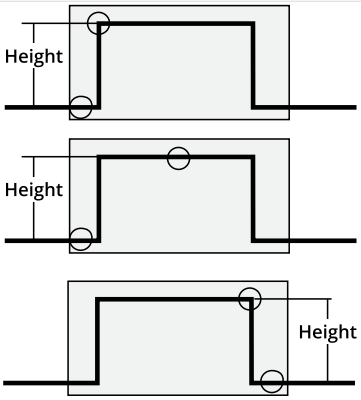


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	Illustration
Count The number of strips.	
X, Z Measures the X and Z position, respectively, of a strip.	
Width Measures the width of a strip.	
Height Measures the height of a strip.	

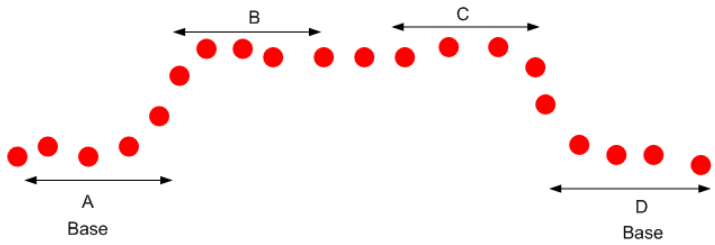
Features

Type	Description
Left Edges	Arrays of Point geometric features representing the left edges, right edges, and

Type	Description
Right Edges	centers of all strips, from left to right.
Centers	

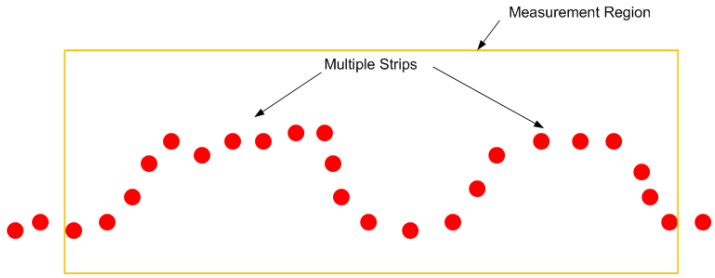
Strip Algorithm

A strip is a flat region bounded on the left and on the right by edges. The Strip tool can measure the edge positions, width, and height of a strip. The Strip tool assumes that regions outside the strip, referred to as the base regions (Region A and D below), deviate in height from the start and end parts of a strip (Region B and C).



When the target is sitting on the surface, the base is lower than the strip (as shown above). Alternatively for a groove the base is above the strip surface. The base could be missing when the target is hanging in the air or the surface holding the target falls outside the sensor's active area. You can control the base type in the measurement panel.

The Strip tool can detect multiple strips. You can select a region of interest, referred to as the measurement region, from which the algorithm search for multiple strips.

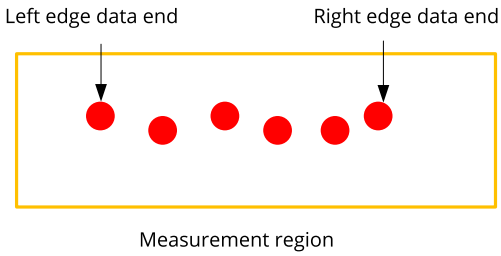
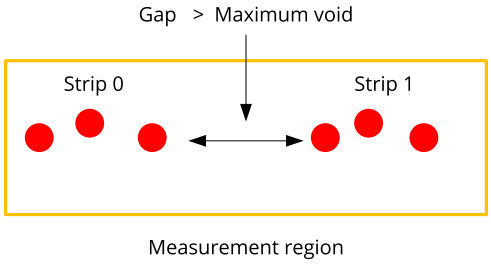
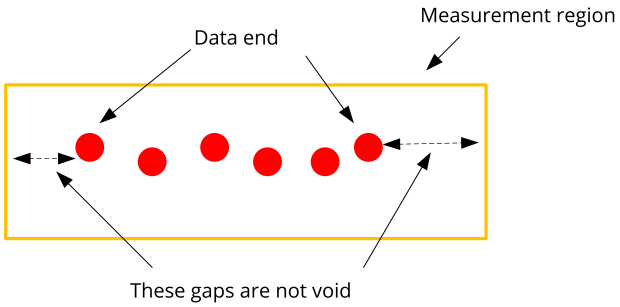


Strip Start and Terminate Conditions

The Strip tool allows you to define how a strip starts and ends. The Left Edge parameter controls how a strip starts and the Right Edge parameter controls how a strip ends.

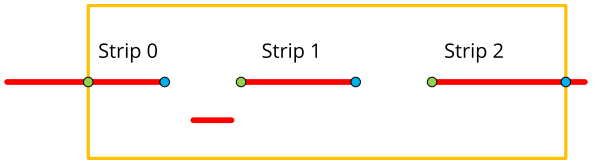
Start / terminate conditions

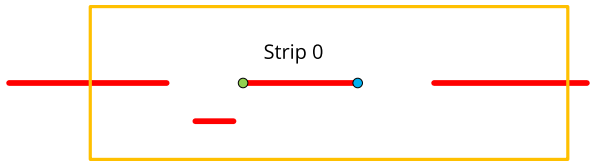
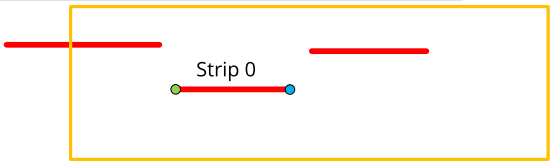
Condition	Description
Rising	Rising step edge detected based on the strip edge parameters. For details on how the step edge is detected, *Strip Step Edge Definitions* on page 645.
Falling	Falling step edge detected based on the strip edge parameters. For details on how

Condition	Description
	the step edge is detected, *Strip Step Edge Definitions* on the next page.
Data end	The first (for the left edge) or the last (for the right edge) valid profile data point in the measurement region. 
Void	Gaps in the data that are larger than the maximum void threshold.  Gaps at the ends of the measurement region's boundary are not considered as a void. 

The following examples show how the parameters affect the strip detection in different scenarios.

Left and Right Edge conditions

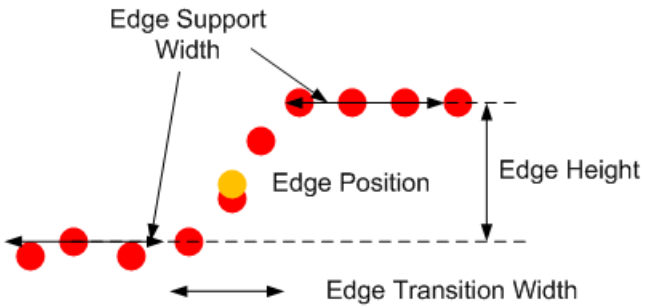
Condition	Example
Left: Rising, data end, void Right: Falling, data end, void	

Condition	Example
Left: Rising, void Right: Falling, void	
Left: Rising Right: Data end, void	
Left: Data end, void Right: Falling	
Left: Falling Right: Rising	

Strip Step Edge Definitions

The Strip tool detects step edges based on the parameters Base Type, Edge Transition Width, Edge Support Width, and Minimum Edge Height.

When Base Type is set to Flat, the regions around the edges are visible and the edge positions are between the base and the strip surface.

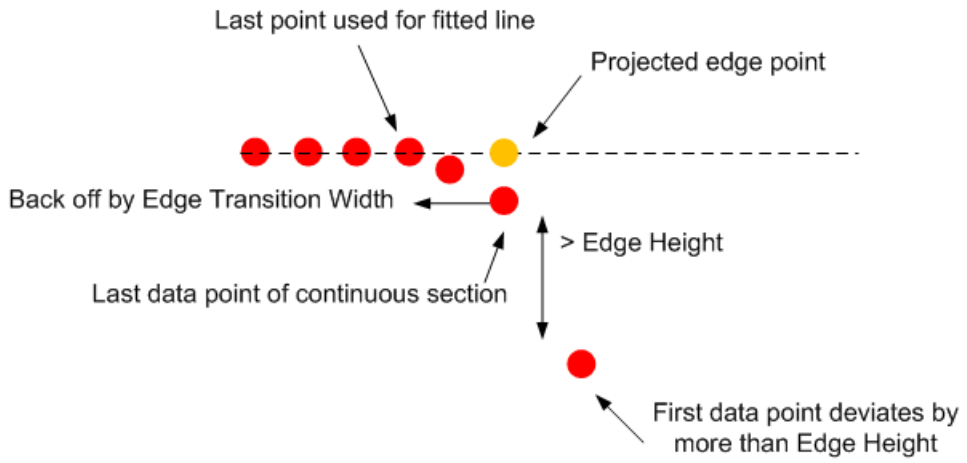


The Minimum Edge Height parameter defines the size of the step edge. The Edge Transition Width parameter specifies the nominal width of the transition, from the base to the strip surface.

The Edge Support Width parameter defines the width of the region around the edges from which the data is used to measure the step change. To improve noise immunity, the height level of the Edge Support Width parameter is calculated by averaging the data within the region.

When the base is set to None, the tool looks for continuous sections that are wider than the Edge Support Width parameter and have no data points that deviate positively or negatively more than

the value of the Minimum Edge Height parameter. The data in the strip support region (the raised area) must be smooth. The height level of the continuous region is calculated based on the fitted line as shown below.



The algorithm then backs off by the value of the Edge Transition Width parameter and uses the data up to the back-off point to create the fitted line and projects the edge point on the line. This step prevents the points near the end of a rounded strip from affecting the height of the strip.

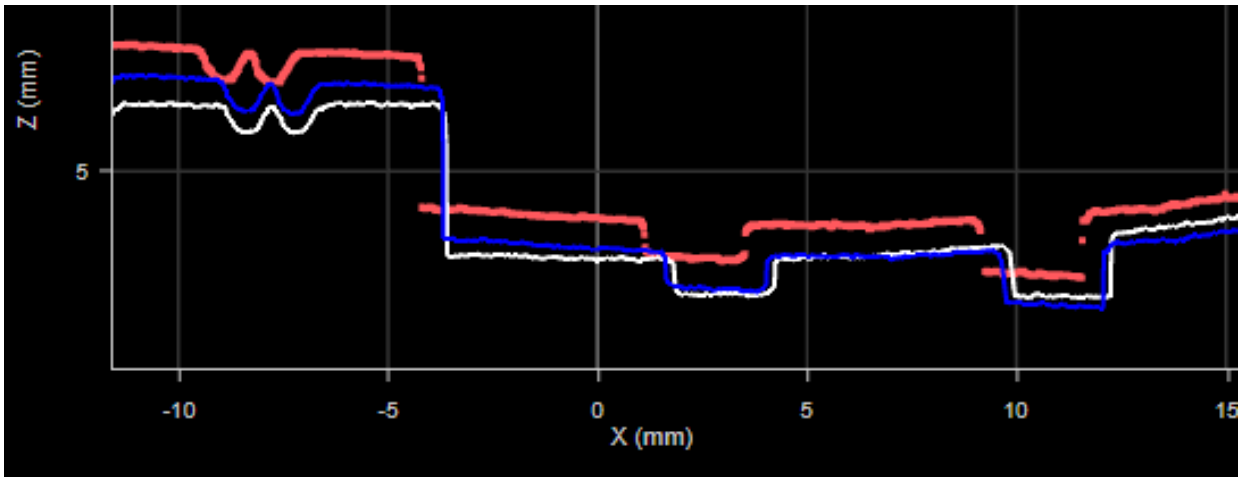
Profile Template Matching

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 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. Finally, the tool produces a "difference" profile on which you can similarly perform measurements.

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.

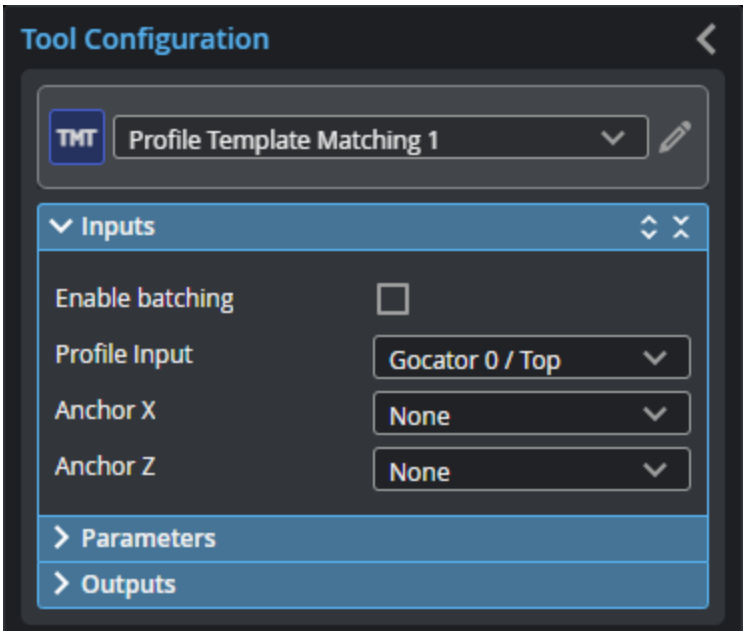
Note that in the image above, the tool is performing only a rough alignment to ensure that the different profiles are clearly visible. Typically, the aligned profile will be on top of the master profile.

The actual colors that are used for aligned and difference profiles *outputs*, as opposed to the profiles toggled with the **Display Master** and **Display Aligned Profile** parameters, will depend on how each profile's color is configured in the Displayed Outputs panel.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

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:

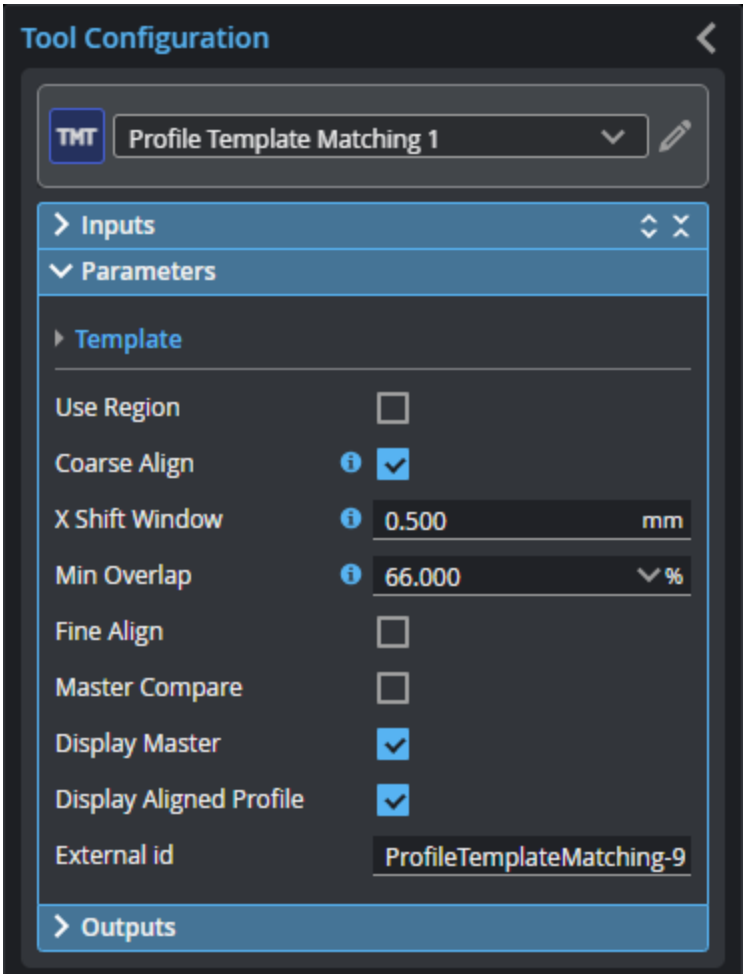
- Max Height Difference
- Max Difference Position X

- Max Difference Position Z
- Standard Deviation
- Difference Average
- Difference Sum
- Variance
- Matching Score

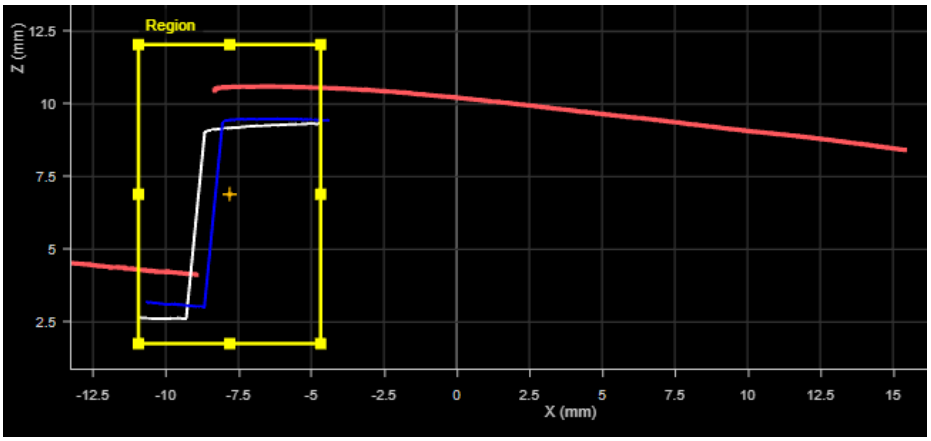
Also, for these “master compare” measurements, if the profile has been aligned to the master (either **Coarse Align** or **Fine Align** is enabled), the measurements compare the *aligned* profile and the master. If the profile has not been aligned (both alignment parameters are disabled), the measurement compares the *original* (unaligned) profile and the master.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



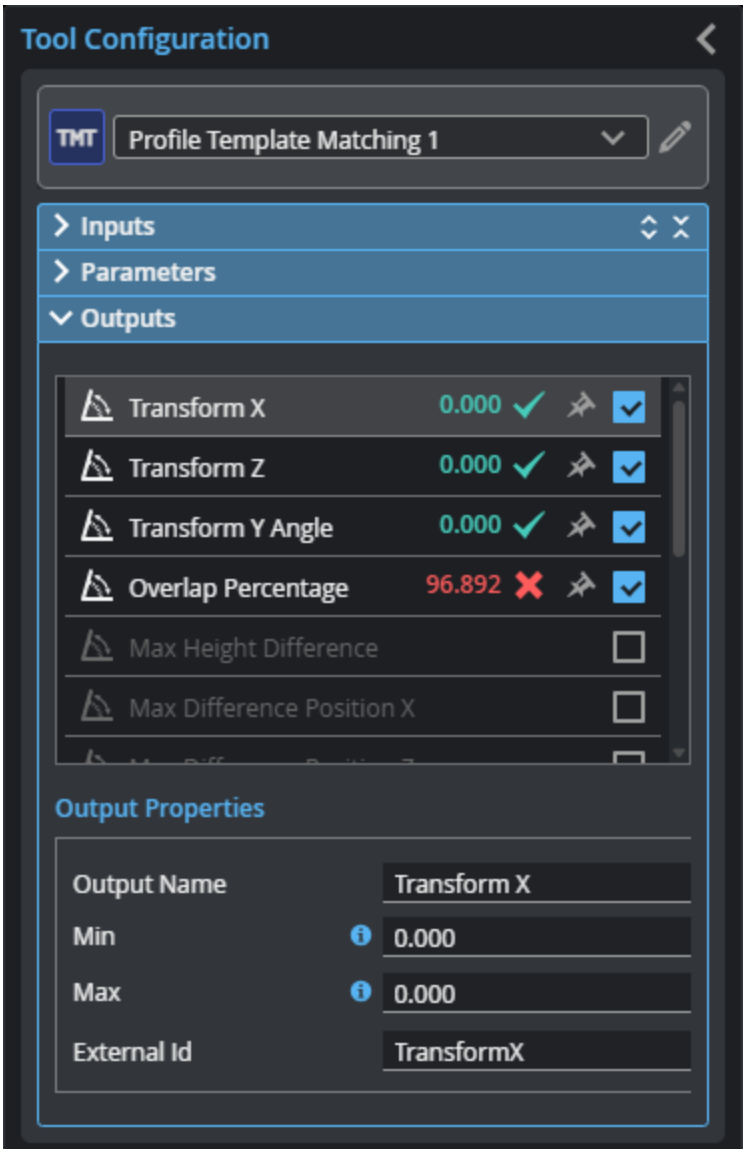
Parameters

Parameter	Description
Template	When expanded, displays **File** and **Operation** parameters.
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: • **Normal**: Selected by the tool after you perform another file operation. • **Create**: Creates a new template file. • **Load**: Loads the template selected in **File**. • **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. • **Delete**:Deletes the template file selected in **File**. • **Refresh**: Refreshes the list of files.
Use Region	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. For more information on regions, see *Regions* on page 214.
Region	The size and position of the region in which the matching (alignment) is performed. This can be useful if only a small part of the profiles is suitable for alignment. 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. 
Coarse Align	When enabled, shows the **X Shift Window** parameter. Use this setting by itself if

Parameter	Description
	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.
Display Master	Displays the Master template (white profile).
Display Aligned Profile	Displays the aligned (blue profile).
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement
Transform X
Transform Z

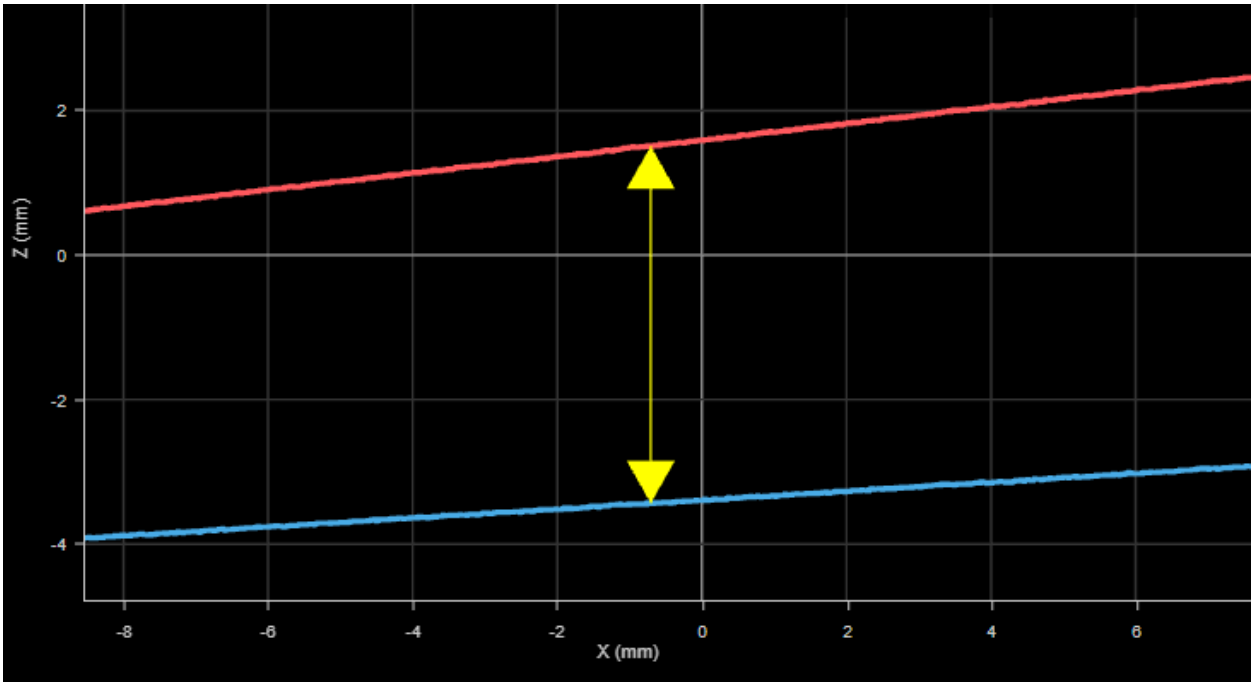
Measurement
The distance the profile has shifted on the X and Z axis after alignment to the master, respectively.
Transform Y Angle
The rotation of the profile around the Y axis after alignment.
Max Height Difference
The maximum height difference between the profile and the master.
Max Difference Position 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.
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.

Profile Thickness

The Thickness tool lets you perform thickness measurements between the two profiles you specify.

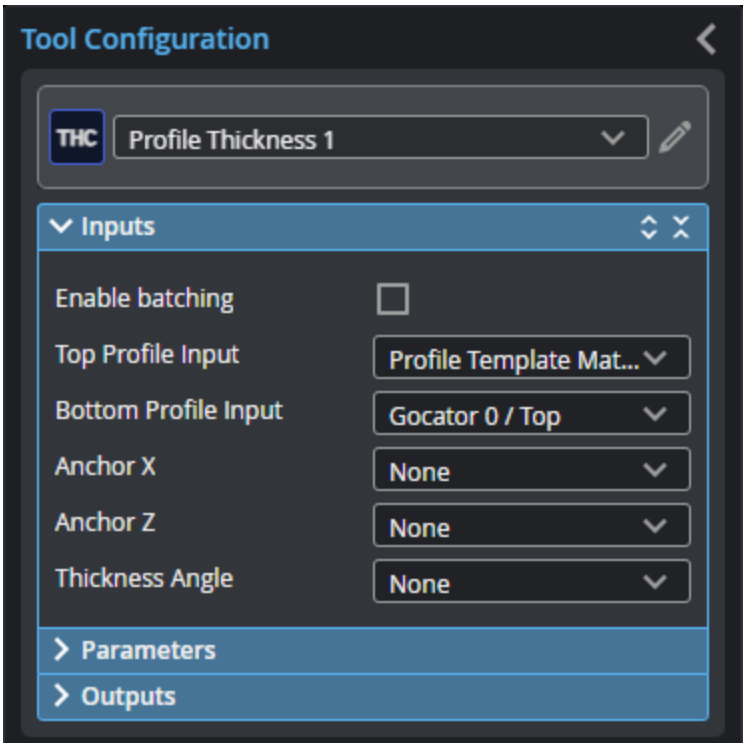


The tool also lets you specify an angle measurement relative to which the tool performs the thickness measurements, up to a maximum of 20 degrees, to compensate for minor tilt of the target. To get the compensation angle, you could first fit a straight line to the top profile using the Profile Line tool, and pass that tool's Angle measurement to Profile Thickness. (For information on Profile Line, see *Profile Line* on page 583.)

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

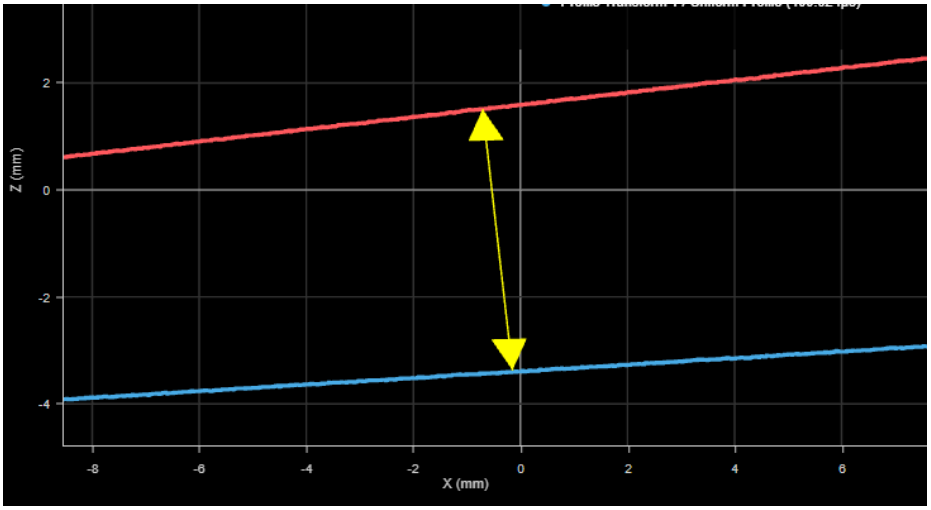


To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

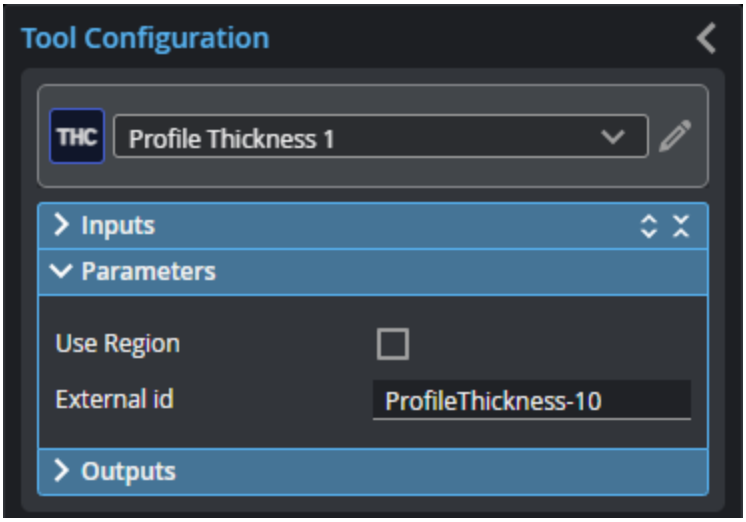
Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with a maximum of two elements. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with a maximum of two elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Top Profile Input	The top and bottom profiles the tool uses to perform thickness measurements. If

Name	Description
Bottom Profile Input	**Enable Batching** is checked, each input takes an array.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Thickness Angle	An angle measurement from another tool that the Thickness tool uses for the angle of the thickness measurement. If no angle is provided, the thickness measurement is perpendicular to the X axis.



Parameters

You configure the tool's parameters in the expandable **Parameters** section.



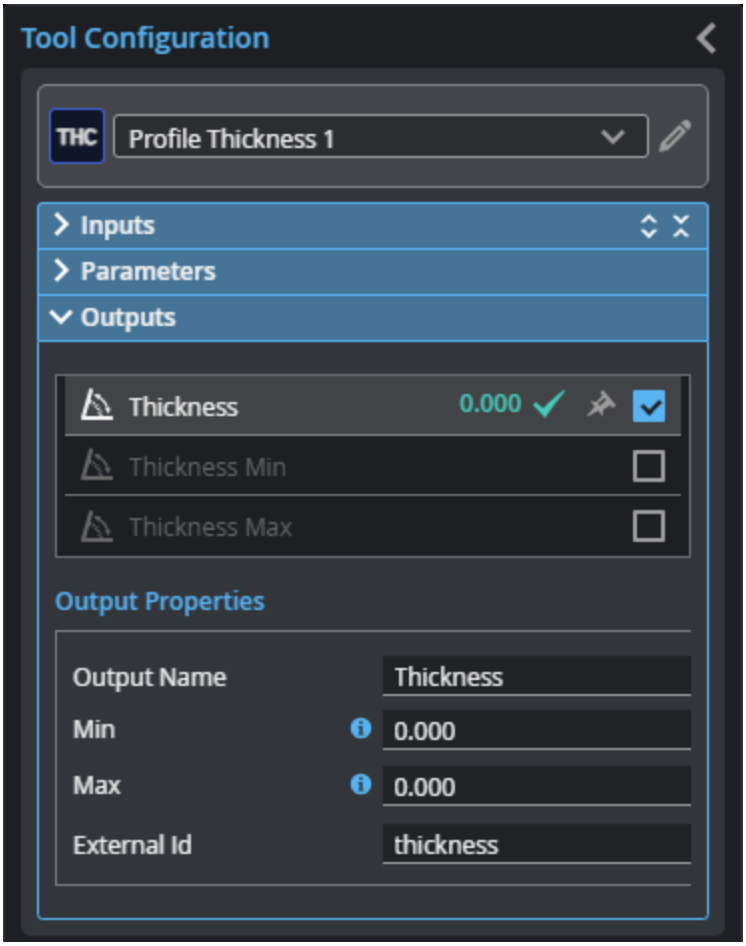
Parameters

Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool

Parameter	Description
	uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

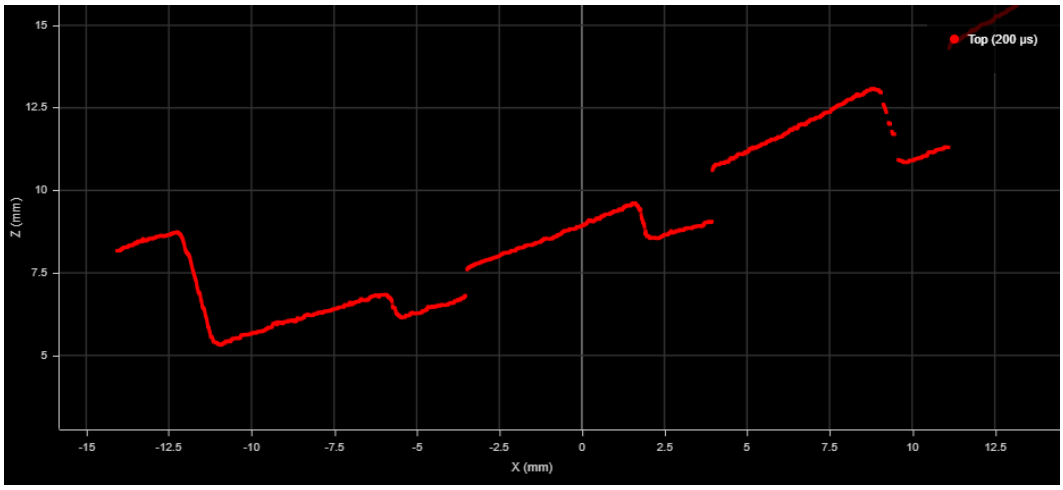
Measurements

Measurement
Thickness The thickness measured in the scan data.
Thickness Min The minimum thickness measured in the scan data.
Thickness Max The maximum thickness measured in the scan data.

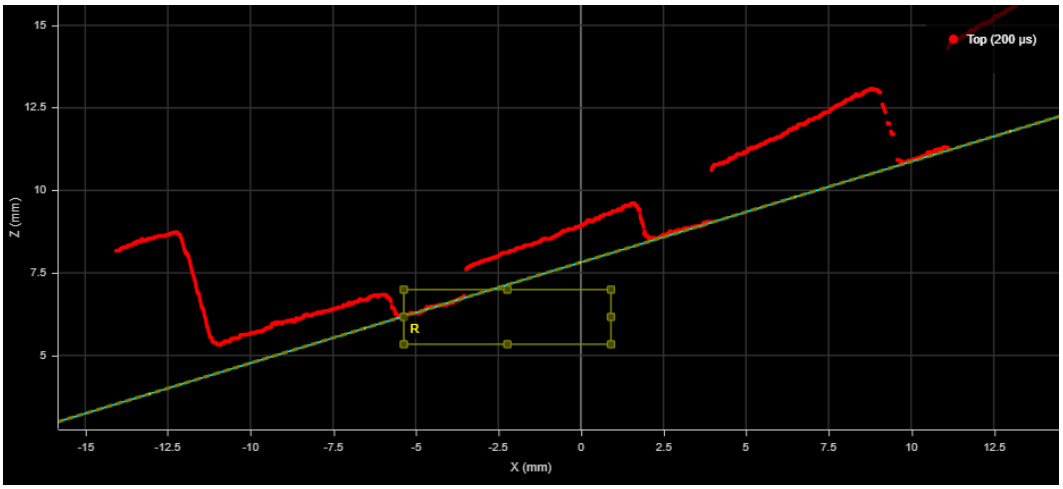
Profile Transform

The Profile Transform creates a new profile based on the coordinate system of geometric features the tool uses as input and/or fixed input values. The tool accepts a Line geometric feature (rotating the profile so that the line is parallel to the X axis) and/or a Point geometric feature (using it as the X and Z origin).

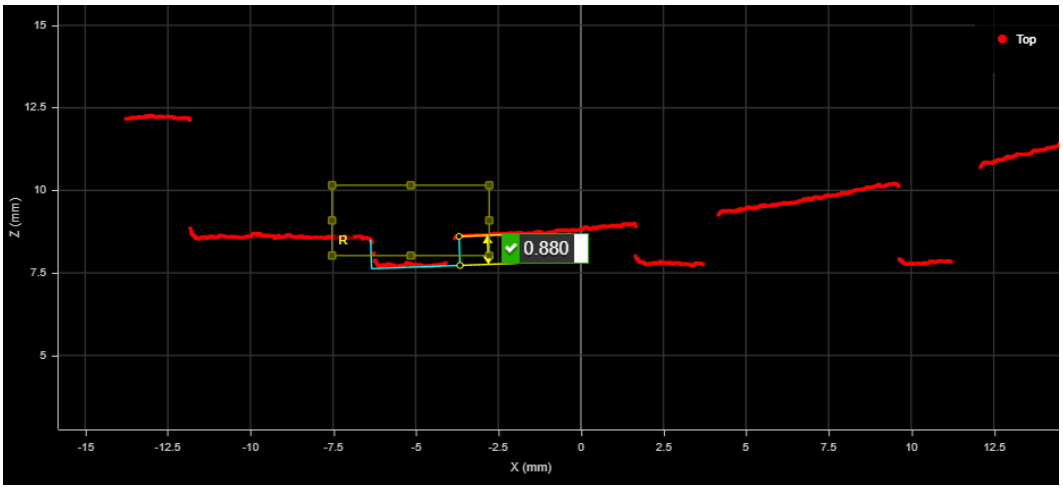
For example, in the following, if you want to measure the characteristics of the first groove on the left, you can use the tool to rotate the profile.



First, a Profile Line tool is added and configured on the bottom of the groove. The tool is configured to output a line geometric feature.



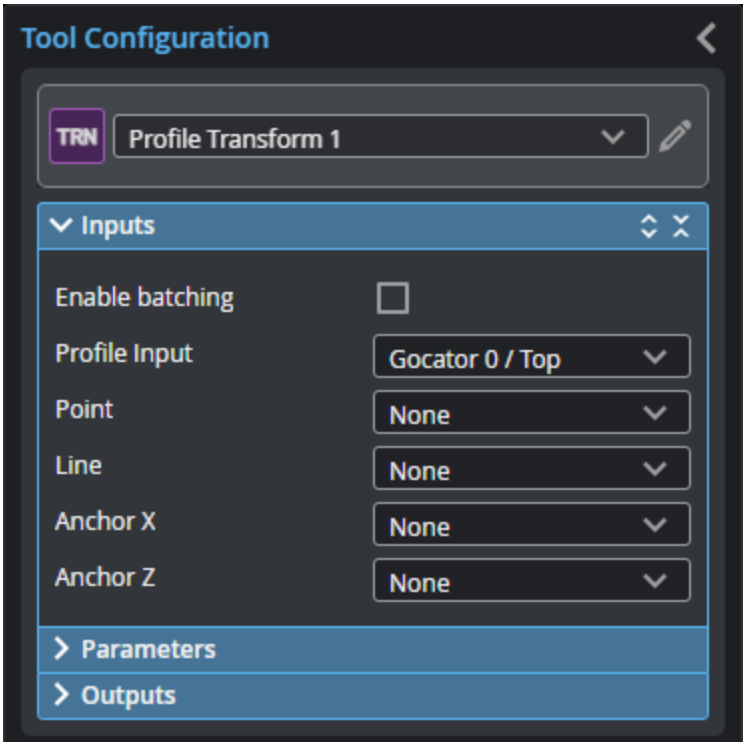
A Profile Transform tool takes the line geometric feature as input, and the transformed profile from that tool is used as input for a Profile Groove tool, which measures the groove's characteristics:



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



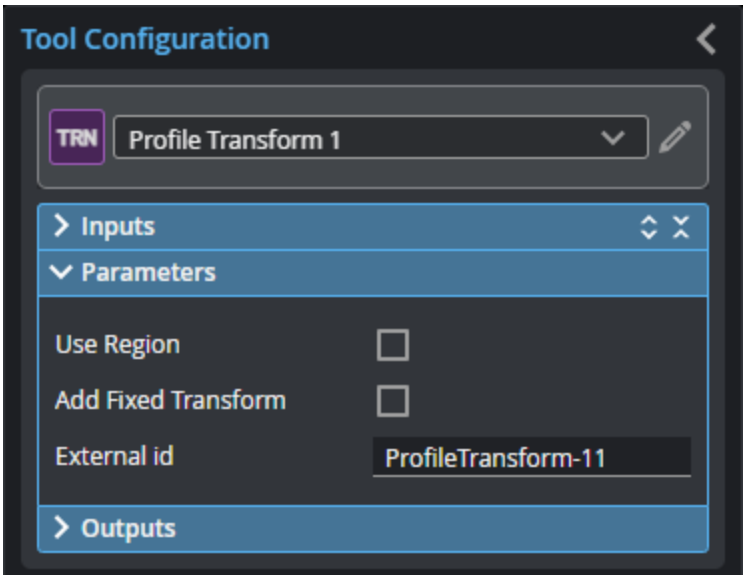
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes.
Point	The Point geometric feature the tool uses to offset a profile to an X and Z origin of 0.
Line	The Line geometric feature the tool uses to position (X and Z) and to rotate a profile Y rotation).
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Use Region	Whether the tool should limit the transformed profile that it outputs to a user-defined region. When enabled, the tool displays a **Region** parameter you use to define the region. If this option is not enabled, the tool transforms the entire profile.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.

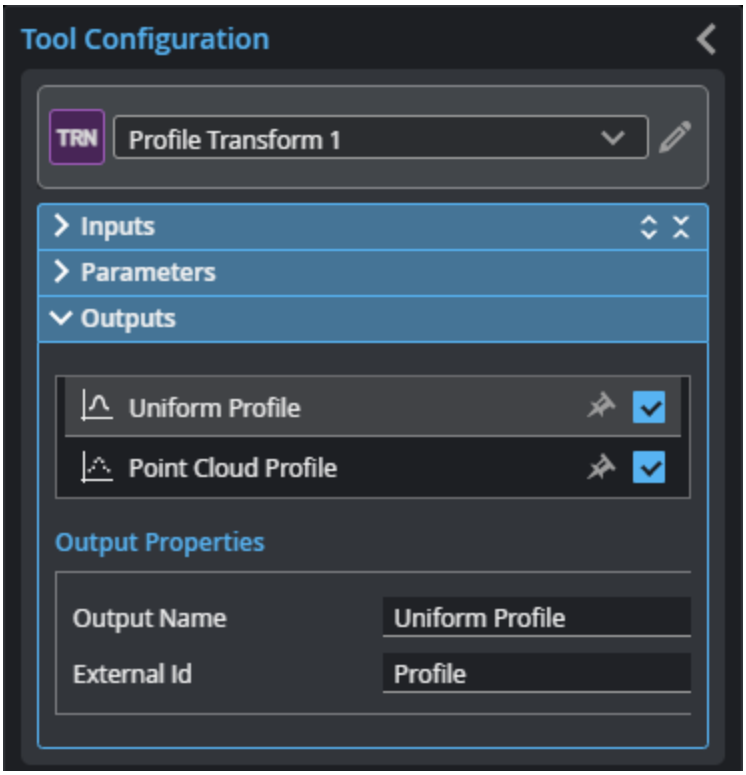
Parameter	Description
Add Fixed Transform	Enables **X Offset**, **Y Offset**, and **Angle** parameters you can use to manually set a transformation. Useful if you know the profiles in the scan data will always be in a certain position or orientation.
Resampling parameters	Expand to set the tool's resampling mode. For more information, see *Sampling Parameters* below.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Sampling Parameters

Parameter	Description
Sampling Mode	The mode for the sampling of the transformed profile. One of the following: **Auto Adjust** The tool selects the best sampling parameters to fit the transformed profile. It adjusts the XZ offsets and XZ spacing automatically. Note that the parameters may change from one frame to the other, based on the input profile. **Keep Original** The parameters of the output profile will be the same as the input profile. This is useful for small transformations, where the output profile can easily fit in the same region as the input profile. **Custom** Lets you manually adjust the output profile's region and spacing. **X Offset** and **Z Offset** These default to 0.000. Click the tooltip icon to see the XZ offsets of the input profile. **X Spacing** This defaults to 0.000. Click the tooltip icon to see shows the X spacing of the input profile. Leave the value at 0.000 to keep the original spacing of the input profile. **Z Range** This defaults to 0.000. Click the tooltip icon to see the Z range of the input profile. Leave the value at 0.000 to keep the original range of the input profile.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Data

Type	Description
Profile Uniform	The transformed profile.
Profile Point Cloud	The outputs will contain the same profile data, but the type (uniform vs. point cloud) will differ, so any Profile tool will be able to take this tool's output as input.

Raw Profile Matching

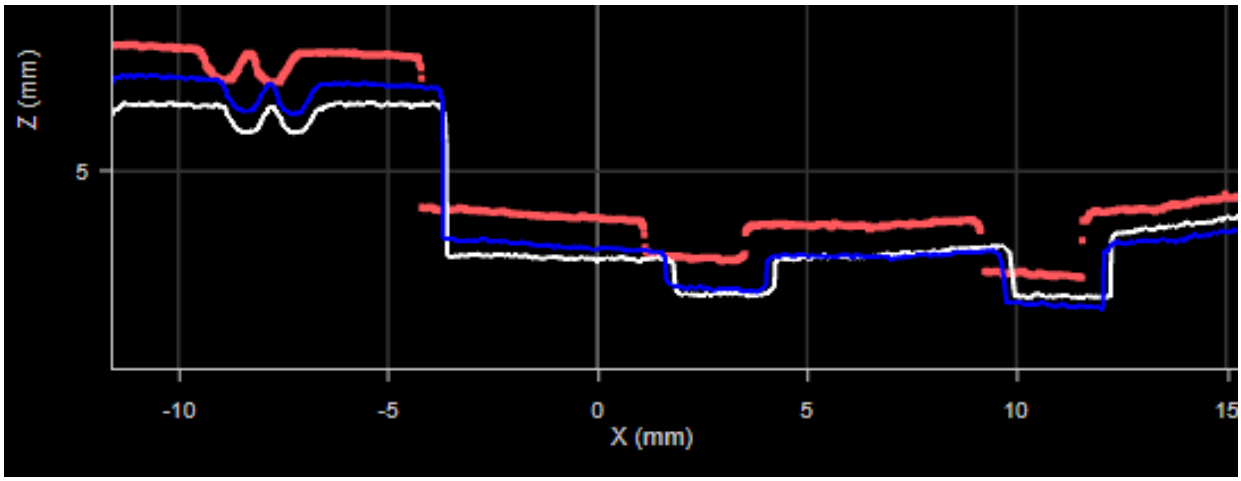
> The Raw Profile Matching tool takes unresampled profile data as input (the **Enable uniform spacing** setting is disabled on Acquire > Scanners).

The Raw Profile 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 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).

In the data viewer, the profiles are rendered using different colors:



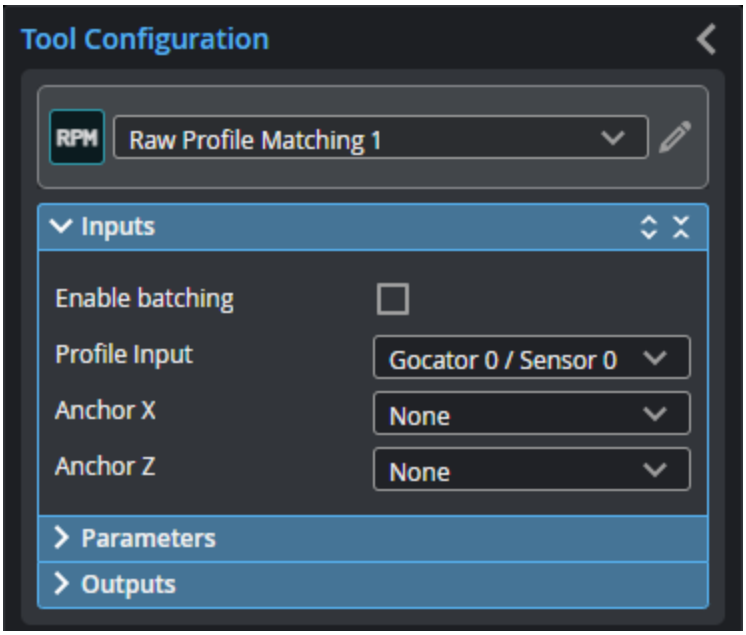
The master profile is rendered in white. The aligned profile is rendered in blue. The current profile is rendered in red.

Note that in the image above, the tool is performing only a rough alignment to ensure that the different profiles are clearly visible for illustration purposes. Typically, the blue aligned profile will be on top of the white master profile.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



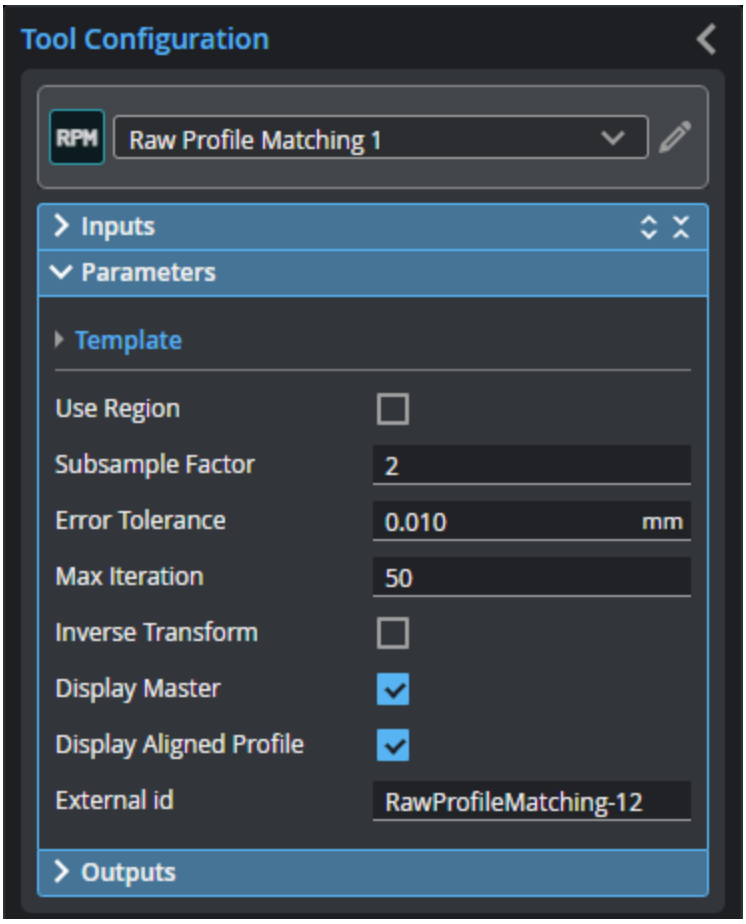
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Profile Input	The data the tool applies measurements to or processes.
Anchor X or Anchor Z	The X or Z measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.

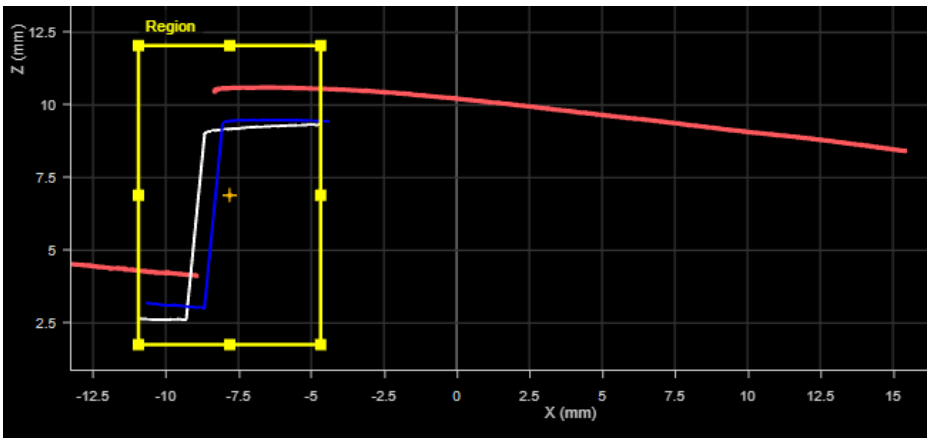
Parameters

You configure the tool's parameters in the expandable **Parameters** section.



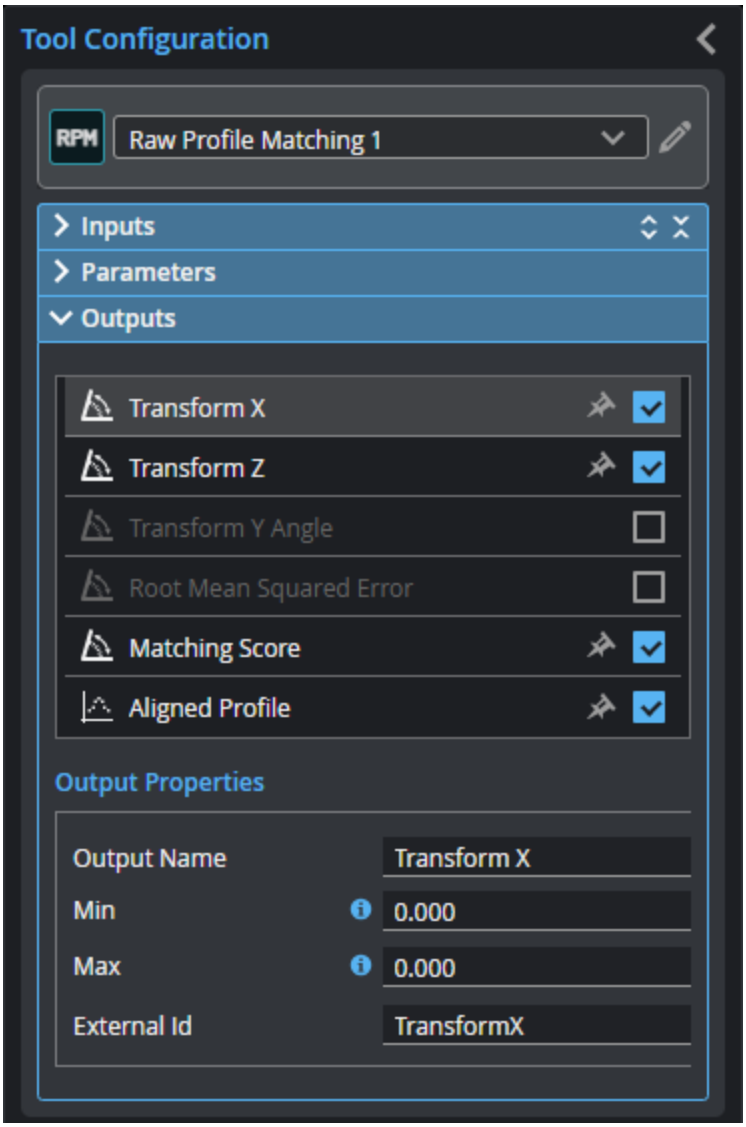
Parameters

Parameter	Description
Template	When expanded, displays **File** and **Operation** parameters.
File	A list of templates available to the tool to use as a master profile for alignment and comparisons. Use the **Operation** parameter to add and remove templates.
Operation	Provides operations related to profile template files (masters). One of the following: • **Normal**: Selected by the tool after you perform another file operation. • **Create**: Creates a new template file. • **Load**: Loads the template selected in **File**. • **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. • **Delete**:Deletes the template file selected in **File**. • **Refresh**: Refreshes the list of files.
Use Region	Indicates whether the tool uses a user-defined region to perform *matching*. (The

Parameter	Description
	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. For more information on regions, see *Regions* on page 214.
Match Region	When **Use Region** is enabled, provides settings for the 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.)
Subsample Factor	The sampling interval of the input profile point cloud.
Error Tolerance	The difference tolerance for the master comparison.
Max Iteration	The maximum number of iterations the tool uses to perform iterative closest point (ICP). The tool stops iterating once the input profile is considered within the error tolerance (see above).
Inverse Transform	When enabled, inverts the output transformation.
Display Master	Displays the Master template (white profile).
Display Aligned Profile	Displays the aligned profile (blue profile).
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement
Transform X
Transform Z

Measurement
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.
Root Mean Squared Error
The root mean squared error of the alignment.
Matching Score
Returns a value between 0 and 1 representing 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.

Script Tool

The Script tool executes a user-defined Python-based script.

For a list of the tool's supported classes and methods, see *Functions and Classes* on page 675.

For examples, see *Examples* on page 688.

The tool supports all GoPxL data types as input and output:

- Surface, Profile, Image, and Mesh data; intensity data is available with Surface and Profile data
- Measurements (values and decisions, and anchor sources for other tools)
- Strings
- Geometric features
- Arrays of the supported types

> The Script tool does *not* support writing files when it is running on a sensor or GoMax device, and will display a permissions error if you attempt this. Writing to files is only supported when the tool is running on a PC instance of GoPxL. When writing to files, you should explicitly set the path to a writable location.

The tool does not support using `print()`, `sys.stdout`, or Python's `logging` module functions to send messages to GoPxL's system log. Instead, use one of the "log_" functions. For more information, see *Logging* on page 681.

Script tools are added, configured, and removed like other measurement tools.

> The Script tool uses Python version 3.12.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Included Python Modules

In addition to the standard Python library modules, the tool also includes the following modules:

- NumPy 1.26.4: NumPy is useful for operating efficiently on arrays of surfaces or profiles, or images.
- OpenCV 4.10.0: A powerful computer vision library that includes a wide variety of algorithms.

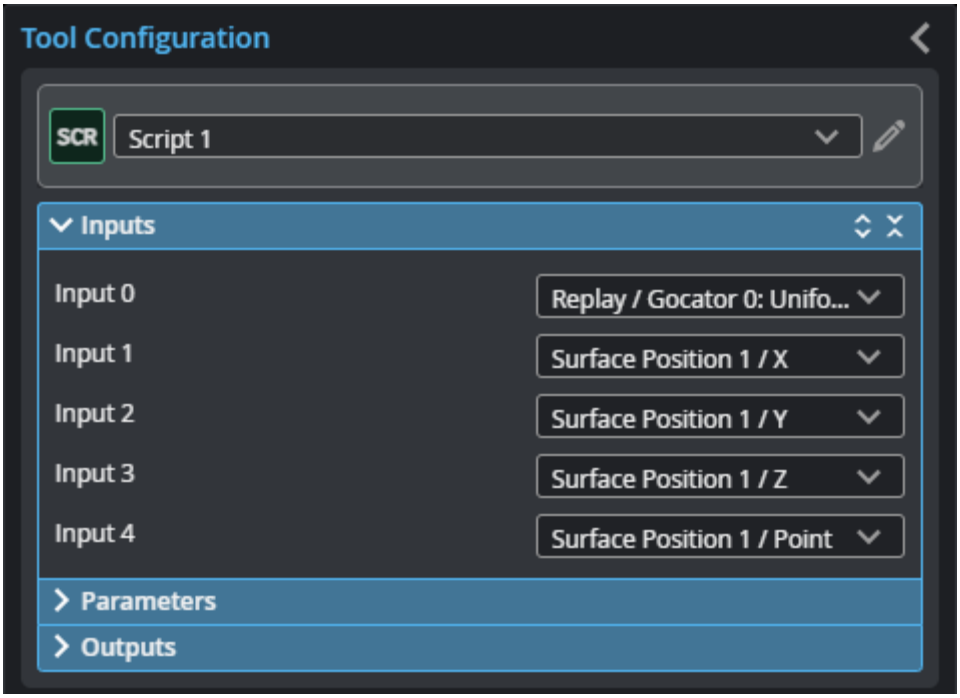
On Windows, you can install additional modules with `pip` by running the following command under the `bin\win64\python` folder in the main GoPxL folder:

```
python3.exe -m pip install <modulename>
```

Note that for some libraries you may need to specify a specific version to ensure compatibility with the version of Python included with GoPxL.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

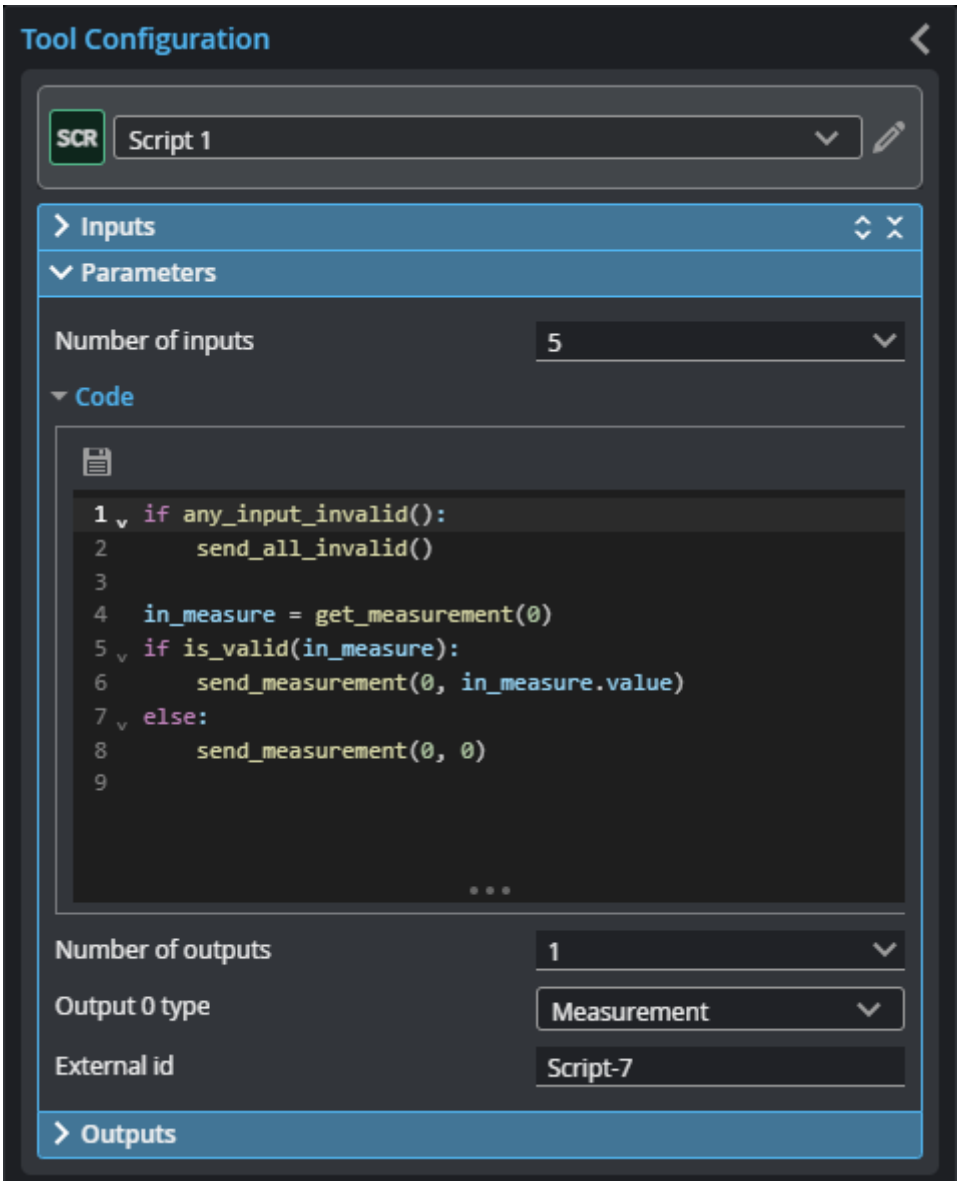


Inputs

Name	Description
Input {n}	Selects the input for the script where {n} is the index of the input. All input types are allowed. Input cannot be 'None'. In the `get_` functions' `index` parameters, use this index. The **Number of inputs** parameter (see below) determines the number of inputs. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.

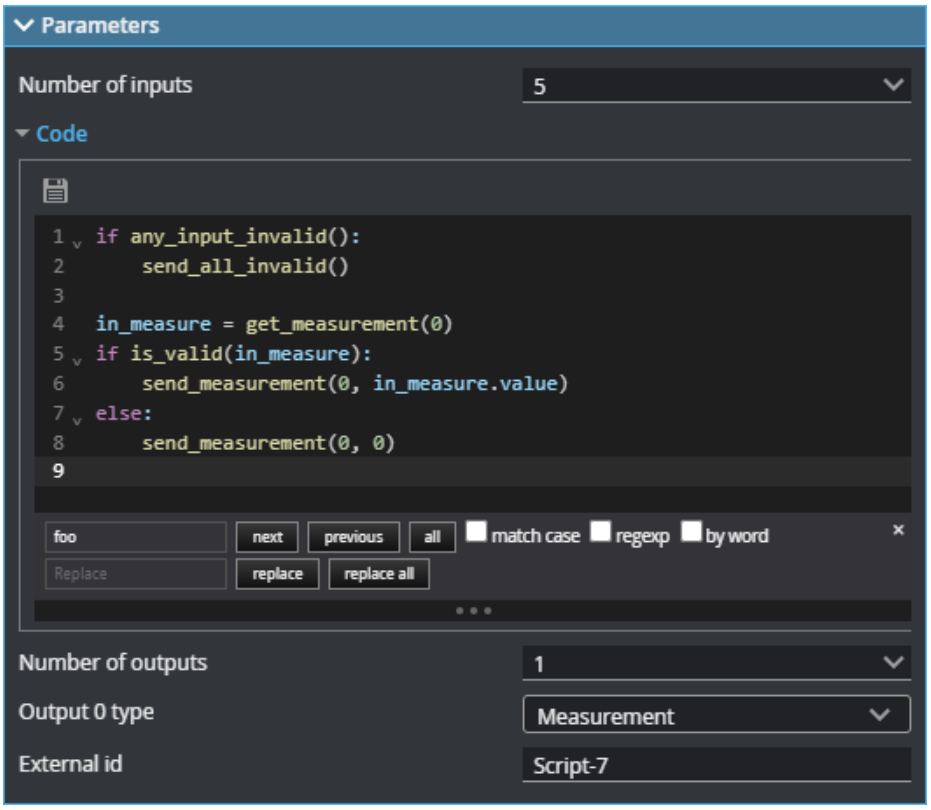
Parameters

You configure the tool's parameters in the expandable **Parameters** section.



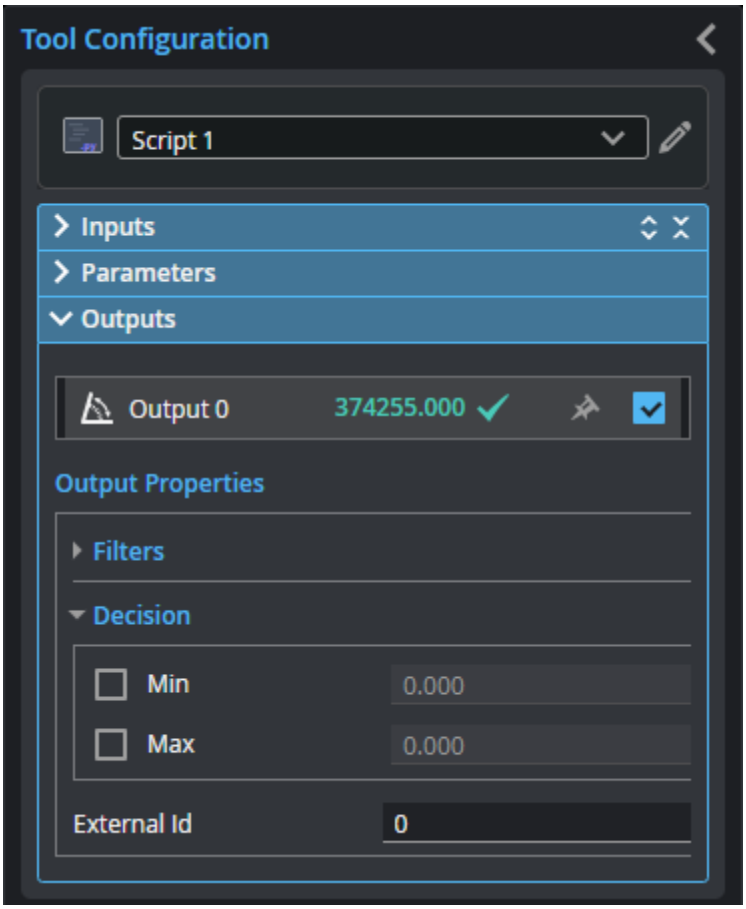
Parameters

Name	Description
Number of Inputs	Specifies the number of inputs needed in the script.
Code	The Python code the tool runs. Supports expanding and collapsing of code blocks. You can resize the editor vertically with the grabber to the lower left of the editor. You can resize the width of the editor by resizing the **Tool Configuration** panel. The following keyboard shortcuts are available in the editor. **Ctrl-S**: Saves the script. The tool checks the syntax. Use Cmd-S on macOS. **Ctrl-F**: Opens a search and replace panel. The search and replace panel supports regular expressions. Use Cmd-F on macOS. To close the panel, click the "x" to the upper right.

Name	Description
	 **Ctrl-D**: Highlights all occurrences of the word under the cursor. Each time you press Ctrl-D after that, the cursor moved to the next occurrence. Use Cmd-D on macOS. **Alt-ArrowLeft / Alt-ArrowRight**: Move the cursor over the next element to the left or right, respectively. **Alt-ArrowUp / Alt-ArrowDown**: Move the selected lines up or down one line, respectively. Combining this with Shift copies lines instead. **Ctrl-/**: Toggles commenting. Use Cmd-/ on macOS. **Ctrl-[/ Ctrl-]**: Decreases or increases the indentation of the selected line. Use Cmd-[or Cmd-] on macOS.
Number of outputs	Specifies the number of outputs the script produces.
Output {n} type	Selects the output type for each output, where {n} is the index of the output.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement
Output {n} Output from the Python script, where {n} is the index of the output. The number of outputs and the type of each output is determined by the **Number of outputs** and **Output type {n}** parameters, respectively.

Functions and Classes

Getting Input

The following Python functions are available for performing input operations and testing input validity.

The index of the input is given as an argument. If the input at the index is missing or it is the wrong type, None is returned. Functions will return an array if input is an array.

For information on the returned types, see *Data Structures* on page 681.

Input Functions

Function	Description
is_valid(obj)	Checks whether the object is valid or not. *Parameters* **obj**: Input message. *Returns* True if valid.
get_surface(index)	Gets a surface message or array of messages from the specified input. *Parameters* **index**: Input index. *Returns* SurfaceMsg, array of SurfaceMsg, or None.
get_profile(index)	Gets a profile message or array of messages from the specified input. *Parameters* **index**: Input index. *Returns* ProfileMsg, array of ProfileMsg, or None.
get_measurement(index)	Gets a measurement message or array of messages from the specified input. *Parameters* **index**: Input index. *Returns* MeasureMsg, array of MeasureMsg, or None.
get_point(index)	Gets a point message or array of messages from the specified input. *Parameters* **index**: Input index. *Returns* PointFeatureMsg, array of PointFeatureMsg, or None.
get_circle(index)	Gets a circle message or array of messages from the specified input. *Parameters*

Function	Description
	index: Input index. *Returns* CircleFeatureMsg, array of CircleFeatureMsg, or None.
get_plane(index)	Gets a plane message or array of messages from the specified input. *Parameters* **index**: Input index. *Returns* PlaneFeatureMsg, array of PlaneFeatureMsg, or None.
get_line(index)	Gets a line message or array of messages from the specified input. *Parameters* **index**: Input index. *Returns* LineFeatureMsg, array of LineFeatureMsg, or None.
get_image(index)	Gets an image message or array of messages from the specified input. *Parameters* **index**: Input index. *Returns* ImageMsg, array of ImageMsg, or None.
get_string(index)	Gets a string or array of strings from the specified input. *Parameters* **index**: Input index. *Returns* StrMsg, array of StrMsg, or None.
get_mesh(index)	Gets a mesh message or array of messages from the specified input. *Parameters* **index**: Input index. *Returns* MeshMsg, array of MeshMsgs or None.

Drawing Graphics

The Python functions listed below are available for drawing graphics in the data viewer. The default values used in draw() functions are as follows:

GRAPHIC_DEFAULT_SIZE = 5

GRAPHIC_DEFAULT_COLOR = Colors.CYAN

Draw Functions

Function	Description
draw_points(points, color = GRAPHIC_DEFAULT_COLOR, size = GRAPHIC_DEFAULT_SIZE, shape = MarkerShape.CIRCLE, output = None)	Draw points with customizable size, shape and color. *Parameters* **points**: A list of points. Each item can be a Point, Point3d, or a tuple containing 2 or 3 numeric values (e.g., (x, y) or (x, y, z)). **color**: RGB color to apply to the points. **size**: Size of the points to be drawn. **shape**: Shape of the points to be drawn. **output**: Output index where the graphics are associated. Use None for tool graphics.
draw_lines(points, color = GRAPHIC_DEFAULT_COLOR, width = GRAPHIC_DEFAULT_SIZE, hasStartPointArrow = False, hasEndPointArrow = False, output = None)	Draw graphic lines with customizable width and color. *Parameters* **points**: A list of points. Each item can be a Point, Point3d, or a tuple containing 2 or 3 numeric values (e.g., (x, y) or (x, y, z)). **color**: RGB color to apply to the points. **size**: Width of the line to be drawn. **hasStartPointArrow**: Whether the start point arrow should be drawn. **hasEndPointArrow**: Whether the end point arrow should be drawn. **output**: Output index where the graphics are associated. Use None for tool graphics.
draw_region(points, center, dimensions, angle, output = None)	Draw region with customizable dimensions. *Parameters* **center**: Region center. **dimensions**: Width, height and depth of the region. **angle**: Rotation angle (about Z Axis) of the region. **output**: Output index where the graphics are associated. Use None for tool graphics.
draw_label(text, position, output = None)	Draw label with customizable location. *Parameters* **text**: The label string. **position**: The position of the label. **output**: Output index where the graphics are associated. Use None for tool graphics.
draw_plane(normal, distance, output = None)	Draw plane with customizable normal and distance. *Parameters* **normal**: The normal of the plane. **distance**: The to the origin of the plane. **output**: Output index where the graphics are associated. Use None for tool graphics.

Function	Description
draw_ray(position, direction, width = GRAPHIC_DEFAULT_SIZE, color = GRAPHIC_ DEFAULT_COLOR,, output = None)	Draw points with customizable size, shape and color. *Parameters* **position**: The position of the ray. **direction**: The direction of the ray. **size**: Width of the line to be drawn. **color**: RGB color to apply to the points. **output**: Output index where the graphics are associated. Use None for tool graphics.

Sending Output

The following Python functions are available for performing output operations.

The index of the output is given as an argument.

For information on the output types, see *Data Structures* on page 681.

Output Functions

Function	Description
is_output_enabled(index)	Checks whether the output at the specified index is enabled. *Parameters* **index**: Output index. *Returns* True if the output is enabled.
send_surface(index, offset, scale, points, intensity=None, is_adjacent=None, header=None, intensity_ format=None)	Sends a surface message or array of messages to the specified output. *Parameters* **index**: Output index. **offset**: Offset of the data or array of offsets. **scale**: Scale of the data or array of scales. **points**: Surface points data or array points data. Points for uniform data are in an array of size (height, width). Points for point cloud (non-uniform) data are in an array of size (height, width, 3). **intensity**: Surface intensity data of size (height, width), or array of intensity data. **is_adjacent**: Set to true if the surface data is adjacent. Can also be an array of adjacent info. **header**: Header part of the data message. Uses header from the first input if not set. **intensity_format**: Pixel format of the intensity data (PixelFormat type).
send_profile(index, offset, scale, points,	Sends a profile message or array of messages to the specified output. *Parameters*

Function	Description
intensity=None, slices=None, header=None)	**index**: Output index. **offset**: Offset of the data or array of offsets. **scale**: Scale of the data or array of scales. **points**: Profile point data or array of profile points. Data is in array of size (width) or (width,2). **intensity**: Profile intensity data of size (width), or array of profile intensities. **slices**: Not used. **header**: Header part of the data message. Uses header from the first input if not set.
send_measurement(index, value, label_ position=None, header=None)	Sends a measurement message or array of messages to the specified output. Note that the decision can't be set with this function because the decision of a measurement is only set after output using the Min and Max thresholds on the outputs. *Parameters* **index**: Output index. **value**: Value of the data message, or array of values. Numpy.nan if measurement is invalid. **label_position**: Position of point in measurement message, or array of positions. **header**: Header part of the data message. Uses header from the first input if not set.
send_point(index, position, header=None)	Sends a point message or array of messages to the specified output. *Parameters* **index**: Output index. **position**: Position of feature point in data message, or array of positions. **header**: Header part of the data message. Uses header from the first input if not set.
send_line(index, position, direction, header=None)	Sends a line message or array of messages to the specified output. *Parameters* **index**: Output index. **position**: Point vector in the line, or array of point vectors. **direction**: Direction vector of the line, or array of direction vectors. **header**: Header part of the data message. Uses header from the first input if not set.
send_circle(index, position, normal, radius, header=None)	Sends a circle message or array of messages to the specified output. *Parameters* **index**: Output index. **position**: Center position of the circle, or array of center positions. **normal**: Normal vector of the circular plane, or array of normal vectors. **radius**: Radius of the circle, or array of radii.

Function	Description
	header: Header part of the data message. Uses header from the first input if not set.
send_plane(index, normal, origin_distance, header=None)	Sends a plane message or array of messages to the specified output. *Parameters* **index**: Output index. **normal**: Normal vector of the plane, or array of normal vectors. **origin_distance**: Shortest distance from origin to plane, or array of distances. **header**: Header part of the data message. Uses header from the first input if not set.
send_image(index, pixels, pixel_format= PixelFormat.MONO_8, flipped_x=False, flipped_ y=False, transposed=False, header=None)	Sends an image message or array of messages to the specified output. *Parameters* **index**: Output index. **pixels**: Pixel data of image of size (height, width, pixel size), or array of image pixels. **pixel_format**: Pixel format of the image, or array of pixel formats. For more information, see *PixelFormat* on page 684. **flipped_x**: Set to `true` if the image data is flipped on the X axis, or array of flags. **flipped_y**: Set to `true` if the image data is flipped on the Y axis, or array of flags. **transposed**: Set to `true` if the image data is transposed, or array of flags. **header**: Header part of the data message. Uses header from the first input if not set.
send_string(index, string, position = None, decision = None, header=None)	Sends a string or array of strings to the specified output. *Parameters* **index**: Output index. **string**: The text to send. **position**: Position of point in measurement message. Type: Point3d(). **decision**: decision of measurement message. Type: MeasurementDecision() enum class. **header**: Header part of the data message. Uses header from the first input if not set.
send_mesh(index, vertex_ data, facet_data, vertex_ normal_data = None, facet_normal_data = None, vertex_texture_data = None, data_offset = None,	Sends a mesh message or array of messages to the specified output. *Parameters* **index**: Output index. **vertex_data**: Vertex data points. Type: numpy array of floats (x,y,z). **facet_data**: Facet data points. Type: numpy array of uint32 triangle indexes (vertex 1, vertex 2,vertex 3). **vertex_normal_data**: Vertex normal data points. Type: numpy array of

Function	Description
data_range = None, header=None)	floats (x,y,z). **facet_normal_data**: Facet normal data points. Type: numpy array of floats (x,y,z). **vertex_texture_data**: Vertex texture data. Type: numpy array of uint8. **data_offset**: Offset of the data or array of offsets. Type: Type: (x,y,z) | Point3d() **data_range**: Range of the data or array of ranges. Type: (x,y,z) | Point3d() **header**: Header part of the data message. Uses header from the first input if not set.

Logging

Logging a message causes GoPxL to add a message to the log at the bottom of the user interface. There are three types of log messages.

Logging Functions

Function	Description
log_info(msg)	Logs a user informational message. *Parameters* **msg**: Message to be logged.
log_warning(msg)	Logs a user warning message. *Parameters* **msg**: Message to be logged.
log_error(msg)	Logs a user error message and displays a pop-up containing the message. *Parameters* **msg**: Message to be logged.

Memory

You can use a Script tool's `memory` dictionary for storing values during runtime. The dictionary is erased when scanning starts.

> Some classes are only available through the `tl` namespace (for example, `tl.Anchor3d`). These classes are indicated below.

Data Structures

Data Structures

Data Structure	Description
Region	2D Region box (center-based). *Constructor*

Data Structure	Description
	Region(x=None, z=None, width=None, height=None) *Parameters* **x**: X position of the center of the region. **z**: Z position of the center of the region. **width**: Width of the region. **height**: Height of the region.
Region3d	3D Region box (center-based). *Constructor* Region3d(x=None, y=None, z=None, width=None, length=None, height=None, z_angle=None) *Parameters* **x**: X position of the center of the region. **y**: Y position of the center of the region. **z**: Z position of the center of the region. **width**: Width of the region. **height**: Height of the region. **length**: Length of the region. **z_angle**: Z Angle of the region.
Rect3d	3D rectangular region box. *Constructor* Rect3d(x=None, y=None, z=None, width=None, height=None, depth=None) *Parameters* **x**: X position of the center of the region. **y**: Y position of the center of the region. **z**: Z position of the center of the region. **width**: Width of the region. **height**: Height of the region. **depth**: Depth of the region.
tl.Anchor3d	3D anchor (center-based). Use the `Anchor3d` class to receive anchor information as *input*. For anchor outputs, use Measurement outputs instead. This class is only accessible through the `tl` namespace (`tl.Anchor3d`). *Constructor* Anchor3d(x=None, y=None, z=None, z_angle=None) *Parameters* **x**: X position of the anchor. **y**: Y position of the anchor. **z**: Z position of the anchor. **z_angle**: Z Angle of the anchor.
Point	2D Point class.

Data Structure	Description
	Constructor Point(x=None, y=None) *Parameters* **x**: X position of the Point. **y**: Y position of the Point.
Point3d	3D Point class. *Constructor* Point3d(x=None, y=None, z=None) *Parameters* **x**: X position of the Point. **y**: Y position of the Point. **z**: Z position of the Point.
tl.Stamp	Represents acquisition metadata associated with a data message. Note that `width`, `height`, `x`, and `y` parameters refer to the dimensions of the camera image; for more information, see `points` in *tl.SurfaceMsg* on page 686. *Constructor* Stamp(frame=None, time=None, signature=None, status=None, state=None, tag=None, encoder=None, encoder_snapshot=None, exposure=None, delay=None, width=None, height=None, y=None, x=None, x_subsampling=None, y_subsampling=None, temperature=None, subframe=None, reserved=None) *Parameters* **signature**: Equal to `kSTAMP_SIGNATURE` for camera messages; otherwise unused. **status**: Digital inputs, gate state, and other information at the time of message capture. **state**: Camera state id associated with message. **tag**: Optional tag that can be used for debugging or other special purposes. **frame**: Frame index of message data (counts up from zero). **time**: Timestamp corresponding to message data (µs). **encoder**: Encoder value corresponding to message data (encoder ticks). **encoder_snapshot**: Encoder value at time of most recent encoder snapshot signal (encoder ticks). **exposure**: Source image exposure (ns). **delay**: Delay before exposure (ns). **width**: Source camera image width, before subsampling. **height**: Source camera image height, before subsampling. **y**: Source camera image Y position. **x**: Source camera image X position. **x_subsampling**: X subsampling amount, expressed as power of 2. **y_subsampling**: Y subsampling amount, expressed as power of 2.

Data Structure	Description
	temperature: Temperature corresponding to message data (centidegrees Celsius). **subframe**: Subframe status information. **reserved**: Reserved for future use.
Transform3d	3D transform matrix. *Constructor* Transform3d(xx=None, xy=None, xz=None, xt=None, yx=None, yy=None, yz=None, yt=None, zx=None, zy=None, zz=None, zt=None) *Parameters* **xx**, **xy**, **xz**, **xt**: X vector (first three) and origin. **yx**, **yy**, **yz**: Y vector and origin. **zx**, **zy**, **zz**: Z vector and origin.
tl.MsgHeader	Header class of data message. *Constructor* MsgHeader(stamp=None, transform=None, bounding_box=None) *Parameters* **stamp**: Data stamp. **transform**: Transform matrix. **bounding_box**: Bounding box for the data.
PixelFormat	Enum class of pixel formats for image data message. **Mono formats** MONO_8 = <PixelFormat.MONO_8: 0x0001> MONO_10 = <PixelFormat.MONO_10: 0x0003> MONO_12 = <PixelFormat.MONO_12: 0x0005> MONO_14 = <PixelFormat.MONO_14: 0x0025> MONO_16 = <PixelFormat.MONO_16: 0x0007> **RGB formats** RGB_8 = <PixelFormat.RGB_8: 0x0014> BGR_8 = <PixelFormat.BGR_8: 0x0015> RGB_10 = <PixelFormat.RGB_10: 0x0018> BGR_10 = <PixelFormat.BGR_10: 0x0019> RGB_12 = <PixelFormat.RGB_12: 0x001A> BGR_12 = <PixelFormat.BGR_12: 0x001B> RGB_14 = <PixelFormat.RGB_14: 0x005E> BGR_14 = <PixelFormat.BGR_14: 0x004A> RGB_16 = <PixelFormat.RGB_16: 0x0033>

Data Structure	Description
	BGR_16 = <PixelFormat.BGR_16: 0x004B> **YUV/YCBR formats** Y_CB_CR_8_CB_Y_CR = <PixelFormat.Y_CB_CR_8_CB_Y_CR: 0x003A> Y_CB_CR_411_8_CB_YY_CR_YY = <PixelFormat.Y_CB_CR_411_8_CB_YY_CR_YY: 0x003C> Y_CB_CR_422_8_CB_Y_CR_Y = <PixelFormat.Y_CB_CR_422_8_CB_Y_CR_Y: 0x0043> **Bayer formats** BAYER_GR8 = <PixelFormat.BAYER_GR8: 0x0008> BAYER_RG8 = <PixelFormat.BAYER_RG8: 0x0009> BAYER_GB8 = <PixelFormat.BAYER_GB8: 0x000A> BAYER_BG8 = <PixelFormat.BAYER_BG8: 0x000B> BAYER_GR10 = <PixelFormat.BAYER_GR10: 0x000C> BAYER_RG10 = <PixelFormat.BAYER_RG10: 0x000D> BAYER_GB10 = <PixelFormat.BAYER_GB10: 0x000E> BAYER_BG10 = <PixelFormat.BAYER_BG10: 0x000F>
tl.ImageMsg	ImageMsg class for image data message. *Constructor* ImageMsg(header=None, pixel_format=None, flipped_x=False, flipped_y=False, transposed=False, pixels=None, offset = None, scale = None, unit = None) *Parameters* **header**: Header part of the data message. **pixel_format**: Pixel format of the image. For more information, see *PixelFormat* on the previous page. **flipped_x**: Set to `true` if the image data is flipped on the X axis. **flipped_y**: Set to `true` if the image data is flipped on the Y axis. **transposed**: Set to `true` if the image data is transposed. **pixels**: Pixel data of image of size (height, width, pixel size). **offset**: Offset of the image, or array of images. Type Point(). **scale**: Scale of the image, or array of images. Type Point(). **unit**: Unit for spatial dimensions, or array of images. Type UnitType enum.
UnitType	Enum class of units for spatial dimensions. NONE = <UnitType.NONE: 0> MILLIMETER = <UnitType.MILLIMETER: 1> PIXEL = <UnitType.PIXEL: 2>

Data Structure	Description
	Properties **label**: Return the string label of the unit. Useful for display or formatting.
tl.SurfaceMsg	SurfaceMsg class for surface data message. *Constructor* SurfaceMsg(header=None, offset=None, scale=None, points=None, intensity=None, is_adjacent=None) *Parameters* **header**: Header part of the data message. **offset**: Offset of the data. **scale**: Scale of the data. **points**: Surface point data of size (height, width) or (height,width,3). **intensity**: Surface intensity data of size (height, width). **is_adjacent**: Set to `true` if the surface data is adjacent. **intensity_format**: Intensity data type (PixelFormat).
tl.ProfileMsg	ProfileMsg class for profile data message. *Constructor* ProfileMsg(header=None, offset=None, scale=None, points=None, intensity=None) *Parameters* **header**: Header part of the data message. **offset**: Offset of the data. **scale**: Scale of the data (with x and z). This is the spacing between points. **points**: Profile point data of size (width) or (width,2) . **intensity**: Profile intensity data of size (width).
tl.MeasureMsg	MeasureMsg Class for measurement data message. *Constructor* MeasureMsg(header=None, value=None, status=None, label_position=None) *Parameters* **header**: Header part of the data message. **value**: Value of the data message. **status**: Types from the MeasurementDecision enum class. For more information, see *MeasurementDecision* below. **label_position**: Position of point in measurement message.
MeasurementDecision	Enum class of decision type for measurement data message. FAIL= <MeasurementDecision.FAIL: 0> PASS= <MeasurementDecision.PASS: 1>

Data Structure	Description
tl.PointFeatureMsg	PoinFeaturetMsg class for point feature message. *Constructor* PointFeatureMsg(header=None, position=None) *Parameters* **header**: Header part of the data message. **position**: Position of feature point in data message.
tl.LineFeatureMsg	LineFeatureMsg class for line feature message. *Constructor* LineFeatureMsg(header=None, position=None, direction=None) *Parameters* **header**: Header part of the data message. **position**: Point vector in the line. **direction**: Direction vector of the line.
tl.CircleFeatureMsg	CircleFeatureMsg class for circular feature message. *Constructor* CircleFeatureMsg(header=None, position=None, normal=None, radius=None) *Parameters* **header**: Header part of the data message. **position**: Center position of the circle. **normal**: Normal vector of the circular plane. **radius**: Radius of the circle.
tl.PlaneFeatureMsg	PlaneFeatureMsg class for planar feature message. *Constructor* PlaneFeatureMsg(header=None, normal=None, origin_distance=None) *Parameters* **header**: Header part of the data message. **normal**: Normal vector of the plane. **origin_distance**: Shortest distance from origin to plane.
tl.StringMsg	StringMsg class. *Constructor* StringMsg(header=None, text=None, decision=None, label_position=None) *Parameters* **header**: Header part of the data message. **text**: The string. **decision**: Decision of measurement message. **label_position**: Position of point in measurement message.
tl.MeshMsg	Sends a mesh message or array of messages to the specified output. *Parameters*

Data Structure	Description
	header: Header part of the data message. Type: MsgHeader(). **vertex_data**: Vertex data points. Type: numpy array of floats (x,y,z). **facet_data**: Facet data points. Type: numpy array of uint32 triangle indexes (vertex 1, vertex 2,vertex 3). **vertex_normal_data**: Vertex normal data points. Type: numpy array of floats (x,y,z). **facet_normal_data**: Facet normal data points. Type: numpy array of floats (x,y,z). **vertex_texture_data**: Vertex texture data. Type: numpy array of uint8. **data_offset**: Offset of the data or array of offsets. Type: Type: (x,y,z) | Point3d() **data_range**: Range of the data or array of ranges. Type: (x,y,z) | Point3d()
MarkerShape(Enum)	Marker shape types for graphics. CROSS = 0 PLUS = 1 CIRCLE = 2 SQUARE = 3 TRIANGLE = 4 DIAMOND = 5

Examples

Measurement input and output

The following example shows how to read a measurement value input and output a new measurement value that is equal to the input multiplied by a factor of two.

```
# Get the measurement object at input 0
measurement = get_measurement(0)

# Extract the value from the measurement
measurement_value = measurement.value

# Calculate a new measurement value
measurement_output = 2 * measurement_value

# Send the calculated value as a new measurement to output 0
send_measurement(0, measurement_output)
```

The following example shows how to read measurement value inputs and write them to a CSV file. (This example is only intended for use on a PC instance of GoPxL, as writing to files on-sensor or on a GoMax device is blocked.)

```
import csv

# Get measurement objects
measurement_1 = get_measurement(0)
measurement_2 = get_measurement(1)

# Extract values from measurement objects
measurement_1_value = measurement_1.value
measurement_2_value = measurement_2.value

# Path to the output CSV file
output_path = r'C:\GoTools\Script\measurement.csv'

# Write measurements to CSV file
with open(output_path, 'a') as f:
    writer = csv.writer(f)
    writer.writerow([measurement_1_value, measurement_2_value])
```

Math operations

You can access mathematical operations through the standard Python math module.

```
# Import the math module to access mathematical functions
import math

# Get the measurement object at input 0
measurement = get_measurement(0)

# Extract the value from the measurement
measurement_value = measurement.value

# Calculate the square root of the absolute value using the Math module
sqrt_value = math.sqrt(math.fabs(measurement_value))

# Send the square root as a new measurement to output 0
send_measurement(0, sqrt_value)
```

Accessing valid/invalid and decision pass/fail

You can use the `is_valid()` function of the `MeasureMsg` class to check whether the value is valid or invalid. An invalid measurement is represented by the value `numpy.nan`. Other input classes also have `is_valid()` functions, which return `true` if the object data is not null.

The pass/fail criteria for the input `MeasureMsg` can be checked using the `decision` member variable.

```
# Check if any inputs are invalid
if any_input_invalid():
    send_all_invalid()
```

```
# Get the input measurement
measurement0 = get_measurement(0)

# Access validity from a measurement
if is_valid(measurement0):
    log_info("The input measurement is Valid")
else:
    log_info("The input measurement is Invalid")

# Access decision from a measurement
if measurement0.decision == MeasurementDecision.PASS:
    log_info("The input measurement has a Pass decision")
else:
    log_info("The input measurement has a Failed decision")
```

Smoothing / moving average of measurement values

This script shows how to use the memory feature to implement a smoothing filter / moving average of measurement values. Values from each frame are stored in the memory and with each new frame the oldest item is removed from memory before calculating the average.

```
import numpy as np

measurement_1 = get_measurement(0)
new_value = measurement_1.value
#log_info(f'Measurement values: {new_value}')

class MovingAverage:
    def __init__(self, window_size=10):
        self.window_size = window_size

    def add_value(self, value):
        memory.setdefault("Values", []).append(value)
        if len(memory["Values"]) > self.window_size:
            memory["Values"].pop(0)
        return self.calculate_average()

    def calculate_average(self):
        if len(memory["Values"]) == 0:
            return np.nan  # if no value present
        return np.mean(memory["Values"])

moving_avg = MovingAverage(window_size=10)
avg = moving_avg.add_value(new_value)
# log_info(f'Avg: {avg}')

send_measurement(0, avg)
```

Stamp access

```
profile = get_profile(0)

stamp = profile.header.stamp

log_info('Stamp = {}'.format(vars(stamp)))

send_measurement(0, stamp.frame)
send_measurement(1, stamp.time)
send_measurement(2, stamp.encoder)
```

Arrays

The following code reads an array input and outputs an array.

```
# Get an array of measurement objects at input 0
measurements = get_measurement(0)

# Extract the values from each measurement object
measurement_values = []
for measurement in measurements:
    measurement_values.append(measurement.value)

# Calculate new values
measurements_output = [value * 2 for value in measurement_values]

# Send the calculated values as an array to output 0
send_measurement(0, measurements_output)
```

Inverting a Surface in Z with null handling

The following code inverts a Surface in Z with null handling. It also shifts the output Surface in X by the bounding box width in order to not overlap the original.

```
surf = get_surface(0)

flipped_points = surf.points.copy()
flipped_offset = surf.offset

flipped_points[flipped_points != -32768] *= -1
flipped_offset.x += surf.header.bounding_box.width


send_surface(0, flipped_offset, surf.scale, flipped_points, surf.intensity, surf.is_
adjacent, None, surf.intensity_format)
```

Combining two point cloud surfaces into a single message

```
s0 = get_surface(0)
s1 = get_surface(1)

if s0.scale.x != s1.scale.x or s0.scale.y != s1.scale.y or s0.scale.z != s1.scale.z or
s0.offset.x != s1.offset.x or s0.offset.y != s1.offset.y or s0.offset.z != s1.offset.z:
    log_error("Two input surfaces are not using the same resolution and offset. Cannot
concatenate")
    log_info(f"Scale0 = {vars(s0.scale)}")
    log_info(f"Scale1 = {vars(s1.scale)}")

    log_info(f"Offset0 = {vars(s0.offset)}")
    log_info(f"Offset1 ={vars(s1.offset)}")

    return


points_final = numpy.concatenate((s0.points, s1.points))
intensity_final = numpy.concatenate((s0.intensity, s1.intensity))

# set is_adjacent to True to visualize polygons and intensity
send_surface(0, s0.offset, s0.scale, points_final, intensity_final, is_adjacent = False,
             intensity_format=s0.intensity_format)
```

Surface Transform based on Plane

```
surface_obj = get_surface(0)
plane_obj = get_plane(1)
int_surface_points = surface_obj.points
intensity_data = surface_obj.intensity

# Store original shape and reshape for processing
original_shape = int_surface_points.shape
num_points = original_shape[0] * original_shape[1]
points_reshaped = int_surface_points.reshape((num_points, 3))

# Convert 3DPoint objects to numpy arrays for calculations
scale = numpy.array([surface_obj.scale.x, surface_obj.scale.y, surface_obj.scale.z])
offset = numpy.array([surface_obj.offset.x, surface_obj.offset.y, surface_obj.offset.z])
normal_vec = numpy.array([plane_obj.normal.x, plane_obj.normal.y, plane_obj.normal.z])

# Convert integer surface to floating-point "world space"
float_surface_points = points_reshaped.astype(float) * scale + offset

# Normalize the normal vector
norm_of_normal = numpy.linalg.norm(normal_vec)
if norm_of_normal == 0:
    raise ValueError("The normal vector cannot be a zero vector.")
unit_normal = normal_vec / norm_of_normal

# Find the center of the object
center_point = float_surface_points.mean(axis=0)
# Find a point on the plane to establish a reference
point_on_plane = plane_obj.origin_distance * unit_normal
```

```
# Determine if the normal points towards or away from the center
vec_to_center = center_point - point_on_plane
if numpy.dot(vec_to_center, unit_normal) > 0:
    # The normal points towards the center of the object. We flip it to get the
    # "outward" normal, ensuring the plane is consistently treated as the "top".
    unit_normal = -unit_normal

# Define the target vector
down_vector = numpy.array([0.0, 0.0, -1.0])

# Calculate the rotation matrix to align the normal with the down_vector
rotation_axis = numpy.cross(unit_normal, down_vector)
cos_angle = numpy.dot(unit_normal, down_vector)

if numpy.allclose(cos_angle, 1.0):
    rotation_matrix = numpy.identity(3)
elif numpy.allclose(cos_angle, -1.0):
    perp_axis = numpy.array([1.0, 0.0, 0.0])
    if numpy.allclose(numpy.abs(numpy.dot(unit_normal, perp_axis)), 1.0):
        perp_axis = numpy.array([0.0, 1.0, 0.0])
    u = perp_axis / numpy.linalg.norm(perp_axis)
    rotation_matrix = 2 * numpy.outer(u, u) - numpy.identity(3)
else:
    # shortcut to the sin between the vectors (based on cross product definition)
    sin_angle = numpy.linalg.norm(rotation_axis)
    rotation_axis /= sin_angle
    K = numpy.array([[0, -rotation_axis[2], rotation_axis[1]],
                     [rotation_axis[2], 0, -rotation_axis[0]],
                     [-rotation_axis[1], rotation_axis[0], 0]])
    rotation_matrix = numpy.identity(3) + sin_angle * K + (1 - cos_angle) * (K @ K)

# Create the full 4x4 homogeneous transformation matrix for rotation only
translation_to_origin = numpy.identity(4)
translation_to_origin[:3, 3] = -center_point
rotation_4x4 = numpy.identity(4)
rotation_4x4[:3, :3] = rotation_matrix
translation_back = numpy.identity(4)
translation_back[:3, 3] = center_point
transformation_matrix = translation_back @ rotation_4x4 @ translation_to_origin

homogeneous_surface = numpy.hstack([float_surface_points, numpy.ones((num_points, 1))])
transformed_homogeneous = (transformation_matrix @ homogeneous_surface.T).T
transformed_float_surface = transformed_homogeneous[:, :3]

# Calculate a new offset to prevent overflow
min_coords = transformed_float_surface.min(axis=0)
max_coords = transformed_float_surface.max(axis=0)
new_offset_np = (min_coords + max_coords) / 2.0

transformed_float_to_int = (transformed_float_surface - new_offset_np) / scale

# Clip and cast back to integer space as a safety measure
dtype_info = numpy.iinfo(int_surface_points.dtype)
min_val, max_val = dtype_info.min, dtype_info.max
numpy.clip(transformed_float_to_int, min_val, max_val, out=transformed_float_to_int)

transformed_int_reshaped = numpy.round(transformed_float_to_int).astype(int_surface_
points.dtype)
```

```
transformed_int_surface = transformed_int_reshaped.reshape(original_shape)
OffsetPointClass = type(surface_obj.offset)
new_offset_obj = OffsetPointClass(new_offset_np[0], new_offset_np[1], new_offset_np[2])

send_surface(
    0,
    new_offset_obj,          # Send the NEW offset
    surface_obj.scale,       # Send the ORIGINAL scale
    transformed_int_surface,
    intensity_data,
    is_adjacent=False,
    intensity_format=surface_obj.intensity_format
)
```

Minimum and maximum Z value of a uniform Surface with null handling

```
surf = get_surface(0)

s = surf.points[surf.points != -32768]

max_z = numpy.max(s) * surf.scale.z + surf.offset.z
min_z = numpy.min(s) * surf.scale.z + surf.offset.z

send_measurement(0, min_z)
send_measurement(1, max_z)
```

Exporting a surface as a PCD file

```
import numpy as np


filename = "outsurface.pcd"
msg = get_surface(0)
height_map = msg.points
intensity_map = msg.intensity
scale = msg.scale


rows, cols = height_map.shape


# Prepare the header for the PCD file
header = (
    "# .PCD v0.7 - Point Cloud Data file format\n"
    "VERSION 0.7\n"
    "FIELDS x y z intensity\n"
    "SIZE 4 4 4 4\n"
    "TYPE F F F F\n"
    "COUNT 1 1 1 1\n"
    f"WIDTH {cols}\n"
```

```
    f"HEIGHT {rows}\n"
    "VIEWPOINT 0 0 0 1 0 0 0\n"
    f"POINTS {rows * cols}\n"
    "DATA ascii\n"
)


# Write the header and points to the PCD file
with open(filename, 'w') as file:
    file.write(header)
    for i in range(rows):
        for j in range(cols):
            x = j * scale.x
            y = i * scale.y
            z = height_map[i, j] * scale.z if height_map[i, j] != -32768 else 0
            intensity = intensity_map[i, j] if height_map[i, j] != -32768 else 0
            file.write(f"{x} {y} {z} {intensity}\n")


log_info(f"PCD file saved successfully as {filename}")
```

Grid of coordinates for anchoring

The following script shows how you can perform the same inspection using a single tool at many locations on a grid. It generates X and Y measurement arrays that you use as anchor inputs in another tool (the anchored tool).

In the anchored tool, you must check **Enable batching** so that the tool generates an array of outputs.

```
x = numpy.linspace(-15, 15, 10) # 10 points spanning 0..30 mm range
y = numpy.linspace(-15, 15, 10) # 10 points spanning 0..30 mm range

xv, yv = numpy.meshgrid(x, y)

send_measurement(0, xv.flatten())
send_measurement(1, yv.flatten())
```

Image Processing (Subtracting two images and normalizing the result)

```
im0 = get_image(0)
im1 = get_image(1)

delta = im0.pixels.astype(numpy.float32) - im1.pixels.astype(numpy.float32)
stdev = numpy.std(delta)

new_pixels = numpy.clip(delta / (2*stdev) * 128 + 128, 0, 255).astype(numpy.uint8)

send_image(0, new_pixels, PixelFormat.RGB_8)
send_measurement(1, stdev)
```

Image processing (forward to HTTP server)

```
# Script (using built in urlib package)
import urllib.request
import json
import numpy as np

# Assuming get_image(0).pixels returns a numpy array
array = get_image(0).pixels
array_bytes = array.tobytes()

# Encode shape information as JSON in the headers
headers = {'Content-Type': 'application/octet-stream', 'X-Array-Shape': json.dumps
(array.shape)}
url = 'http://127.0.0.1:5000/receive_array'

# Encode array bytes data and prepare request
req = urllib.request.Request(url, data=array_bytes, headers=headers)
urllib.request.urlopen(req)


# SERVER (server.py)

from flask import Flask, request, Response
import numpy as np
import cv2
import json

app = Flask(__name__)

@app.route('/receive_array', methods=['POST'])
def receive_array():
    # Get the content of the request
    data = request.data

    # Extract shape information from headers
    shape_info = json.loads(request.headers.get('X-Array-Shape'))
    height, width, channels = shape_info

    # Extract image bytes
    image_array = np.frombuffer(data, dtype=np.uint8)
    image_array = image_array.reshape((height, width, channels))

    # Display image
    #cv2.imshow('Received Image', image_array)
    #cv2.waitKey(0)
    #cv2.destroyAllWindows()

    # Print size of received image
    print("Received image size:", image_array.shape)

    return Response(status=200)

if __name__ == '__main__':
    app.run(debug=False, threaded=True, port=5000)
```

The following is a sample client to demonstrate the functionality of the HTTP server forwarding snippet above.

```
# SAMPLE CONSOLE CLIENT (client.py)
# A standalone demonstration of the functionality used by the image HTTP
# forwarding script.

import requests
import numpy as np
import json

# Assuming you have a function to generate numpy arrays as images
def generate_numpy_array():
    # Generating a sample numpy array
    array = np.random.rand(100, 100, 3) * 255  # Random 100x100x3 array of floats between
0 and 255
    return array.astype(np.uint8)

def send_array_to_server(array):
    url = 'http://127.0.0.1:5000/receive_array'
    array_bytes = array.tobytes()

    # Encode shape information as JSON in the headers
    headers = {'Content-Type': 'application/octet-stream', 'X-Array-Shape': json.dumps
(array.shape)}

    response = requests.post(url, data=array_bytes, headers=headers)

# Generate and send arrays to the server
while True:
    array = generate_numpy_array()
    send_array_to_server(array)
```

Image Processing - Warp an image based on four corners

```
import cv2
import numpy as np

def warp_image_to_rectangle(image: np.ndarray, src_corners: np.ndarray, output_size:
tuple) -> np.ndarray:
    """
    Resample an image to a rectangle based on four corner points.

    :param image: Input image as a NumPy array.
    :param src_corners: Source corner points as a (4, 2) NumPy array (float32).
    :param output_size: Desired output size (width, height).
    :return: Warped and resampled image.
    """
    if src_corners.shape != (4, 2):
        raise ValueError("src_corners must be a (4, 2) array of corner points.")

    width, height = output_size

    # Define destination rectangle corners
    dst_corners = np.array([
```

```
        [0, 0],
        [width - 1, 0],
        [width - 1, height - 1],
        [0, height - 1]
    ], dtype=np.float32)

    # Compute perspective transform matrix
    matrix = cv2.getPerspectiveTransform(src_corners.astype(np.float32), dst_corners)

    # Warp the image
    warped = cv2.warpPerspective(image, matrix, (width, height), flags=cv2.INTER_LINEAR)

    return warped

if any_input_invalid():
    send_all_invalid()
    return

image = get_image(0).pixels
c0 = get_point(1).position
c1 = get_point(2).position
c2 = get_point(3).position
c3 = get_point(4).position

src_corners = np.array([
    [c0.x, c0.y],
    [c1.x, c1.y],
    [c2.x, c2.y],
    [c3.x, c3.y]
], dtype=np.float32)

output_size = (800, 400)
result = warp_image_to_rectangle(image, src_corners, output_size)
result = result.reshape((output_size[1], output_size[0], 1))

send_image(0, result, PixelFormat.MONO_8)
```

Image Processing (Sending out an image with calibrated units)

```
im = get_image(0)
h = im.header
offset = Point(im.offset.x - 100 , im.offset.y - 100)
scale = Point(im.scale.x / 10, im.scale.y / 10)

send_image(0, im.pixels, pixel_format= PixelFormat.RGB_8,
           offset = offset, scale = scale, unit = UnitType.MILLIMETER)
```

Profile processing (using a profile from disk as a correction to new profiles from sensor)

```
# Save profile template to disk
import csv
```

```
profile = get_profile(0)
points = profile.points

template_path = r'C:\GoTools\Script\profile.csv'
numpy.savetxt(template_path, points, delimiter=',', fmt='%d')

send_measurement(0, 1)

# Load and correct
import csv

profile = get_profile(0)
points = profile.points

# load template
template_path = r'C:\GoTools\Script\profile_template.csv'

if 'template' not in memory:
    template_points = numpy.genfromtxt(template_path, delimiter=',', dtype=numpy.int16)
    memory['template'] = template_points
    log_info("Loaded template")
else:
    template_points = memory['template']

# subtract while handling null values
mask = (template_points != -32768) & (points != -32768)
result = numpy.full_like(points, -32768)
numpy.subtract(points, template_points, out=result, where=mask, dtype=numpy.int16)

send_profile(0, profile.offset, profile.scale, result)

# optionally send original profile and template as outputs
send_profile(1, profile.offset, profile.scale, points)
send_profile(2, profile.offset, profile.scale, template_points)

# optionally save all three into a new CSV file

output_path = r'C:\GoTools\Script\profile_log.csv'
combined_array = numpy.vstack((points, template_points, result)).T

with open(output_path, 'w', newline='') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['Profile', 'Template', 'Difference'])
    writer.writerows(combined_array)
```

Getting function signatures

Instead of looking up functions in the manual to see arguments to be passed in, you can use the `signature` function in the `inspect` module to get the signature of any function. For example, the code below displays an error message: (`index, value, label_position=None, header=None`).

```
from inspect import signature
```

```
log_error(str(signature(send_measurement)))
```

FTP - Writing surface data to disk, and sending it to an FTP server

This sample was tested with Rebex Tiny FTP server.

```
import os
import struct
import zlib
from datetime import datetime
from ftplib import FTP
import numpy as np
import time

# FTP Configuration
FTP_HOST = '192.168.1.200'       # CHANGE THIS TO THE FTP HOSTNAME OR IP ADDRESS
FTP_USER = 'tester'              # Default User for Rebex Tiny FTP
FTP_PASS = 'password'            # Default Password for Rebex Tiny FTP

try:
    # Get the surface data
    surface = get_surface(0)

    # Convert surface points to numpy array (height data)
    height_data = np.array(surface.points)

    # Intensity data (already in the 0-255 range, so no normalization needed)
    intensity_data = np.array(surface.intensity)

    # Normalize height data to 0-255, handle NaNs, and convert to uint8
    def normalize(data):
        return np.nan_to_num(((data - np.nanmin(data)) / (np.nanmax(data) - np.nanmin
(data) + 1e-8)) * 255).astype(np.uint8)

    # Correct orientation (flip vertically and transpose)
    def correct_orientation(arr):
        return np.flipud(arr.T)

    # Normalize and correct the orientation for height data
    height_img = correct_orientation(normalize(height_data))

    # Intensity data is already in the range 0-255, just ensure it is uint8
    intensity_img = correct_orientation(intensity_data.astype(np.uint8))

    # Write the PNG image to the specified path
    def write_png(path, image):
        try:
            if image.size == 0:
                log_info("Image is empty. Skipping write.")
                return

            # Helper function to create PNG chunks
            def chunk(typ, dat):
                return struct.pack('>I', len(dat)) + typ + dat + struct.pack('>I',
zlib.crc32(typ + dat) & 0xffffffff)
```

```
            # Open the file and write the PNG data
            with open(path, 'wb') as f:
                f.write(b'\x89PNG\r\n\x1a\n')
                f.write(chunk(b'IHDR', struct.pack(">IIBBBBB", image.shape[1], image.shape
[0], 8, 0, 0, 0, 0)))  # Write header chunk
                raw = b''.join([b'\x00' + image[i].tobytes() for i in range(image.shape
[0])])  # Write pixel data
                f.write(chunk(b'IDAT', zlib.compress(raw)))  # Write compressed image data
                f.write(chunk(b'IEND', b''))  # End of PNG file
                f.flush()  # Ensure data is written
                os.fsync(f.fileno())  # Sync the file to disk

        except Exception as e:
            log_info(f"Exception in write_png: {e}")

    # Define output directory and file paths for saving images
    output_dir = r'C:\Temp'  # Use a simpler directory for testing
    os.makedirs(output_dir, exist_ok=True)  # Create the output directory if it doesn't
exist
    timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
    path_height = os.path.join(output_dir, f"{timestamp}_height.png")
    path_intensity = os.path.join(output_dir, f"{timestamp}_intensity.png")

    # Write the height and intensity images to files
    try:
        write_png(path_height, height_img)
        log_info(f"Height image written: {path_height}")
    except Exception as e:
        log_info(f"Error writing height image: {e}")

    try:
        write_png(path_intensity, intensity_img)
        log_info(f"Intensity image written: {path_intensity}")
    except Exception as e:
        log_info(f"Error writing intensity image: {e}")

    # Function to wait for file size to stabilize before proceeding with upload
    def wait_for_file_stability(path, check_interval=0.1, max_wait_time=2):
        last_size = -1
        start_time = time.time()

        # Continuously check if the file size has stabilized within the given timeout
        while time.time() - start_time < max_wait_time:
            try:
                current_size = os.path.getsize(path)
                if current_size == last_size:
                    # If file size is stable, exit the loop
                    return True
                last_size = current_size
                time.sleep(check_interval)  # Sleep for a very short interval before
checking again
            except FileNotFoundError:
                pass  # Ignore if the file doesn't exist yet
        return False

    # FTP upload function to send files
    def upload(path):
        try:
```

```
            with FTP(FTP_HOST) as ftp:
                ftp.login(FTP_USER, FTP_PASS)
                with open(path, 'rb') as f:
                    ftp.storbinary(f'STOR {os.path.basename(path)}', f)  # Upload the file
            return 1
        except Exception as e:
            log_info(f"FTP error: {e}")
            return 0

    # Upload both height and intensity images via FTP
    send_measurement(0, upload(path_height))
    send_measurement(1, upload(path_intensity))

except Exception as e:
    log_info(f"Error in processing surface data: {e}")
    send_measurement(0, 0)  # Indicate failure if an exception occurs during processing
```

Moving Average Using Memory

```
import numpy as np


class MovingAverage:
    def __init__(self, window_size=10):
        self.window_size = window_size

    def add_value(self, value):

        # Add value if the 'Values' key is present else initialize with an empty list.
        memory.setdefault("Values", []).append(value)
        if len(memory["Values"]) > self.window_size:
            memory["Values"].pop(0)
        return self.calculate_average()

    def calculate_average(self):
        if not memory["Values"]:
            return np.nan
        return np.mean(memory["Values"])


# Get values
measurement = get_measurement(0)
measurement_value = measurement.value

# Process values
moving_avg = MovingAverage(window_size=10)
avg = moving_avg.add_value(measurement_value)

# Send values
send_measurement(0, measurement_value)
send_measurement(1, avg)
```

Communicate with Another Sensor via ASCII Using Socket Programming

```
import socket

HOST = "192.168.1.10"
PORT = 8190

DELIMITER = ','
TERMINATION = '\r\n'  # Matches %r%n set by default in GoPxL > Ethernet ASCII page

START_COMMAND = f'start{TERMINATION}'
STOP_COMMAND = f'stop{TERMINATION}'
TRIGGER_COMMAND = f'trigger{TERMINATION}'
SYSTEM_STATE_COMMAND = f"readprop{DELIMITER}/system#/runState{TERMINATION}"

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(15)
    s.connect((HOST, PORT))
    log_info(f"Connected to sensor at {HOST}:{PORT}")

    _ = s.send(SYSTEM_STATE_COMMAND.encode()) # Requires data to be sent in byte format
    response = s.recv(1024).decode().strip().split(',') # Receives up to 1024 bytes of
data from the socket connection
    log_info(f"Response received: {response}")

    if len(response) > 1 and response[0] == "OK":
        run_state = response[1]
        if run_state == "0": # If sensor is not running, first run then trigger
            _ = s.send(START_COMMAND.encode())
            _ = s.send(TRIGGER_COMMAND.encode())
        elif run_state == "1":
            _ = s.send(TRIGGER_COMMAND.encode())
        else:
            log_info(f"Unexpected system state: {run_state}")

except socket.timeout:
    log_error(f"Error: Operation timed out. ASCII is disabled or sensor is unreachable at
{HOST}:{PORT}.")
except Exception as e:
    log_error(f"An error occurred: {e}")
finally:
    s.close()
    log_info("Socket closed.")
```

Save screenshot with the displayed outputs in the data viewer

Before using this sample, you must install two python modules: selenium and webdriver_manager. Note that this script is intended for use on GoPxL for PC.

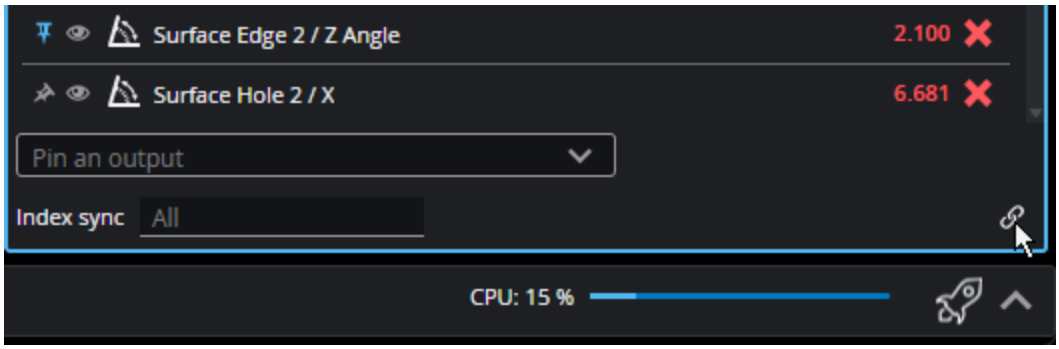
1. Using Command Prompt or PowerShell, go to the `GoPxl_UserTools\bin\win64\python` folder.
2. (Optional) To show the installed Python modules, `run python3.exe -m pip list`.
3. To install selenium, run `python3.exe -m pip install selenium`.
4. To install webdriver_manager, run `python3.exe -m pip install webdriver_manager`.

To save a screenshot in a script, after adding a Script tool and copying and pasting the sample code into the **Code** field, do the following:

1. If it's not already open, expand the **Displayed outputs** pane.
2. Configure which outputs you want displayed in the data viewer by pinning or hiding outputs.

   For more information on the Displayed outputs pane, see *Displaying and Pinning Outputs* on page 94.
3. Click the link icon at the lower right.

   This copies the current state of the data viewer, including the currently .



4. Paste the link into the code as the value of `url`.

The script saves a .png file to the location in the script.

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from webdriver_manager.chrome import ChromeDriverManager
from datetime import datetime
import os

if not hasattr(self, 'long_mem'):
    self.long_mem = {}

if "service" not in self.long_mem:
    self.long_mem["service"] = Service(ChromeDriverManager().install())
service = self.long_mem["service"]

# Set the URL here
url = "www.google.com"
log_info(f"{type(url)}")

if "driver" not in self.long_mem:
    chrome_options = Options()
    chrome_options.add_argument("--headless=new")  # Use "--headless=new" for newer
versions of Chrome
    chrome_options.add_argument("--window-size=1920,1080")  # Important for full
screenshots
    #chrome_options.add_argument("--disable-gpu")  # Optional: helpful on Windows

    self.long_mem["driver"] = webdriver.Chrome(service=service, options=chrome_options)
    self.long_mem["last_url"] = url  # Initialize last_url to None
```

```
# Check if the URL has changed
if self.long_mem["last_url"] != url:
    log_info(f"changing url to {url}")
    self.long_mem["driver"].get(url)
    self.long_mem["last_url"] = url  # Update last_url to the current URL

driver = self.long_mem["driver"]

# Create a timestamped filename
timestamp = datetime.now().strftime('%Y-%m-%d_%H-%M-%S')
filename = f"screenshot_{timestamp}.png"

# Optional: specify a directory to save screenshots
output_dir = os.path.expanduser("~/Pictures/Screenshots")
os.makedirs(output_dir, exist_ok=True)

filepath = os.path.join(output_dir, filename)
driver.save_screenshot(filepath)

log_info(f"{filepath} saved")

send_measurement(0,0)
```

Buffering a sequence of surfaces and outputting them as an array

```
# specify the captured count when we send out the array
# it should be the same as the value of the parameter `Surface Count` in the Surface
Stitch tool
capturedCount = 3

surf = get_surface(0)

if 'captured' not in memory: # memory is type of dict
    memory['captured'] = [] # a list with fixed items

# if 'captured' not in memory or len(memory['captured']) != capturedCount: # memory is
type of dict
#     memory['captured'] = [None] * capturedCount # a list with fixed items

captured = memory['captured']

surf = get_surface(0)
captured.append(surf)

if len(captured) == capturedCount: # if the passed Captured value is equal to the item
count of the list
    offsets = []
    scales = []
    points = []
    intensities = []
    headers = []

    for s in captured:
```

```
        str = type(s)
        log_info(str)

        offsets.append(s.offset)
        scales.append(s.scale)
        points.append(s.points)
        intensities.append(s.intensity)

    send_surface(0, offsets, scales, points, intensity=intensities, header = surf.header,
intensity_format=surf.intensity_format)
    log_info('sending out a surface point cloud array')
    memory['captured'].clear()

    # don't clear the items to ease the configuration of downstream tools
    # captured.clear()
```

Down-sampling Surface Point Cloud Array

```
import numpy as np

# Downsample factor
FACTOR = 4

# Get the list of surfaces
surfaces = get_surface(0)

if surfaces is not None:
    # Initialize lists to hold the output batch data
    out_points = []
    out_intensities = []
    out_scales = []
    out_offsets = []
    out_adjacents = []
    intensity_format = None

    for surf in surfaces:
        # Downsample Points/Intensity
        # Use slicing [start::step] to keep every Nth pixel
        out_points.append(surf.points[::FACTOR, ::FACTOR])
        out_intensities.append(surf.intensity[::FACTOR, ::FACTOR])
        intensity_format = surf.intensity_format

        # Pass through scale, offset & is_adjacency
        out_scales.append(surf.scale)
        out_offsets.append(surf.offset)
        out_adjacents.append(surf.is_adjacent)

    # Send the surface point cloud array
    send_surface(
        0,
        offset=out_offsets,
```

```
        scale=out_scales,
        points=out_points,
        intensity=out_intensities,
        is_adjacent=out_adjacents,
        intensity_format=intensity_format
    )
```

Drawing Graphics

This script demonstrates the use of points, lines, regions, and labels.

```
# Calculates the average of two points and outputs coordinates
# as individual measurements and a combined Point feature.
# Inputs should be configured as:
#   Input 0 - Surface
#   Input 1 - Point 1 X
#   Input 2 - Point 1 Y
#   Input 3 - Point 2 X
#   Input 4 - Point 2 Y
# Outputs should be configured as:
#   Output 0 - Measurement (Midpoint X)
#   Output 1 - Measurement (Midpoint Y)
#   Output 2 - Point (Midpoint)

if any_input_invalid():
    send_all_invalid()
    return

# 1. Configuration: Visual Styles
POINT_SIZE   = 15
MID_SIZE     = 10

# 2. Data Extraction
x1, y1 = get_measurement(1).value, get_measurement(2).value
x2, y2 = get_measurement(3).value, get_measurement(4).value


# 3. Math: Calculate the Midpoint
mx = (x1 + x2) / 2.0
my = (y1 + y2) / 2.0
midpoint = (mx, my)

# 4. Visualization
# Draw a connection line between the two source points
draw_lines([(x1, y1), (x2, y2)], color=Colors.GRAY, width=1)

# Draw a region that perfectly bounds the two points
# Using abs() ensures valid positive dimensions for the region width and height
draw_region(midpoint, (abs(x2 - x1), abs(y2 - y1), 1), 0.0)

# Draw source points as white circles
# For the first point, make the color dependent on the decision Pass/Fail of the input
measurement
```

```
pointColor1 = Colors.RED
if get_measurement(1).decision == MeasurementDecision.PASS:
    pointColor1 = Colors.GREEN
draw_points((x1, y1), color=pointColor1, size=POINT_SIZE, shape=MarkerShape.CIRCLE)
draw_points((x2, y2), color=Colors.WHITE, size=POINT_SIZE, shape=MarkerShape.CIRCLE)

# Draw crosshair lines centered on the midpoint
# lineX spans the Y-distance; lineY spans the X-distance
# The lines are associated the X and Y tool outputs, so
# they are only drawn when that output is selected
lineX = [(mx, y1), (mx, y2)]
lineY = [(x1, my), (x2, my)]

draw_lines(lineX, color=Colors.CYAN, width=2, output=0)
draw_lines(lineY, color=Colors.CYAN, width=2, output=1)

# Label the midpoint with its coordinates
draw_label(f"Mid: {mx:.2f}, {my:.2f}", (mx, my))

# 5. Output Results
send_measurement(0, mx)
send_measurement(1, my)
send_point(2, midpoint)
```

Drawing green / red border around a Surface for Pass/Fail of a measurement

```
# --- GoPxL Surface Bounding Box Example ---
# Draws a red/green outline around the surface data extents
# based on the pass/fail decision of a specific measurement.
# Also outputs a string PASS/FAIL that can be pinned to keep
# the graphics visible.

if any_input_invalid():
    send_all_invalid()
    return

# 1. Configuration & Data Extraction
# For surface data:
surface = get_surface(0)
# For image data:
# image = get_image(0)
decision_meas = get_measurement(1)

# Offset and scale are also available on images
ox = surface.offset.x
oy = surface.offset.y

sx = surface.scale.x
sy = surface.scale.y

# Get dimensions from the points array shape (height, width)
height = surface.points.shape[0]
width = surface.points.shape[1]
# For images:
```

```
# image.pixels.shape[0]
# image.pixels.shape[1]

# 2. Math: Calculate Bounding Box Extents
min_x, min_y = ox, oy
max_x = ox + (width * sx)
max_y = oy + (height * sy)

# 3. Decision Logic
status_color = Colors.GREEN if decision_meas.decision == MeasurementDecision.PASS else
Colors.RED

# 4. Visualization
# Define the four corners of the surface area
outline_points = [
    (min_x, min_y), (max_x, min_y), # Top Edge
    (max_x, min_y), (max_x, max_y), # Right Edge
    (max_x, max_y), (min_x, max_y), # Bottom Edge
    (min_x, max_y), (min_x, min_y)  # Left Edge
]

# Draw the border and associate it with output 0 so it can be pinned
draw_lines(outline_points, color=status_color, width=5, output=0)

# 5. Output a string PASS/FAIL
send_string(0, "PASS" if  decision_meas.decision == MeasurementDecision.PASS else "FAIL")
```

Obtaining the versions of Python and the OpenCV and NumPy libraries

```
import sys
import numpy as np
import cv2

log_info(f"Python: {sys.version.split()[0]}")
log_info(f"NumPy:  {np.__version__}")
log_info(f"OpenCV: {cv2.__version__}")
```

Reading Detection Results (Image or Surface Inference)

This example reads the detection count and the per-detection class names and confidences, and outputs the highest confidence score as a measurement.

```
# Inputs (from the inference tool, in connected order):
#   0: Count (measurement)
#   1: Class Names (string array)
#   2: Confidences (measurement array)

count = get_measurement(0)
names = get_string(1)
confs = get_measurement(2)
```

```
if is_valid(count) and count.value > 0:
    best = 0.0
    for c in confs:
        if is_valid(c) and c.value > best:
            best = c.value
    send_measurement(0, best)   # highest confidence this frame
else:
    send_measurement(0, float('nan'))
```

Filtering by Class Name

Count detections whose class name matches a target and output that as a measurement.

```
TARGET = 'defect'

names = get_string(1)   # Class Names array
confs = get_measurement(2)   # Confidence Scores array

hits = 0
for n, c in zip(names, confs):
    if is_valid(n) and n.value == TARGET and is_valid(c):
        hits += 1

send_measurement(0, float(hits))   # number of 'defect' detections
```

Using Bounding Boxes (Image Inference)

Compute the area of the largest detected bounding box, in pixels.

```
TARGET = 'defect'

names = get_string(1)   # Class Names array
confs = get_measurement(2)   # Confidence Scores array

hits = 0
for n, c in zip(names, confs):
    if is_valid(n) and n.value == TARGET and is_valid(c):
        hits += 1

send_measurement(0, float(hits))   # number of 'defect' detections
```

Reading 3D Polygon Points (Surface Inference)

Read the left-boundary polyline of the first detected region and compute its average height (Z).

```
# Input 0: Polyline Left (array of 3D point features)
import numpy as np

points = get_point(0)   # array of point features

zs = [p.position.z for p in points if is_valid(p)]
if zs:
    send_measurement(0, float(np.mean(zs)))   # average Z of the boundary
else:
    send_measurement(0, float('nan'))
```

Surface Tools

Surface measurement involves capturing scan 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 different ways. This section describes some of the differences between these methods.

- For line profiler sensors, on the **Acquire** page, by enabling continuous Surface generation and configuring the part detection settings.
- By using the Surface Blob or Surface Segmentation measurement tools (for more information, see *Surface Blob* on page 760 and *Surface Segmentation* on page 951).
- For snapshot sensors (and G2, G4, or G5 sensors when Surface generation is *not* set to Continuous), by using the Surface Part Detection tool.
- For line profile sensors, by using the Profile Part Detection tool.

The table below lists several differences between the different methods. A key difference however is that Scan page 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 Surface Blob and Surface Segmentation, on the other hand, areas are not extracted as individual frames. They are extracted either as individual outputs or as elements in an array. (For more information on arrays, see *Arrays, Batching, and Aggregation* on page 199.) These tools also 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	Yes

	Part Detection	Surface Blob	Surface Segmentation
		control is more limited than with Surface Segmentation.	
Supports background present	Yes Height threshold must be set above or below the background	Yes Height threshold must be set above or below the background	Yes
Supports background with significant tilt or intensity gradient	No Fixed height threshold is used	No Fixed height threshold is used	Yes 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

Surface Align Ring

This tool is only intended for performing high-accuracy alignment of multi-sensor ring and partial ring layouts with G2 (line profile) sensors, using recorded scan data.

Surface Align Wide

This tool is only intended for trouble-shooting high-accuracy alignment of multi-sensor wide layouts with G2 (line profile) sensors, using recorded scan data.

Surface Anomaly Detector

> This tool requires a special dongle and license to function with live sensor data. (Available for purchase from LMI.) You must also install the dongle's supporting CodeMeter runtime to license the tool, which you do in GoPxL Manager. For more information, see *Installing Licensing Software* on page 1195.

> The Surface Anomaly Detector toolset only runs on GoMax ORIN / ORIN+ and on PC.
>
> For information on getting and uploading a GoMax firmware that provides this tool, see the GoMax ORIN / ORIN+ user manual.

> The Surface Anomaly Detector tool only works with uniform surface data. For more information, see *Scan Modes and Intensity* on page 139.

The GoPxL Anomaly Detector toolset, which is part of LMI's new GoPxL Pro Tools, lets you train and deploy AI-powered anomaly detection in parts and surfaces in scan data produced by a compatible Gocator sensor.

You can download the Pro Tools package for PC, which contains the toolset (`14658-x.x.x.x_GoPxL_Pro_Utilities.exe`), from the downloads area of the LMI Technologies website: https://lmi3d.com/product-downloads/. Run the installer, and provide a location where the content of the package will be unzipped.

The toolset includes the following:

- The Surface Anomaly Detector tool: This tool provides an AI-based model training routine and an inference engine that uses a trained model to predict whether something in production scan data is an anomaly. The training routine steps you through the process of training on "good" (OK) and "bad" (NG) example frames. The tool can also simulate part orientation and acquisition variations, as well as synthetic NG frames to improve model performance.
- The Training Archive tool: An auxiliary tool that conditionally saves frames to a project created in Surface Anomaly Detector, letting you improve the model on scan data acquired at a later time. For information on the Training Archive tool, see *Training Archive* on page 742.

The toolset supports both 2D intensity and 3D data. Surface Anomaly Detector runs on GoMax NX / ORIN / ORIN+ hardware accelerators or on PC. You must install a special firmware that includes the tool on your GoMax unit, and the on-sensor firmware version must match the GoMax firmware version.

Sample datasets are available in a package on LMI's Download page. For more information, see *Sample Files* on page 746.

If you are using this tool on PC, you install the CodeMeter runtime. For more information, see *Pro Tools* on page 113.

> The Anomaly Detector tool can be used with all Gocator sensors, except G1 sensors, G200 sensors, and G2300 revision A or B G2 models.

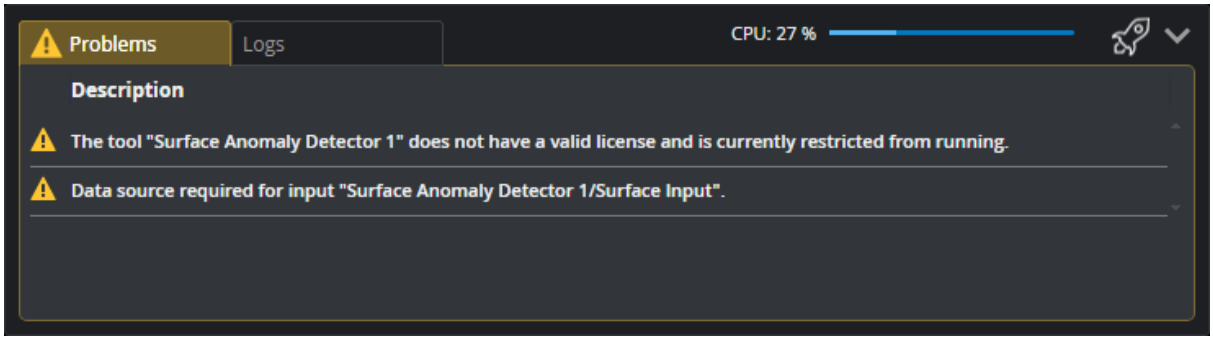
Hardware and Software Requirements

The following hardware and software is required to use the GoPxL Anomaly Detector in production:

1. One or more compatible Gocator 3D sensors: G2, G3, G4, or G5. Note that GoPxL can only accelerate a single G3 sensor.
2. A GoPxL Anomaly Detector license and dongle, inserted into a USB port on the GoMax unit or PC.



A license and dongle are required for running the tool on live scan data. Without the dongle, you can only evaluate the tool on previously scanned data. The tool will also display an error message without a dongle:



Plugging in the license dongle will license the tool. If the sensor system is running when you insert the dongle (that is, acquiring scan data), you must stop the system before it will license the tool.

3. A GoMax NX, GoMax ORIN, or GoMax ORIN+ accelerator, depending on scan resolution, cycle time and pre- or post-processing requirements.

   GoPxL Anomaly Detector can also run on a PC. To exceed the capabilities of GoMax ORIN in terms of training times and inference, an RTX 4070 or higher is recommended.If you choose to run this tool on a PC instance of GoPxL, the PC must have a graphics card with a recent Nvidia GPU that supports Nvidia GeForce Game Ready Driver version 576.57 or higher. To verify the driver version of your video card, launch the NVIDIA Control Panel, and in the main panel of the

application, make note of the driver version.



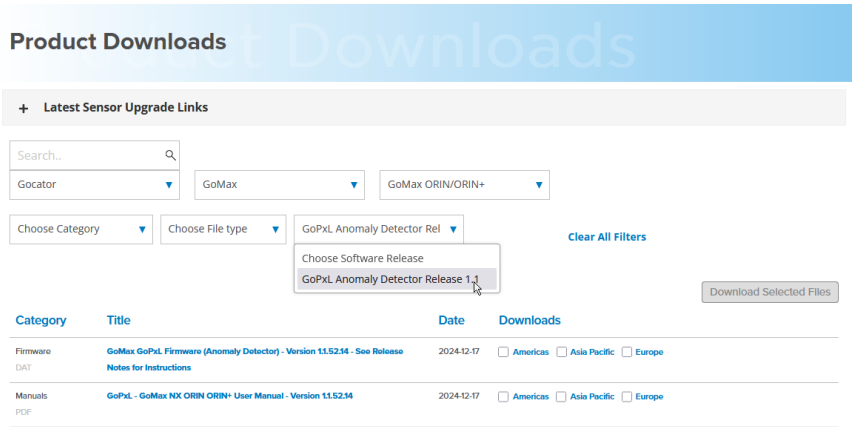
Note the following general requirements for running GoPxL on a PC:

- Processor: x86-64 (Core i3 or equivalent)
- RAM: 8 GB
- Hard drive: 64 GB
- Operating system: Windows 10 or 11 (64-bit only)
- Display: 1366 x 768

4. A special version of GoPxL that includes GoPxL Anomaly Detector for GoMax or for PC.

   You can find the appropriate version of GoPxL on LMI's Product Downloads page (https://lmi3d.com/product-downloads/).

   For GoMax, choose "Gocator" as the Product Brand, and then choose GoMax under Product Family, and your model under "Product Series." Under "Choose Software Release," choose a release that specifies Anomaly Detector. Scroll to find the firmware.



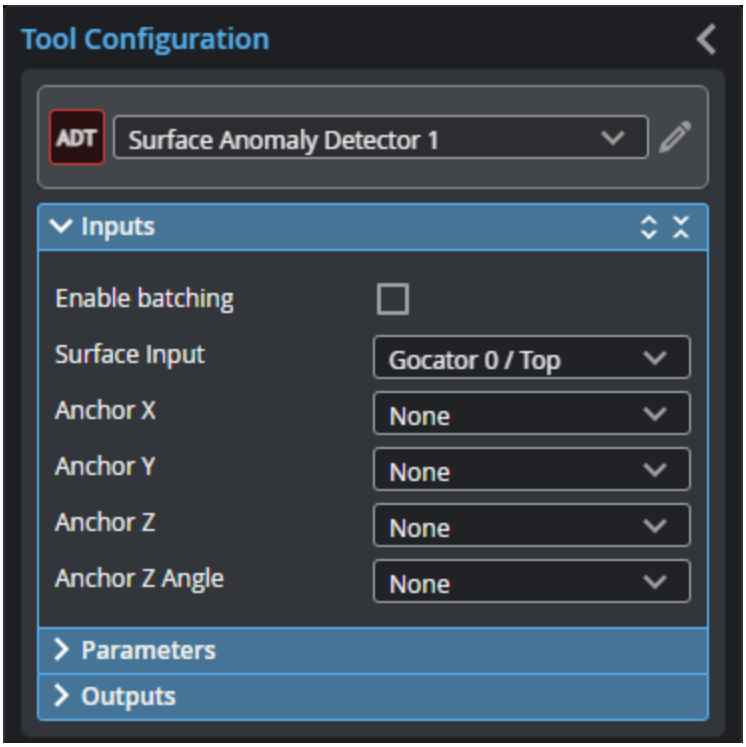
   For sensors and GoPxL for PC, under "Product Family," choose your the family of your sensor. Scroll to find the firmware and software, respectively.

5. If you are running GoPxL Anomaly Detector on a PC, a version of the CodeMeter runtime that is compatible with the dongle and license is required.

   You install the runtime in GoPxL Manager; for more information, see *Pro Tools* on page 113.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

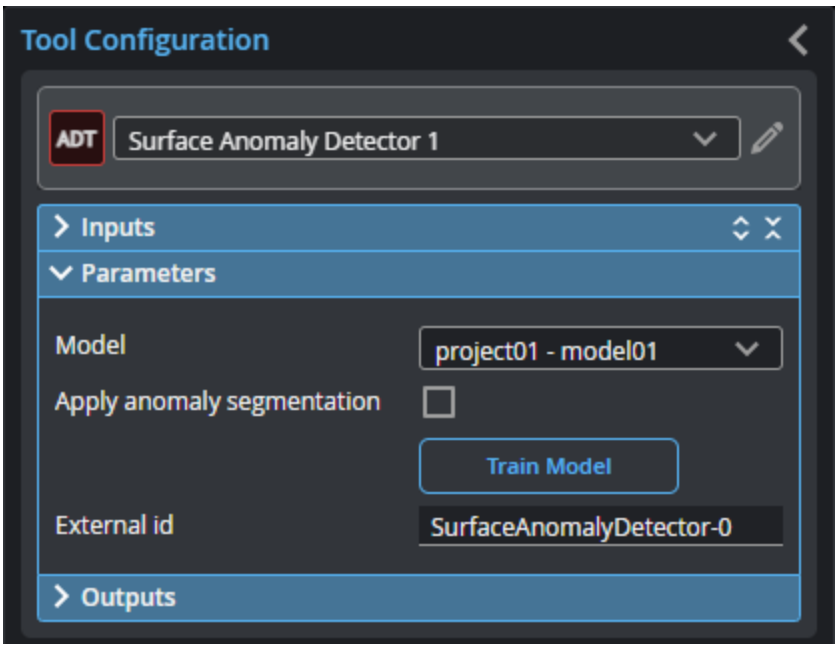
Inputs

Name	Description
Enable Batching	Batching (when a tool takes an array as input and processes each element in the array individually) is currently not fully supported by this tool. If you choose to use batching, keep the following limitations in mind: • The Label page only shows the first element of the input array. • For anomaly detection, blobs representing the anomalies are not displayed. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.

Name	Description
	This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X Anchor Y	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Model	Chooses the trained model the tool uses to detect anomalies during production.
Apply anomaly segmentation	When you set **Model** to a previously trained model, this parameter is available. When you enable this parameter, configure the parameters in **Segmentation parameters** (see below).
Segmentation parameters	The parameters that control anomaly output. **Threshold Offset** The offset that determines whether data is considered an anomaly. **Open kernel X**

Parameter	Description
	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 anomalies 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. **Use area filter** If **Use Area Filter** is enabled, the tool applies an area filter to anomalies using the values you provide in **Max area** and **Min area**. **Ordering** Orders the measurements, features, and surface data of the individual anomalies output by the tool. One of the following: • Position - X increasing • Position - X decreasing • Position - Y increasing • Position - Y decreasing • Area - Large to small • Area - Small to large
Train Model	Starts the model training routine, which steps you through process. For more information, see *Train Model* on the next page.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Development Workflow

The high level workflow is as follows:

1. Collect representative scans of the part you need to inspect in order to train an Anomaly Detector model. In addition to "good" parts, you need to scan "bad" parts, that is, ones with typical defects.
2. Create a pre-processing toolchain and train an Anomaly Detector model on GoMax.
3. Deploy the Anomaly Detector model and connect post-processing tools.

4. Optionally, collect new production data for model retraining on a GoMax.

Collect dataset

Unless you are using a previously created dataset, you must create the dataset that the GoPxL Anomaly Detector will use for training a model before it can detect anomalies in new production data. To do this, perform the following steps:

1. Record representative scans of 20 or more OK targets (targets with no flaws) and 20 or more NG targets (targets with flaws).

   You can record the scans to a single .gprec file or to multiple files. If you record scans to a single file, grouping each type (OK and NG) together will make it easier to label them later in the wizard. Make sure there are no empty frames: each frame should contain either an OK or an NG target. Do not rely on .gpsup or .rec for this as the Anomaly Detector wizard requires .gprec files when importing from PC.

2. Save a separate .gpsup file with the sensor configuration settings such as exposure time and related imaging parameters.

   Ensure the same parameters are used on the production system. Otherwise, you will need to collect new training data.

Implement a Preprocessing Toolchain

The preprocessing tool chain limits variation in the scan data to the region or surface of interest (with and without anomalies). With less variation in the scan data, the total number of frames needed for training is reduced.

The preprocessing tool chain must not change between training and inference. Otherwise, the system will not function properly. You must save your preprocessing toolchain in the .gpsup file created previously or using a separate .gpjob file.



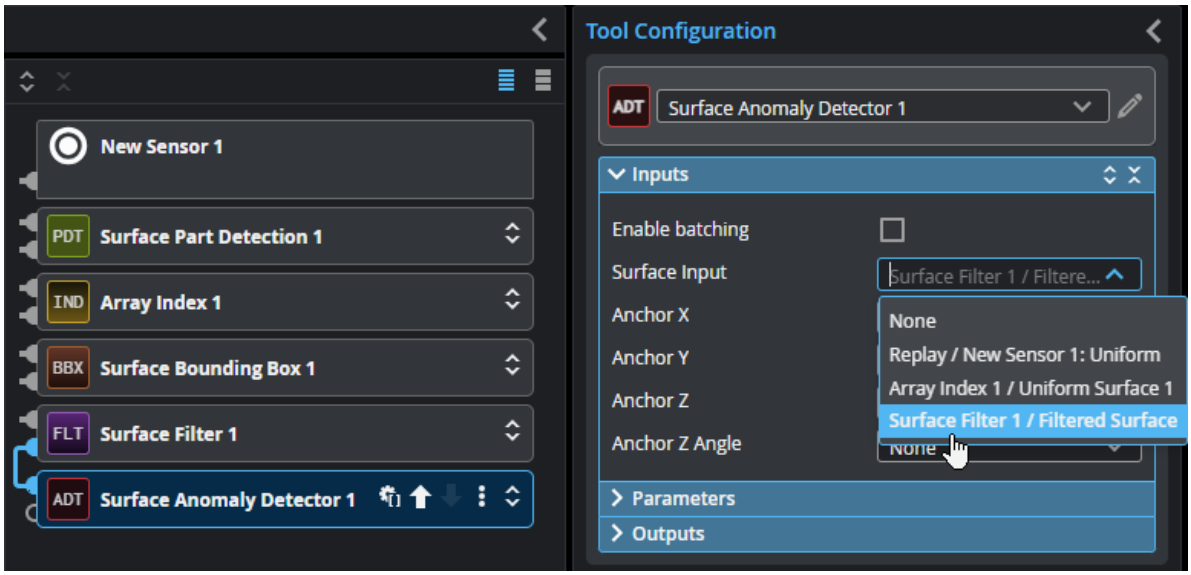
You *must* use the same preprocessing toolchain for training and inference to ensure proper detection of anomalies.

Train Model

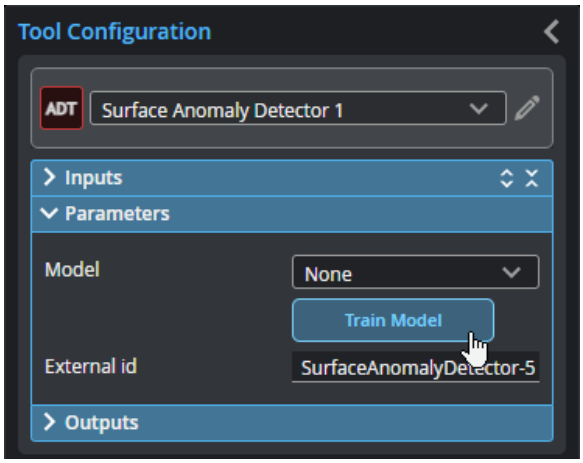
ADD SURFACE ANOMALY DETECTOR TOOL

1. Add and connect a Surface Anomaly Detector tool to the output of the preprocessing tool chain.

   In the following the first four tools in the Tool Diagram on the left are the preprocessing tool chain. The last tool in that tool chain, Surface Filter, is used as input for the Anomaly Detector tool.



2. In tool's **Parameters** section, click the **Train Model** button.

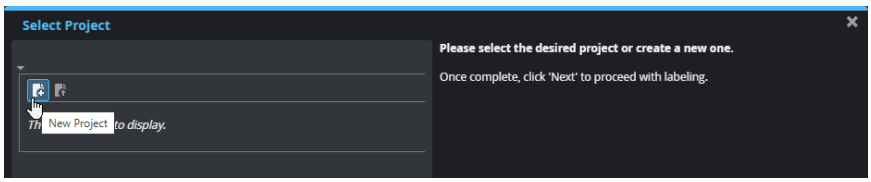


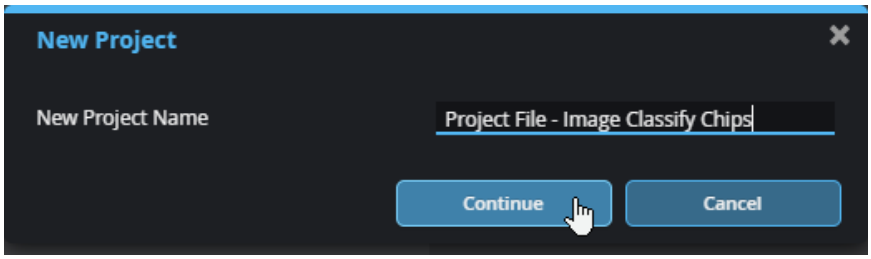
The routine starts and the **Select Project** page opens. You will create a new project in this page, or open a previously created project.

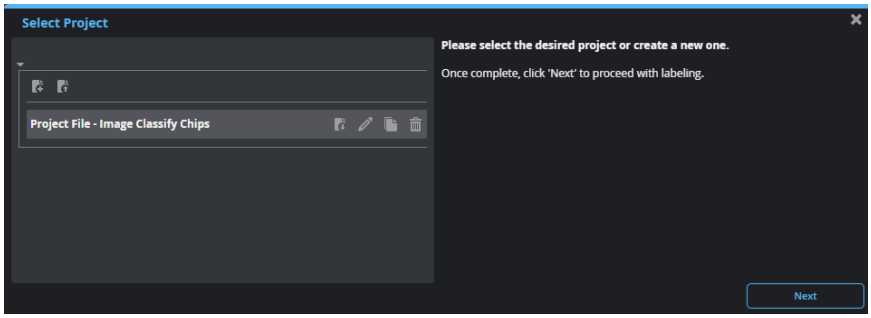
CREATE NEW OR OPEN PROJECT

A project is a collection of frames used for training, labels you apply to those frames (OK and NG), and the models trained on those frames. You should choose a meaningful project name, such as the part name, especially when working with multiple projects.

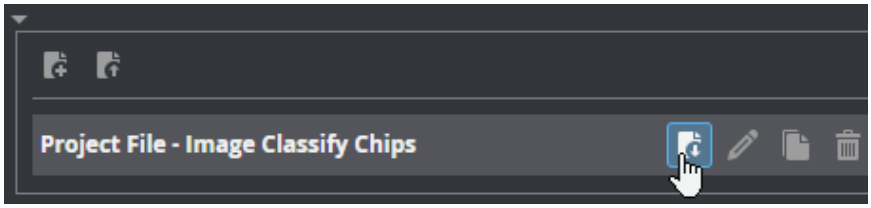
1. On the **Select Project** page, click the New Project icon, provide a project name, click **Continue**, and then click **Next**.



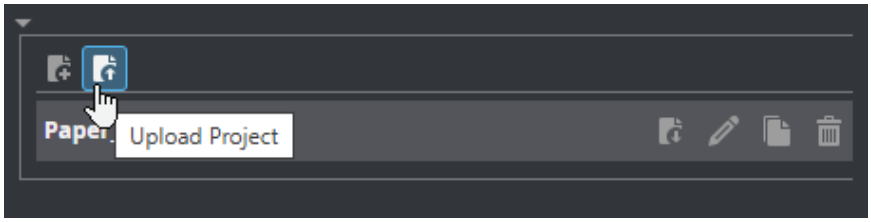




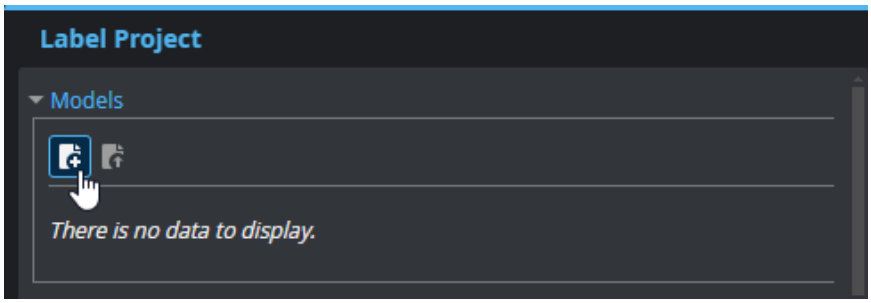
Use the download icon on a project's row to download this to your PC and upload to another device.

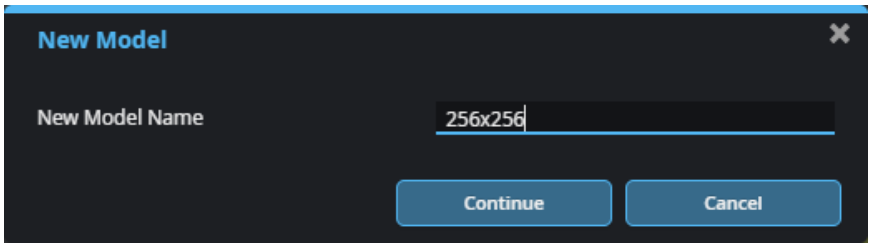


Use the upload icon to import a previously downloaded project.



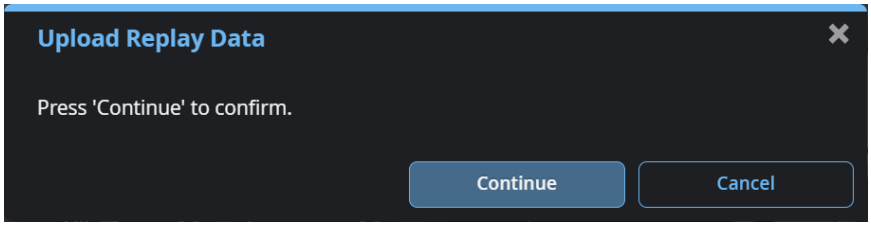
2. In the **Label Project** page, in the **Models** section, create a new model and click **Continue**.





3. In the **Frames** section, do one of the following

   - To upload currently loaded replay data, click the Upload Replay Data button ( ) , and then click **Continue**.



   - To upload the data from a .gprec file on the local PC, click the Upload From Disk button ( ), choose a file in the Open dialog a, click Open.

   The tool loads the frames into the model.

⚠	Loading replay data into the project can take several minutes. If a “Delayed Response” message displays, click “Wait.” In this case, add a Decimation Filter to the pipeline.

LABEL FRAMES AND CHOOSE SET

You must now label each frame as “OK” or “NG” (under the **User Label** column). You must also choose the set each frame belongs to (Train or Test, under the **Set** column).

1. (Optional) To have the system automatically place frames in the Train or Test set, click the Auto-split icon.

   The Set column will become read only. When you set a frame's user label, the tool assigns it to the sets in the proportions described below.



   Auto-split uses the proportions recommended for training to place each frame into a set:

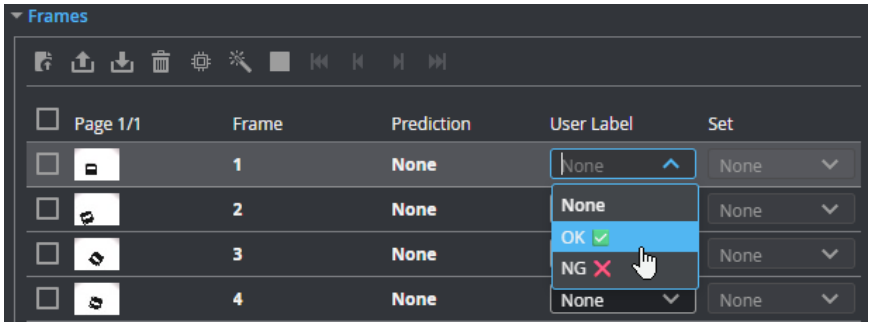
   80% of OK is in Train

   20% of OK is in Test

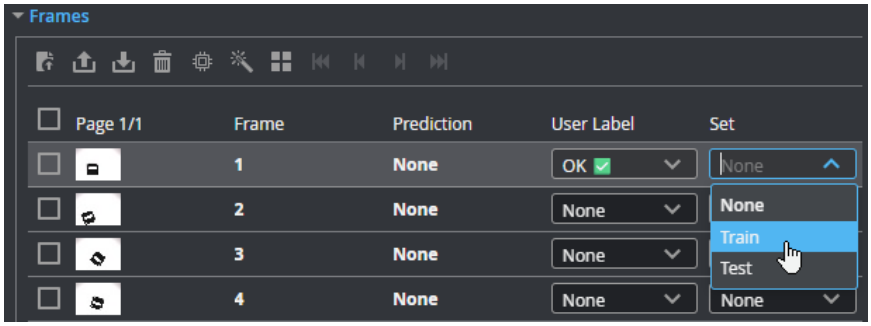
   100% of NG is in Test

   You can also enable auto-split after setting the user labels.

2. In the row of each frame, under **User Label**, choose the correct label.



If you have not enabled auto-split, also choose the frame's set under **Set**. Make sure you use the proportions described in step 1.



To quickly choose the label and set for multiple frames, do one of the following to first select the frames:

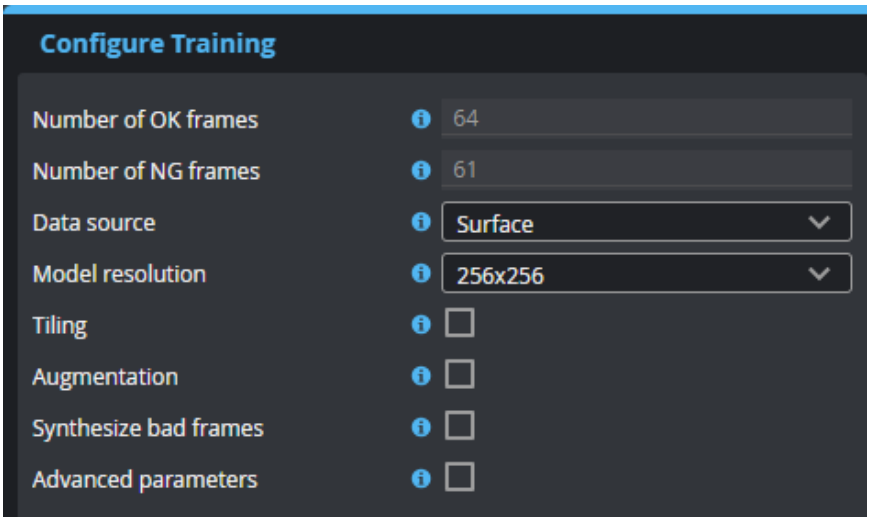
- To select a range of frames, click the checkbox of the first frame, press Shift, and click the checkbox of the last frame of the range.
- To select multiple frames, click the checkbox of the first frame, press Ctrl, and click the checkbox for each frame you want to select.
- To select non-contiguous ranges, alternate between clicking Ctrl and Shift.

After selecting the frames, choose the label and the set as described above. To deselect all frames, click anywhere between the checkbox and the User Label and Set drop-down menus.

3. Click **Train Model** to configure training.

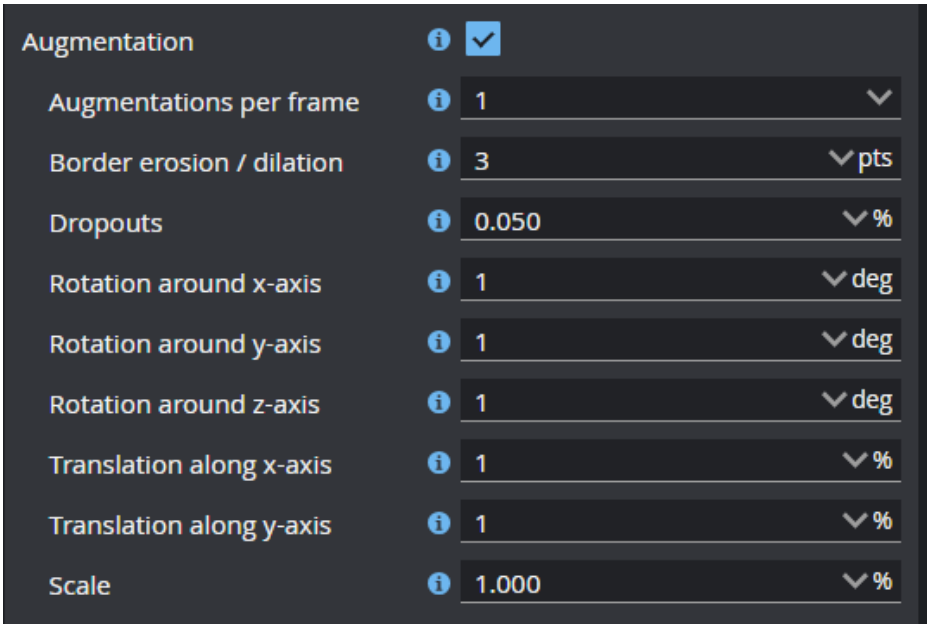
CONFIGURE TRAINING

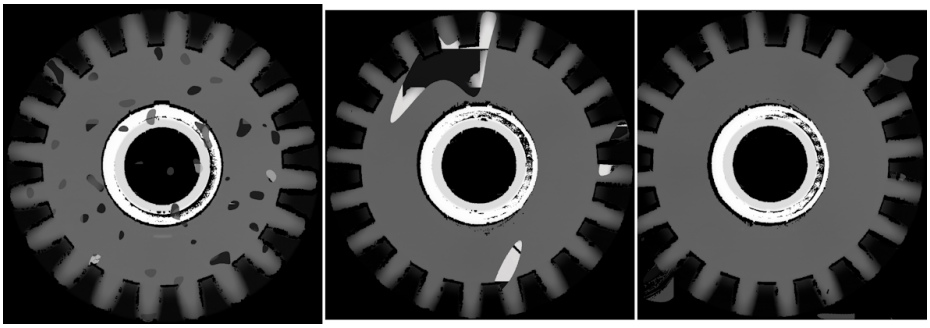
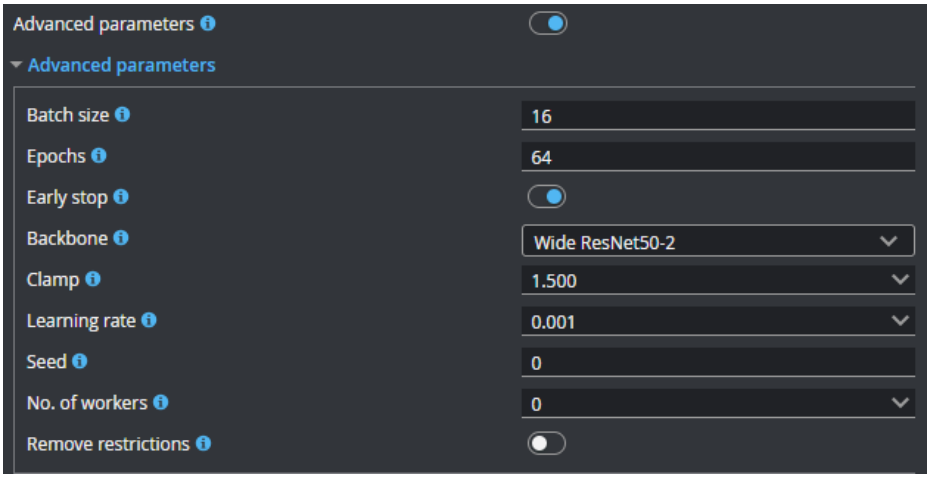
After labeling the frames and choosing the set they belong to, you must configure training. For descriptions of the settings, see below.



Training Parameters

Parameter	Description
Number of OK images Number of NG images	Read-only fields that display the number of frames you have labeled OK and NG, respectively.
Data source	The source the tool uses during training: Surface or, if available, Intensity data. You use Surface data for defects detectable on a 3D surface (that is, using height data). You use Intensity data for defects on flat surfaces, such as printing flaws. For more information on scan modes, see *Scan Modes and Intensity* on page 139.
Model resolution	Decimates the input data to the specified resolution for processing. A higher resolution improves anomaly location accuracy but increases processing time for training. For most applications, leave this at 256x256.
Tiling	Enables tiling. Tiling divides the frames of training data into sections. The tool uses the same model for each tile, which increases the resolution of the model. This lets the tool potentially find smaller defects or increase detection accuracy. You should only use tiling with homogenous targets (for example, paper towel, wood decking, and so on) or symmetric targets. For discrete parts with a complex or asymmetric shape, disable this setting. After enabling tiling, set the number of vertical and horizontal tiles. You can also set tiles to overlap in **Tile margin**. Use this to detect anomalies that might fall on a boundary between tiles. Generally, you should use a value between 12.5% and 25%. Note that using tiling and tiling margins increases the memory requirements for training considerably. For example, with 5x5 tiling, the tool uses 25 times more memory.
Augmentation	When **Augmentation** is enabled, the tool simulates random part orientation, size, and acquisition variations to generate more training data, which improves training. The rotation, translation, and scale values are used to create a range. So for example, a Z axis rotation value of 1 degree results in a range between -1 and 1,

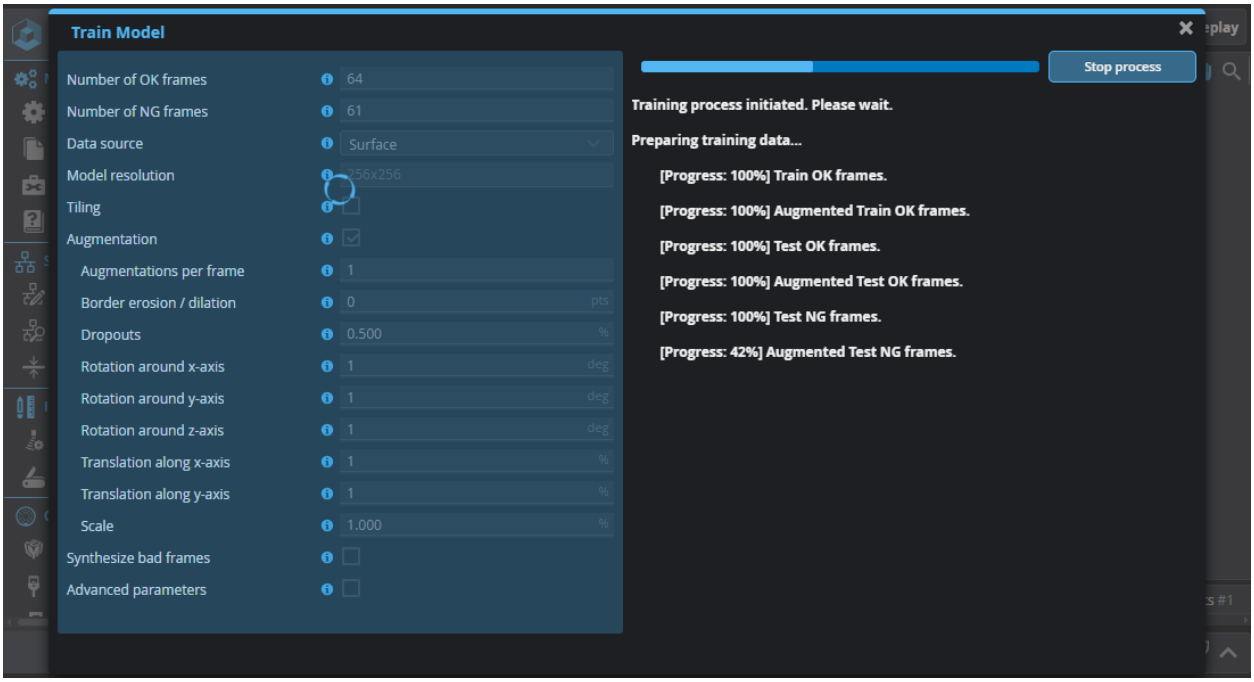
Parameter	Description
	from which the tool chooses a random value to generate data. For **Dropouts**, the value represents the upper limit of the range between 0 and the provided percentage. Choose augmentations that are representative of the variations you encounter during production. Augmentations that aren't representative may lead to poor results. You can visualize example augmentations on the Model Performance page, which is available after training is complete; for more information, see *Test Model Performance* on page 732.  **Augmentations per frame** The number of augmentations generated for each frame in the project (OK and NG). Augmentations are not added to the total number of frames in the project itself. **Border erosion / dilation** Performs erosion and dilation operations on each frame. Set to 1 or higher for discrete parts, and set to 0 for continuous surfaces or when using a pre-processing pipeline that crops the input surface. **Dropouts** Removes data points randomly to simulate low exposure, up to the specified percentage of the original scan data. Set this to 0 if you are using gap filling or don't expect exposure issues.

Parameter	Description
	Rotation settings The Rotation settings limit rotation of simulated data around the X, Y, and Z axis. **Translation** settings The Translation settings limit movement of simulated data on the XY plane, as a percentage of the data points in the frame. **Scale** Limits the scaling of simulated data.
Synthesize bad frames	When this setting is enabled, the tool synthesizes bad (NG) frames from the good (OK) frames in the project. The frames are used internally during training and are not added to the frames in the project or shown in the tool. The following are examples of synthesized frames:  Enable this setting if your project contains very few NG frames. The setting is automatically enabled if the project contains *no* NG frames.
Advanced parameters	If the training results are generally good but the tool is reporting a small number of false positives or false negatives, you can modify these parameters to further improve results. 

Parameter	Description
	Batch size The number of frames in a batch (a subset of the entire dataset passed to the training algorithm for an iteration). If you are using a model with a 512x512 resolution or tiling and the tool has memory issues during training, decrease this value. **Epochs** The number of times the entire dataset passes through the training algorithm. To reduce false positives or negatives, increase this value. Note that training will take longer. **Early stop** Leave enabled for the initial training runs. To help model performance, disable to improve model performance; note that training will take longer. **Backbone** The feature extractor used by the tool. ResNet18: A lighter model that is fast and a good starting point. Wide ResNet50-2: Slow and larger memory, with better for complex textures. **Clamp** The range within which the data values should be clamped. Reduce this value to help reduce false positives or false negatives. **Learning rate** Set this to a higher value to reduce training time. Note that a value that is too high can cause the model to miss the point where the difference between predicted and actual OK / NG is at its smallest (that is, overshooting it). A lower value causes the model to learn more slowly, but potentially more accurately; if the value is too low, the model can get stuck. **Anomaly size** One of the following: Small and big, Big, Small. Choose Big to reduce false positives caused by smaller features like reflections or drop-outs. Choose Small to ignore large features in the data. For most applications, leave this as Big and small.

Parameter	Description
	Remove restrictions By default, this option is disabled, and the following limitations are in place to ensure the system trains successfully and does not run out of memory: • The maximum number of tiles under the **Tiling** parameter is limited to 6. • The highest resolution in **Model resolution** is 512x512.

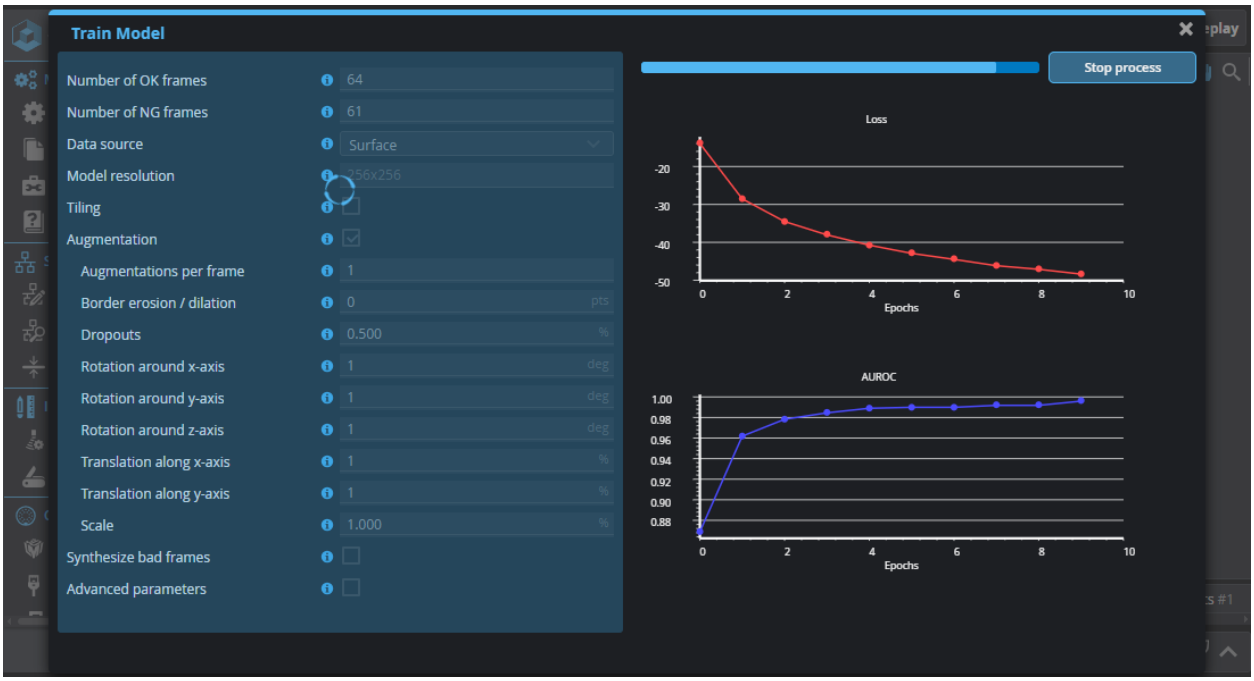
After you configure training, click **Start Training** to train the model. The tool will start by preparing the training data, providing progress feedback:



After the data is prepared, the tool will start generating Loss and AUROC graphs.

Loss: This graph represents the cumulative error for each frame in the Test set, determined after each training run iteration or epoch. A decreasing number (the graph's line is moving down) implies the model is improving.

AUROC: This graph defines how well the model can distinguish between the OK and NG user labels. A value of 1 (the graph's line is moving up) represents the best possible result.



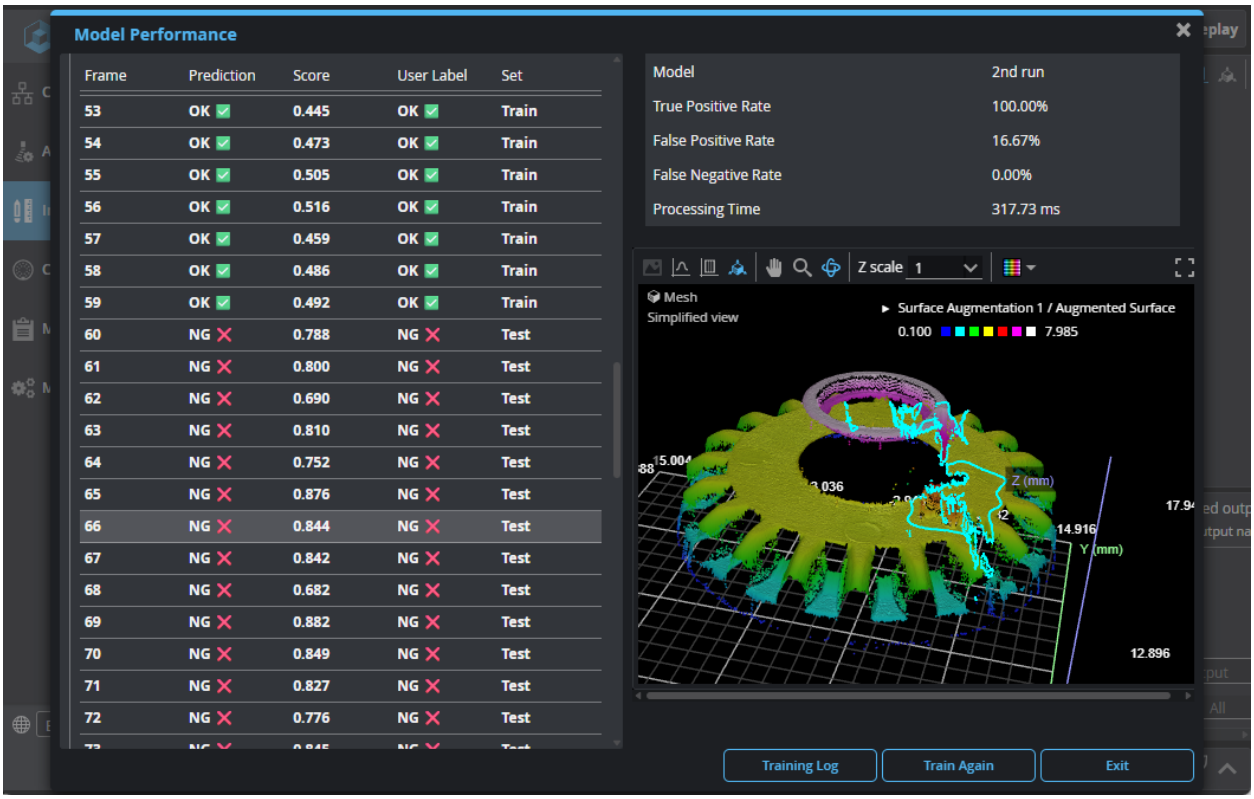
As the tool trains the model on the labeled frames, the graphs should converge, that is, Loss will decrease, and AUROC will increase, approaching 1 (the best possible result). After the training has completed, click **Next** to test the model. Note that training can take several minutes.

TEST MODEL PERFORMANCE

After the tool has finished training the model, it will use the model to make predictions on the frames to determine the anomaly score for each frame, as well as the anomaly threshold. The Model Performance page displays the following results:

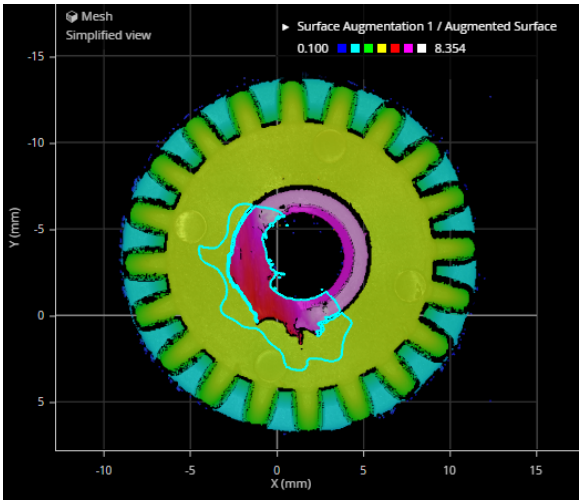
Performance Results

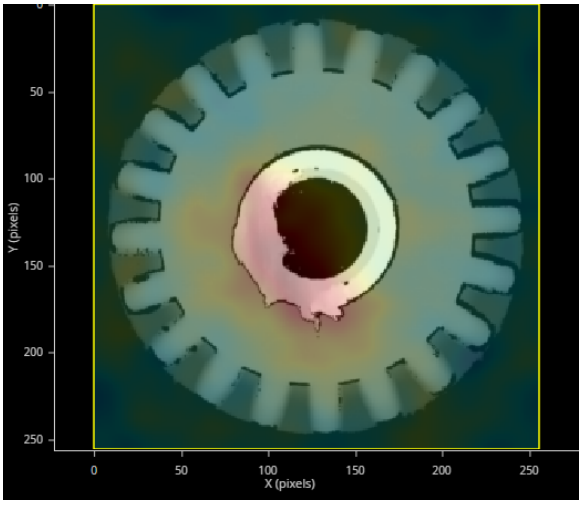
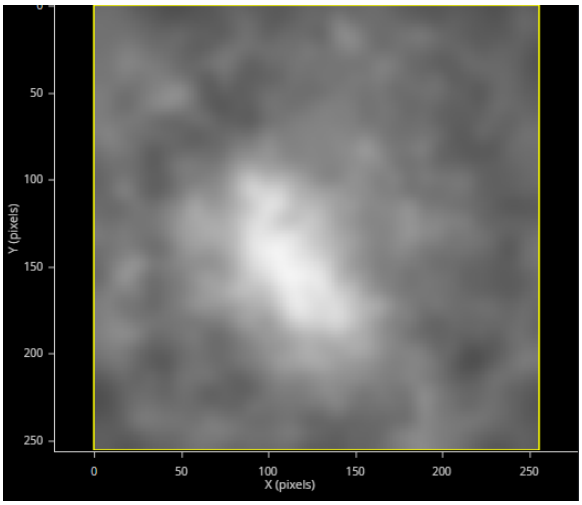
Name	Description
Model	The trained model the tool used to detect anomalies during the model performance phase.
True Positive Rate	The percentage of frames correctly predicted as OK.
False Positive Rate	The percentage of frames incorrectly predicted as OK.
False Negative Rate	The percentage of frames incorrectly predicted as NG.
Processing Time	The average time required for inference. Includes the pre-processing tool chain.



You can use the **Anomaly visualization** drop-down to view different visualizations of the selected frame:

Anomaly visualizations

Name	Description
Anomaly Outline	Displays the Surface data of the frame with anomalies outlined, if present. 
Heatmap	A diagnostic Image output overlaying the Anomaly Map output on the input data. This output can make it easier to see whether the brighter areas in the anomaly

Name	Description
	map correspond with flaws in the target. 
Anomaly Map	A diagnostic Image output that contains grayscale values representing a per-pixel score of how anomalous that pixel is. Brighter areas should correspond to flaws. When Anomaly Map is passed to segmentation tools such as Surface Blob, the data can be used to locate and measure the size of the anomalies. 

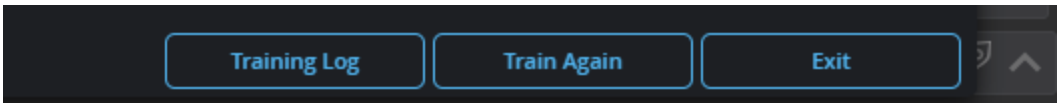
If you have enabled augmentations in the training configuration, you can use the **Show augmentations** drop-down to switch between a frame's original scan data and a simulated frame to get an idea of the effect of the augmentation settings. Note that the augmentations the tool shows are not the ones used in training, which are generated randomly during training using the provided ranges. Also, the score displayed in the selected row is the original frame's score, not the augmented frame. The following options are available:

Augmentations

Name	Description
Original	The frame's original scan data.
Upper Limit Lower Limit	These options display a frame with maximum and minimum augmentation values applied to the scan data, respectively. Note that the maximum / minimum values are determined by the settings in the Configure Training step of the routine. For example, if you have set the Z axis rotation to 1 degree, Upper Limit will show the frame rotated 1 degree clockwise. Lower Limit will show the frame rotated 1 degree counter clockwise. For **Border erosion / dilation**, Upper Limit and Lower Limit show the effect of dilation and erosion operations, respectively, using the provided kernel size. If you have enabled dropouts (by setting the **Dropouts** slider in the training configuration to a non-zero value), with Upper Limit, the tool removes a random selection of data points using the percentage provided. With Lower Limit (which is equivalent to 0% in the setting), no data points are removed.
Random	The tool applies random values chosen from the augmentation ranges provided during the Configure Training step.

The **Anomaly threshold mode** setting lets you choose whether to favor OK or NG frames, or to use a balanced approach. You can also set a custom value.

The Model Performance page also lets you download a tar archive containing reports and other diagnostics. You typically only need to do this if asked to do so by LMI Support.



You can also retrain the model with different training configuration if you wish.

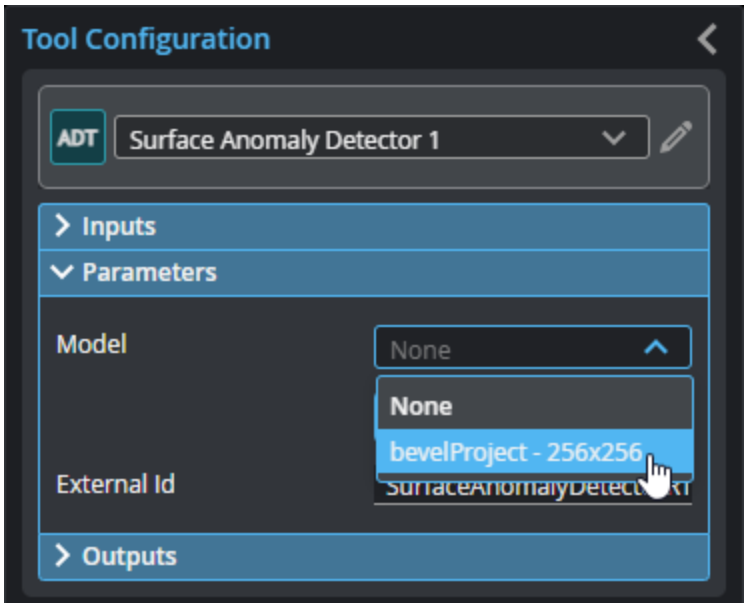
You are now ready to use the trained model on new scan data in production.

Deploy in production

ANOMALY TOOL SETUP

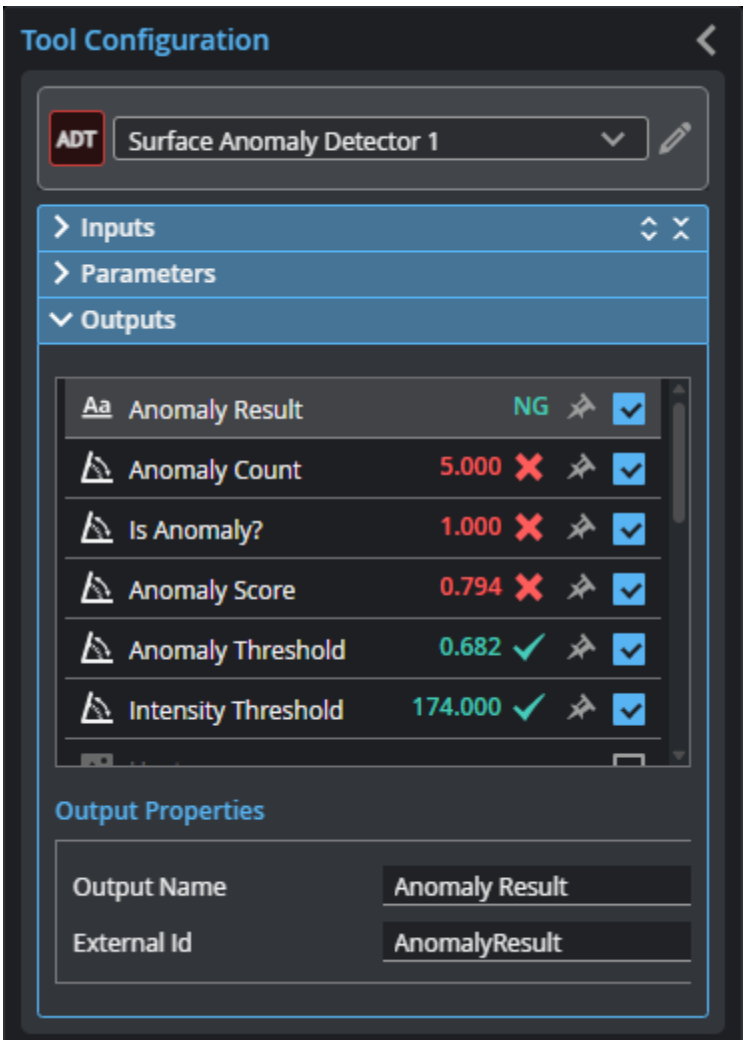
1. Connect to the GoMax using a laptop or PC on the same network.
2. Upload a .gpsup or .gprec file created during the dataset collection.

   Ensure the recording contains the same preprocessing pipeline used during training.
3. Discover the Gocator sensor and create a new sensor group.
4. Insert the GoPxL Anomaly Detector USB dongle to a USB port on the GoMax.
5. In the Surface Anomaly Detector tool, in **Model**, choose the model you trained and want to use in production.



6. Configure the outputs as required, and implement post-processing to isolate and prepare anomalies for measurement. For more information, see *Outputs* on the next page and *Surface Anomaly Detector* on page 716.

Outputs



Although the tool has several outputs, the "Is Anomaly" measurement and "Anomaly Surface" output are the main outputs intended for processing by tools later in the tool chain.

To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Some outputs are only displayed if **Apply anomaly segmentation** is enabled in the **Parameters** section.

Measurements

Measurement
Anomaly Result A string representing the result: OK or NG, corresponding to 0 and 1, respectively, in the "Is Anomaly?" measurement.
Anomaly Count The number of anomalies detected in the frame.
Is Anomaly? Indicates whether the tool has detected an anomaly in the current frame of scan data.
Anomaly Score Indicates how strongly the tool believes the current frame contains an anomaly. When the value is higher than the Anomaly Threshold measurement, this measurement is set to 1.0.
Anomaly Threshold Threshold value (0..1) of the 'Anomaly Score' above which the model considers an input surface anomalous. This value is determined during training to achieve the best possible detection rate.
Intensity Threshold A grayscale value in the 'Anomaly Surface' intensity data above which the model considers input intensity data anomalous. The value is constant for a trained model. The value is equivalent to 'Anomaly Threshold' x 255 and rounded to the nearest integer.
Center X **Center Y** Arrays of the X and Y positions of the centers of mass of the anomalies in the frame.
Average Z An array of the average Z positions of the anomalies in the frame.
Min Z **Max Z** Arrays of the minimum and maximum Z positions of the anomalies in the frame.
Width **Length** Arrays of the length and width of the rotated bounding boxes that encapsulate the anomalies in the frame. These are always the major and minor axis of a blob, respectively.
Area An array of the areas of the anomalies in the frame. 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 1014.

Data

Type	Description
Anomaly Surface	Contains the Surface data of the frame, with the anomaly map (see Anomaly Map below) added as intensity data. The intensity data represents a per-pixel score of how anomalous that pixel is.
Heatmap	Diagnostic Image output overlaying the Anomaly Map output on the input scan data. This output can make it easier to see whether the brighter areas in the anomaly map correspond with flaws in the target.
Blob Surface	An array containing the frame's anomalies. Pass these to other tools for measurement or further processing.

Store production data

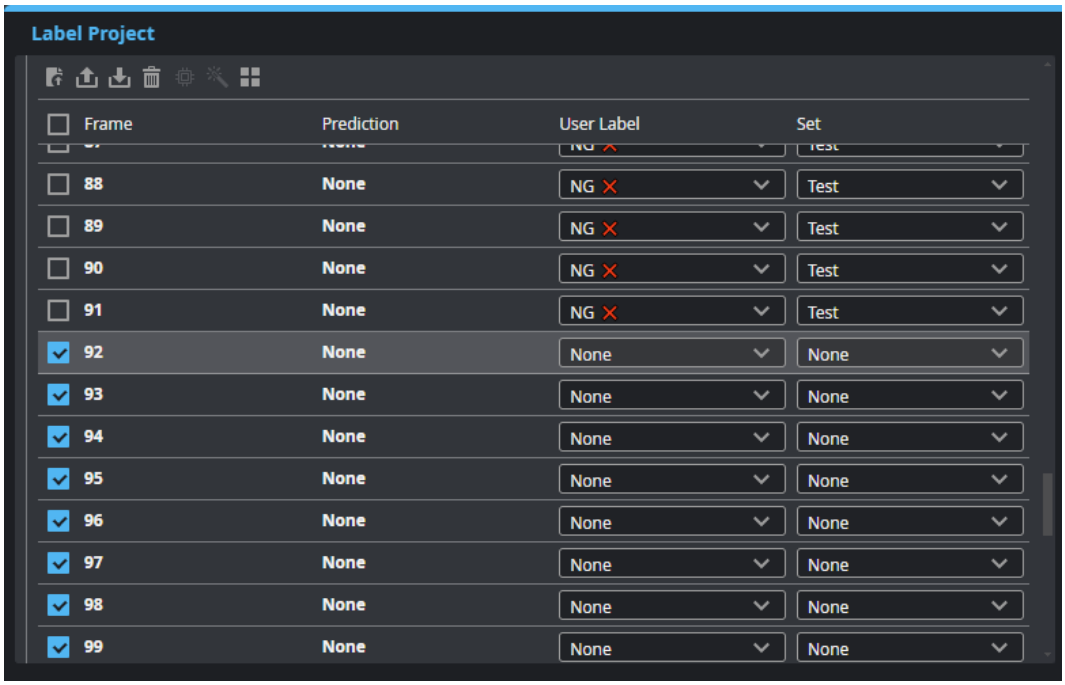
You can use the Training Archive tool to conditionally add frames to a Surface Anomaly Detector project. For more information, see *Training Archive* on page 742.

Train using additional data

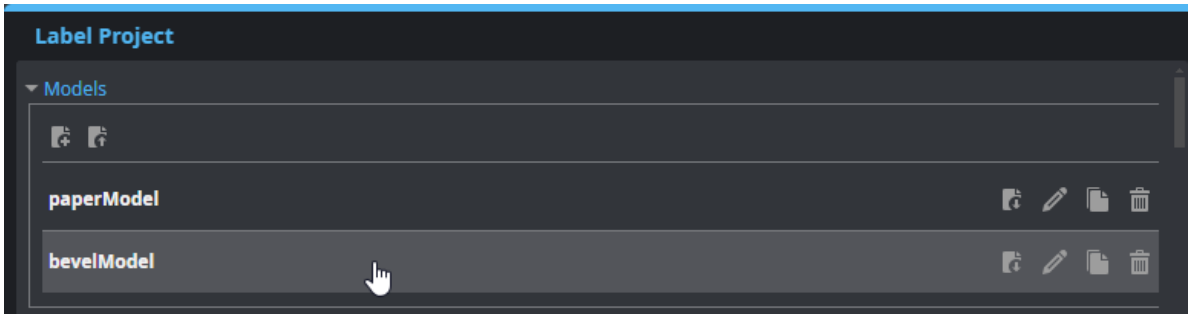
After adding new frames to a project using the Training Archive tool (see *Training Archive* on page 742) or uploading additional frames manually (from the replay buffer or from a .gprec file), you can further train the model with new frames, taking advantage of the trained model's prediction capabilities to automatically label frames as OK or NG.

To train using new data:

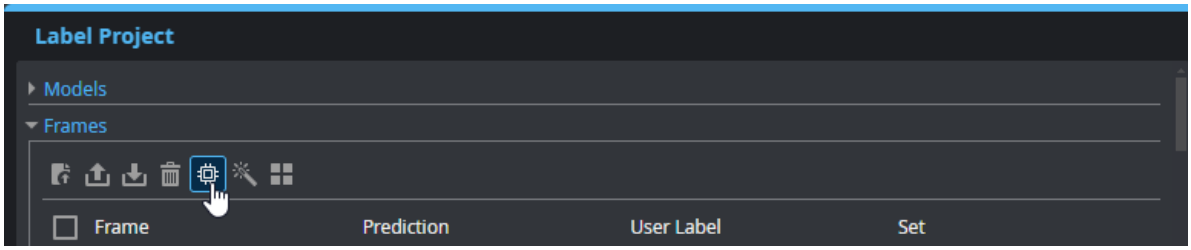
1. After adding frames to a project, go to the Label Project page.
2. Select the new frames.



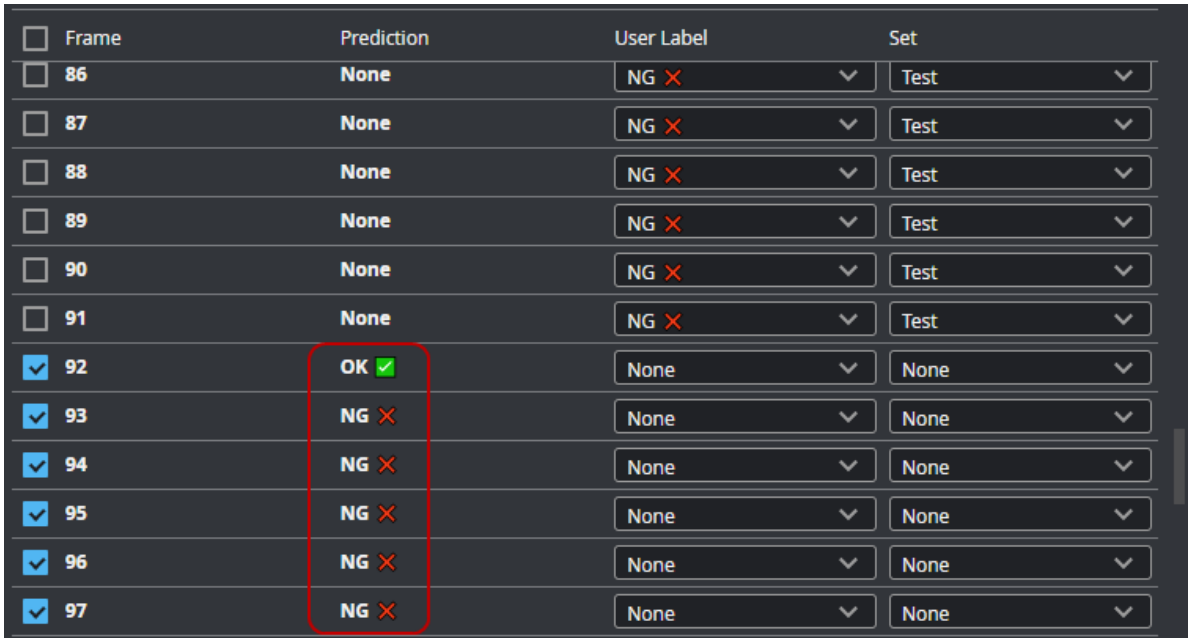
3. Choose the correct model you want to use for prediction.



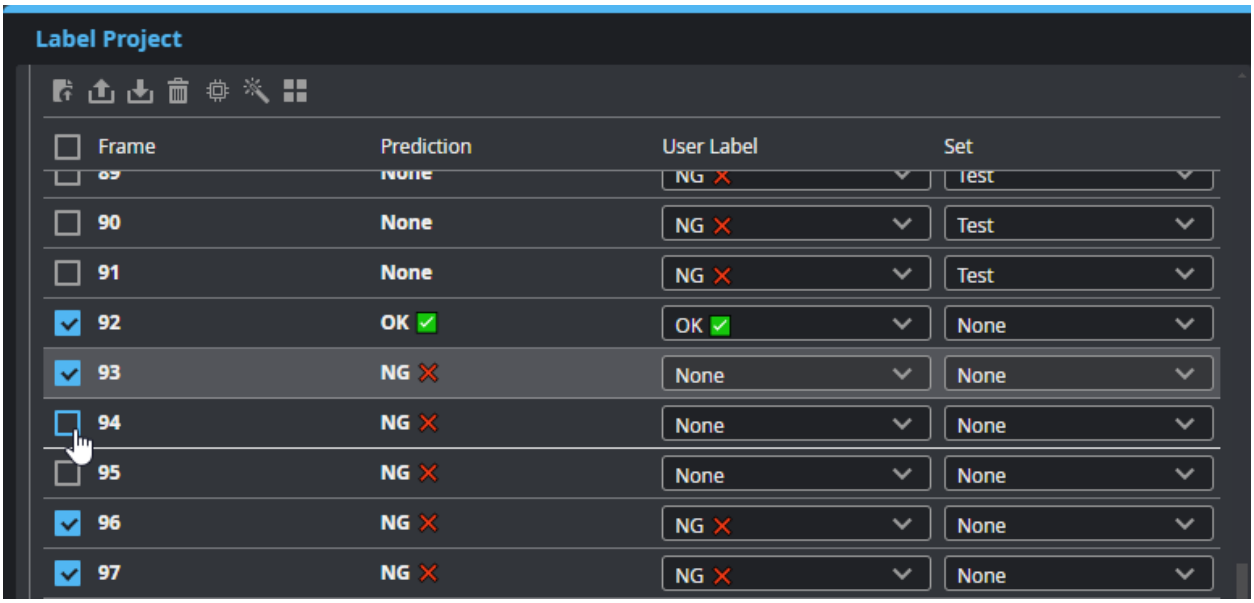
4. At the top of the Frames section, click the Generate Prediction button.



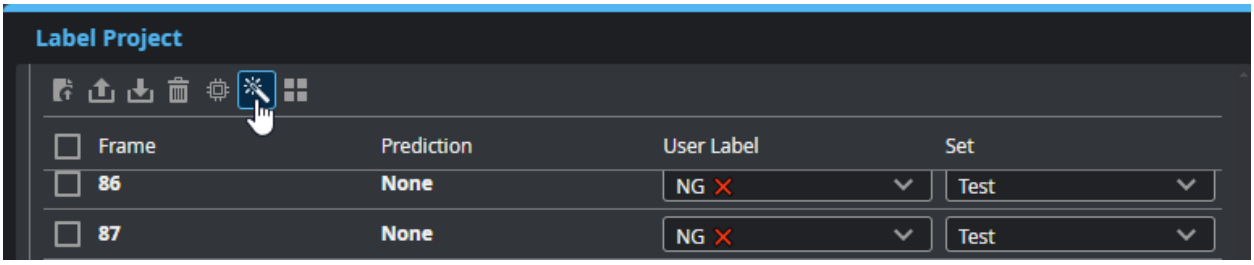
The tool uses the selected model to generate predictions (OK or NG) for the selected, unlabeled frames. Note that the user labels of the frames are unchanged at this point.



5. Review the predictions and do the following:
   a. Uncheck any invalid predictions.



b. Click the Apply Prediction button.



c. Toggle Auto-Split to On to automatically set the Set of the selected frames.

d. Manually set the user label of any incorrectly predicted frames.

The predictions of the selected frames are copied to the user label drop-down.

6. Retrain the model with the newly added frames.

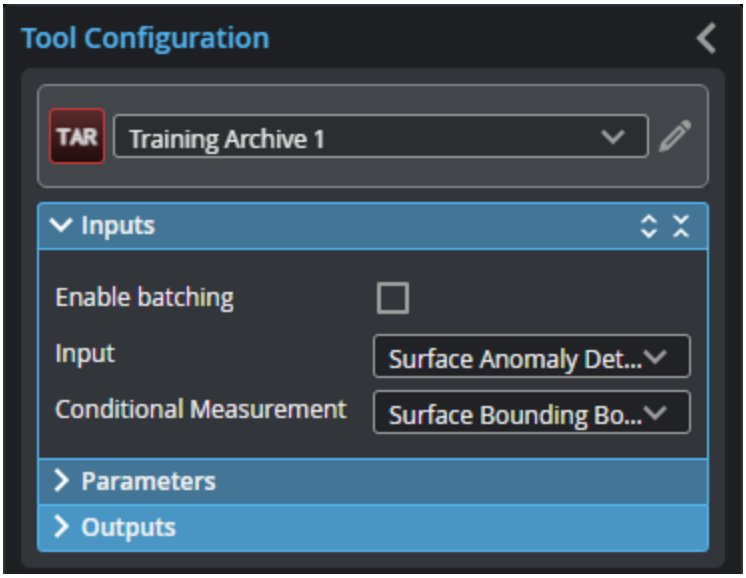
For more information, see *Train Model* on page 722.

Training Archive

The Training Archive tool lets you conditionally add frames to an Anomaly Detector project. (For more information on the Surface Anomaly Detector tool, see *Surface Anomaly Detector* on page 716.)

INPUTS

You configure the tool's inputs in the expandable **Inputs** section.



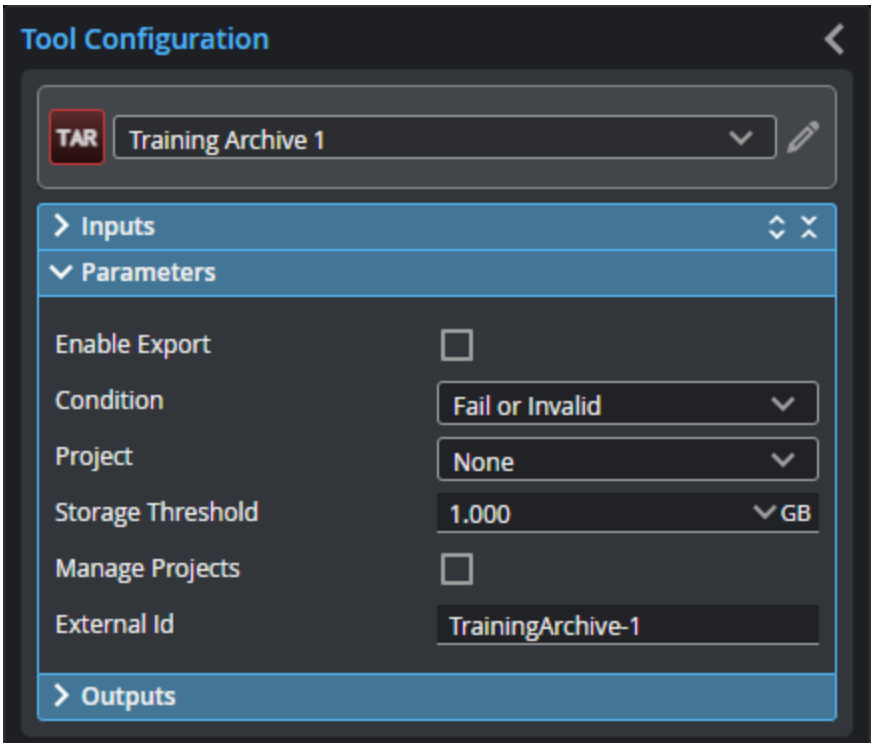
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see Arrays, Batching, and Aggregation.
Conditional Measurement	The measurement the tool uses as a condition to add frames to a project. Some examples of measurements typically used are the "Is Anomaly" output of the Surface Anomaly Detector tool, the Result measurement of the Measurement Formula tool, or an "Output" output from a Script tool.

PARAMETERS

You configure the tool's parameters in the expandable **Parameters** section.



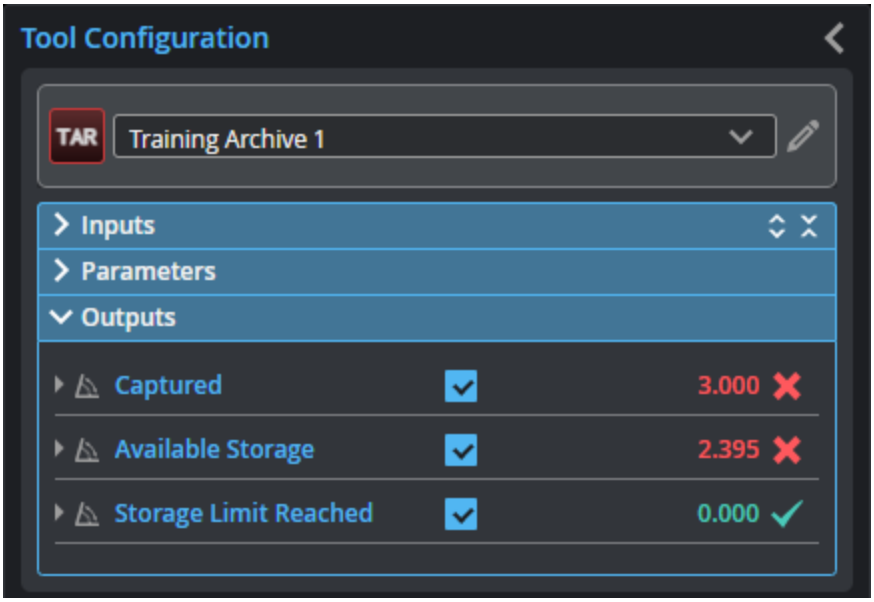
Parameters

Parameter	Description
Enable Export	When this setting is checked, if the conditions are met (**Condition** and **Condition Threshold** parameters) the tool will add the current frame to the selected project.
Condition	Displayed if the **Conditional Measurement** input is set to a measurement. Set this to the type of condition (with respect to the measurement set in **Conditional Measurement**) the tool should use to determine whether to add a frame to the selected project. The frame is added based on the following options: **Fail or Invalid**: If the selected measurement returns a Fail or Invalid result. **Invalid**: If the selected measurement returns an Invalid result. **Pass**: If the selected measurement returns a Pass result. **Less or Equal to**: If the selected measurement less than or equal to the value provided in **Condition Threshold**.

Parameter	Description
	Greater or Equal to: If the selected measurement greater than or equal to the value provided in **Condition Threshold**.
Condition Threshold	The value the tool compares with the measurement set in **Conditional Measurement** to determine whether to add a frame to the selected project. This parameter is only displayed when **Condition** is set to "Less or Equal to" or "Greater or Equal to".
Project	The project the tool will add frames to if the selected conditions are met.
Storage Threshold	The minimum amount of storage allowed on the GoMax unit or the PC. If the storage goes below this value, Training Archive will stop adding frames to the selected project, and the Storage Limit Reached output will be set to 1. Use the Available Storage output to see the current amount of storage on the device.
Manage Projects	When you enable **Manage Projects**, in the **Action** drop-down, you can delete the project currently selected in **Project**, or delete *all* of the projects on the device.
Action	⚠ You are not prompted to confirm whether to delete projects!
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

OUTPUTS

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	
Captured	The number of frames added to the project selected in **Project** so far.
Available Storage	The amount of storage available on the GoMax or PC.
Storage Limit Reached	When the available storage goes below the value in the **Storage Threshold** parameter, this output returns 1.

Sample Files

Sample files are available in a package on LMI's Download page (https://lmi3d.com/product-downloads/) to help you better understand the Surface Anomaly Detector tool and the required workflow. The sample data includes scans of Lego bevel gears, paper towel with debris, and tires with bulges. For each sample type, the following is provided:

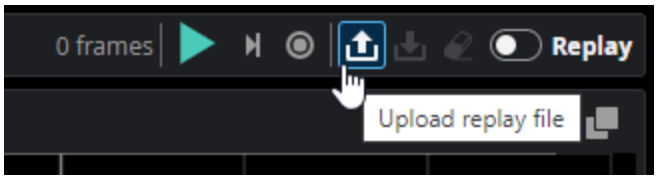
- A `.gprec` file containing a small set of "bad" (NG) frames of previously recorded scans. You can use these to test inference on a trained model. These files also contain a pre-processing tool chain appropriate for the sample, both for the training phase and the inference phase, and a Surface Anomaly Detector tool. You should load these first.
- An Anomaly Detector project file that contains previously scanned and labeled OK and NG frames, as well as a trained model. You can use the project files to try out configuring and training a new model without needing to acquire data. Note that loading a project file can take a few minutes.

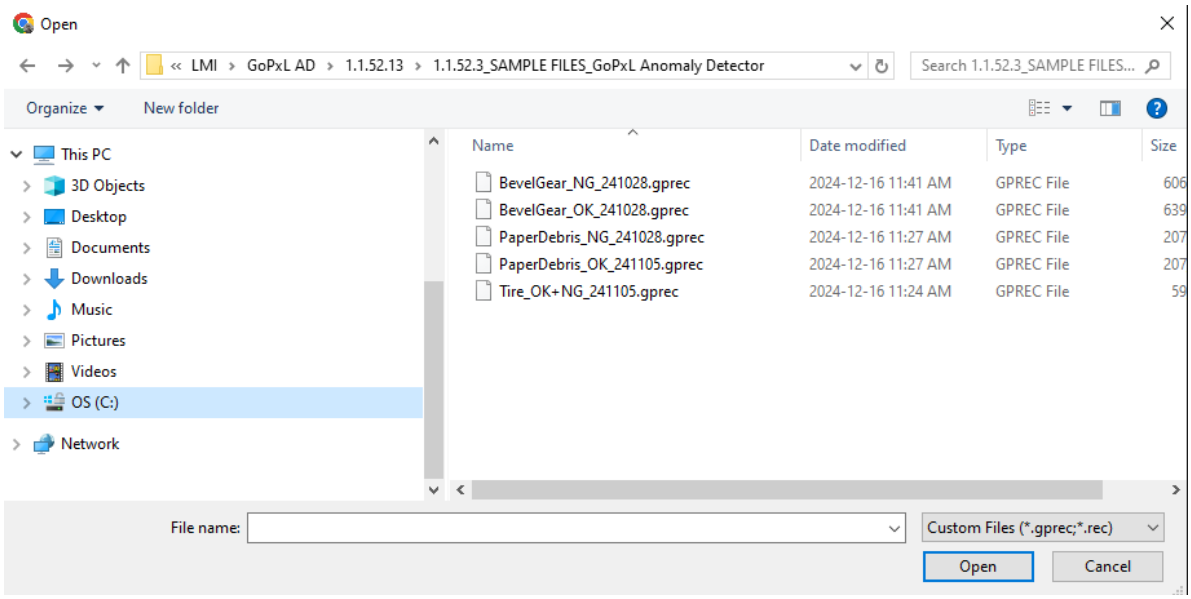
To load any of these samples:

1. Using a browser, connect to your GoMax unit.

   By default, it's IP address is 192.168.1.6.

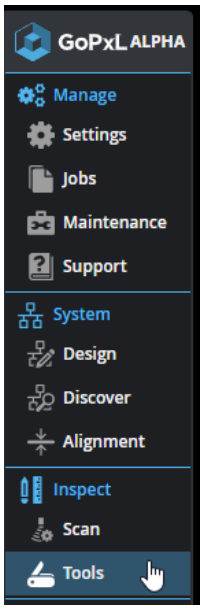
2. Upload the `.gprec` file corresponding to the sample you want by clicking the Upload replay file button and navigating to the location of the sample files.

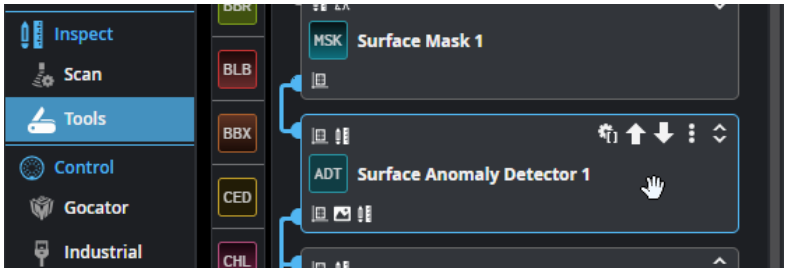




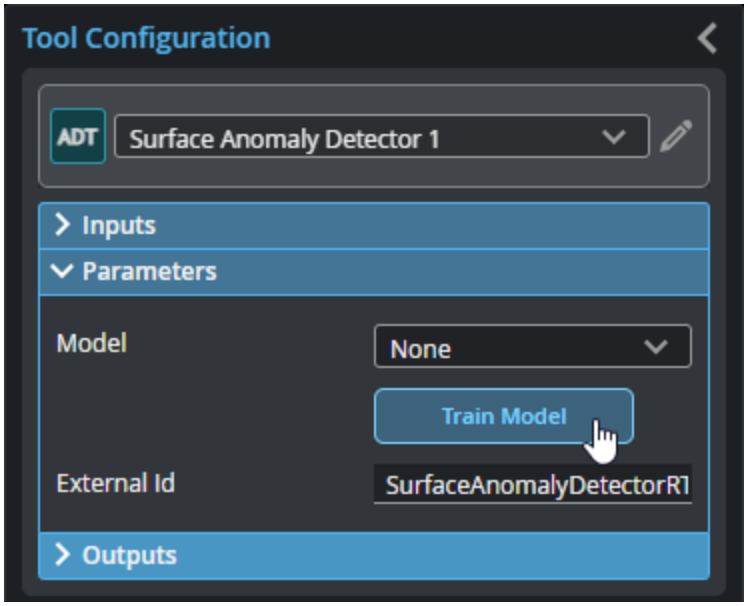
   The replay data, with a previously added pre-processing tool chain, opens.

3. Switch to the Tools page and click the Surface Anomaly Detector tool.

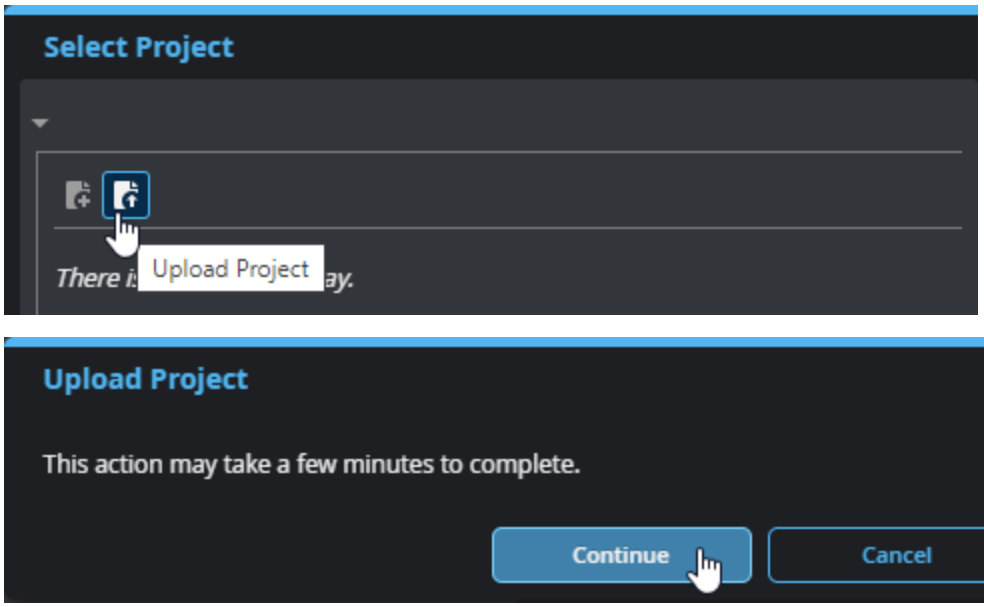




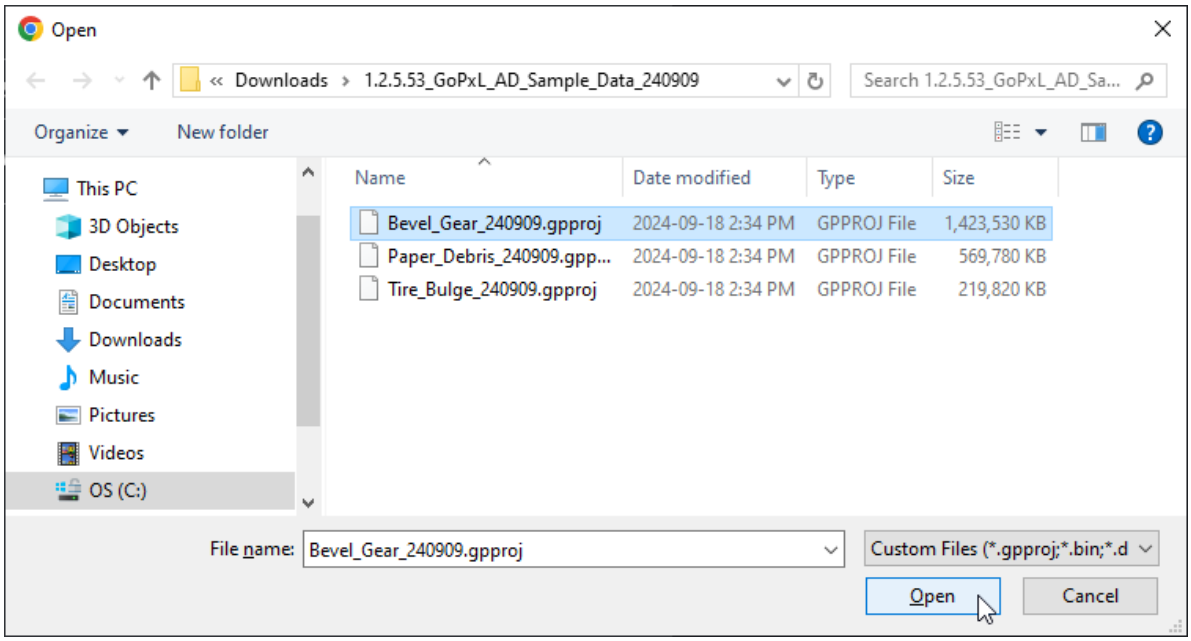
4. In the Tool Configuration panel, click **Train Model**.



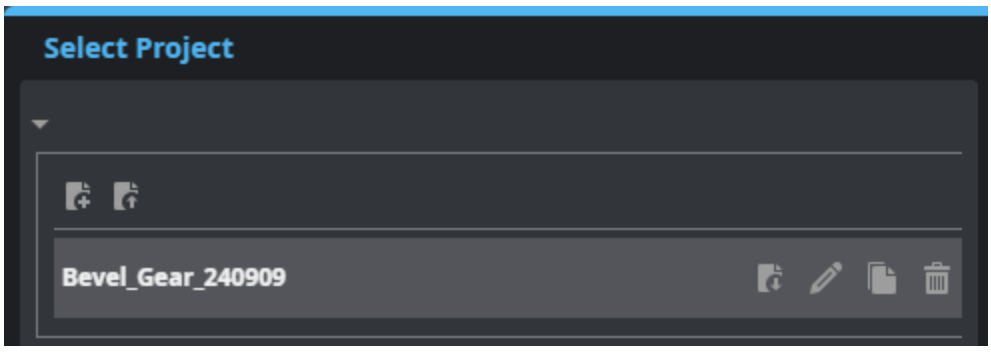
5. In the Select Project page, click Upload project and click **Continue**.



6. Go to the location of the sample files and select and open the corresponding project file.

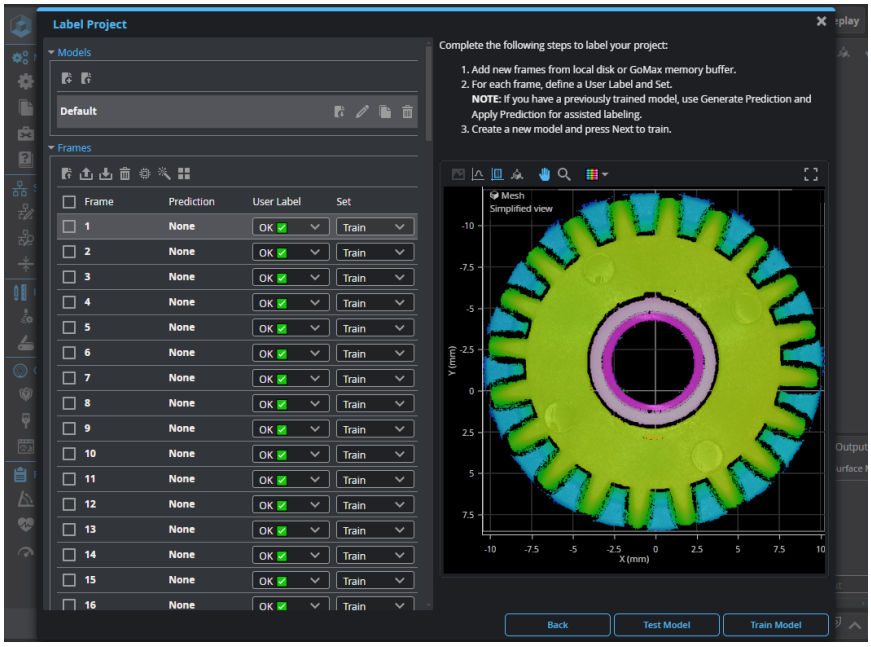


Loading the project may take several minutes.



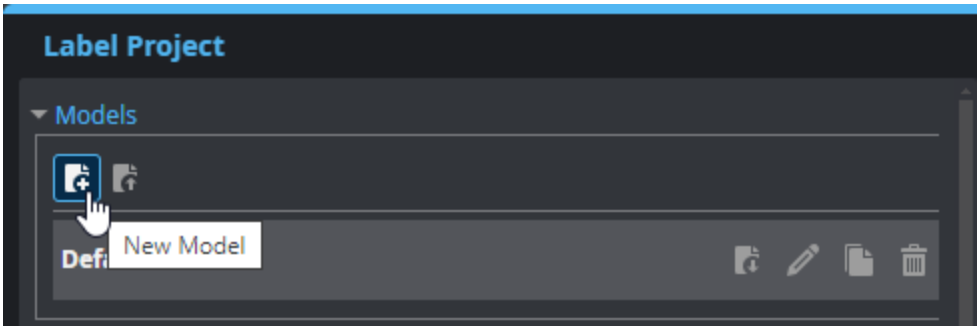
7. After the file has loaded, click **Next** in the lower right of the Select Project page.

   The Label Project page opens, displaying the labeled frames in the project.



At this point, you can familiarize yourself with the Label Project page, or do one of the following:

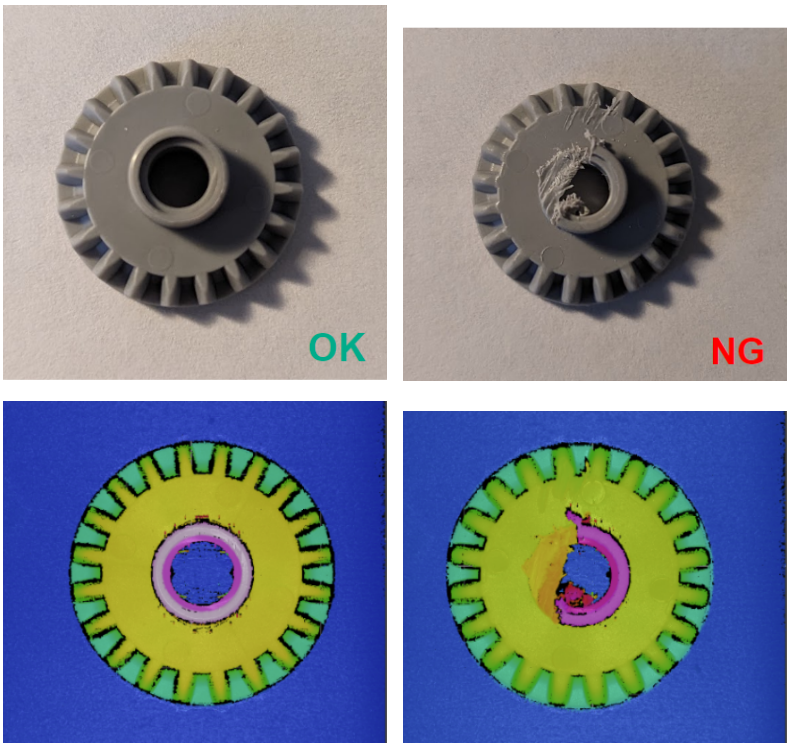
- Click **Test Model** to test the performance of the previously trained model on the frames. For more information, see *Test Model Performance* on page 732.
- Create a new model, click **Train Model** at the lower right of the Label Project page, configure the new model's training parameters, and start the training. For more information, see *Configure Training* on page 726.

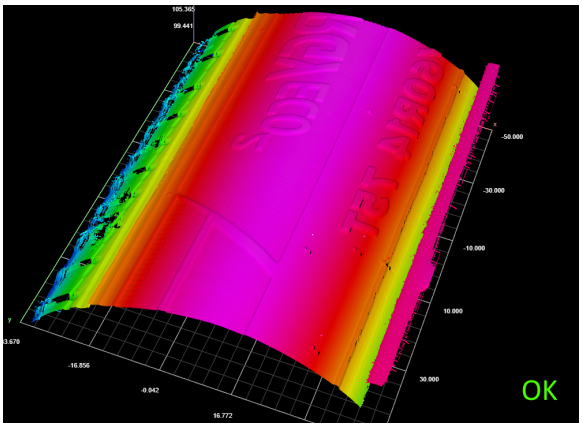
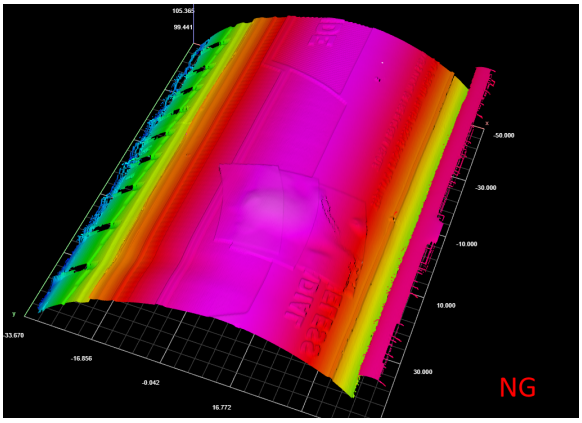


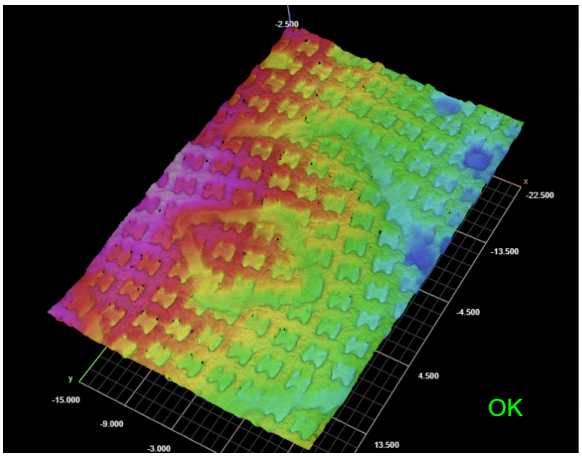
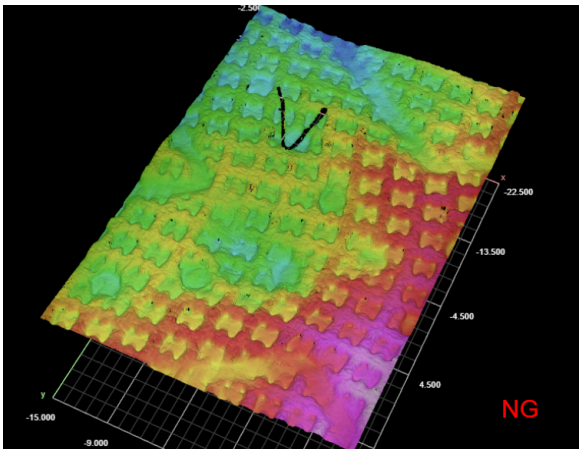
The following datasets are available:

Datasets

Sample	Description
Bevel gear	This dataset demonstrates discrete part inspection of mechanical parts, such as gears and automotive components. We added defects manually to simulate anomalies.

Sample	Description
	
Tire bulge	This dataset was created to demonstrate detection of a bulge defect on a tire. This type of defect could be the result of foreign materials. 

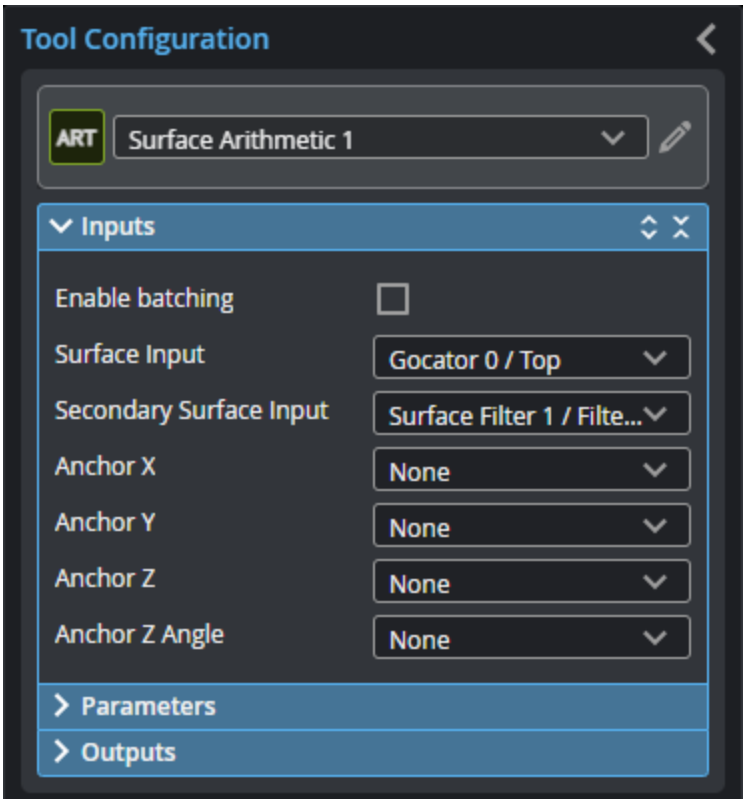
Sample	Description
	  The previously trained model uses a **Model resolution** setting of 512x512 and includes augmentations. In the advanced settings, **Epochs** is set to 20 and **Clamping** is set to 1.
Paper debris	This dataset demonstrates foreign object detection on web materials such as foil, metal, tire, battery, paper, and so on. 

Sample	Description
	  The previously trained model uses a **Model resolution** setting of 512x512.

Surface Arithmetic

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.

Inputs



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

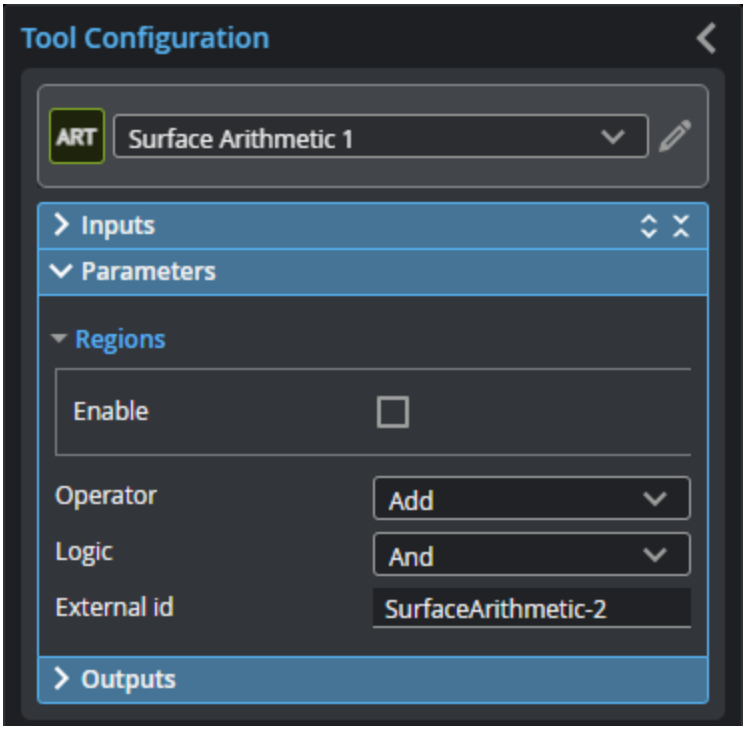
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Secondary Surface Input	The secondary input the tool uses to perform arithmetic.

Name	Description
Anchor X Anchor Y Anchor Z	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

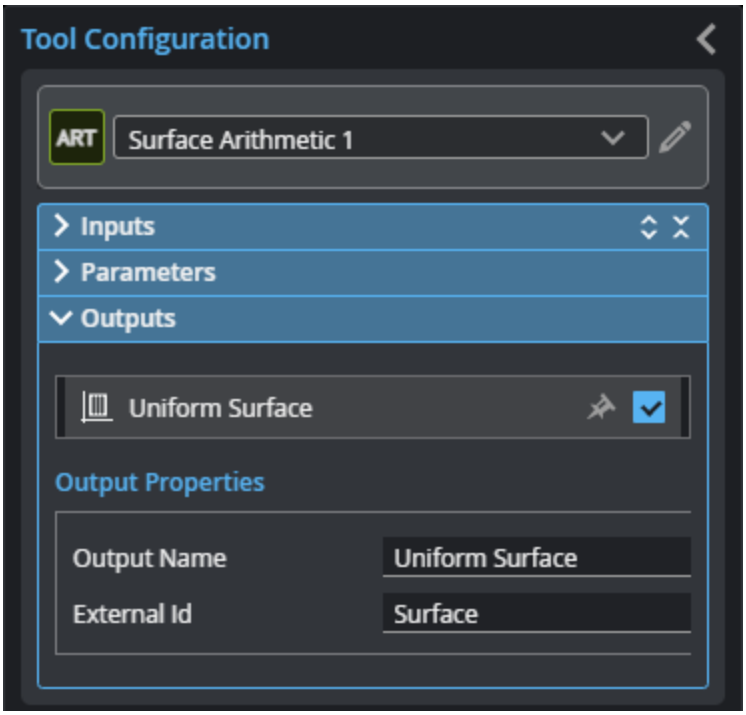


Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Operator	One of the following: **Add**: Adds the height values of the corresponding data points in the two sources.

Parameter	Description
	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**.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Data

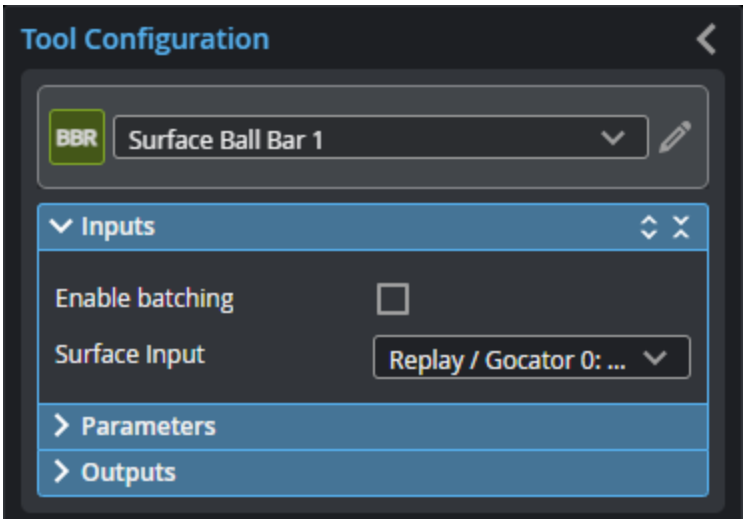
Type	Description
Uniform Surface	The processed surface data.

Surface Ball Bar

The Surface Ball Bar tool returns measurements useful for calibrating systems using a ball bar, particularly systems that include a robot.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

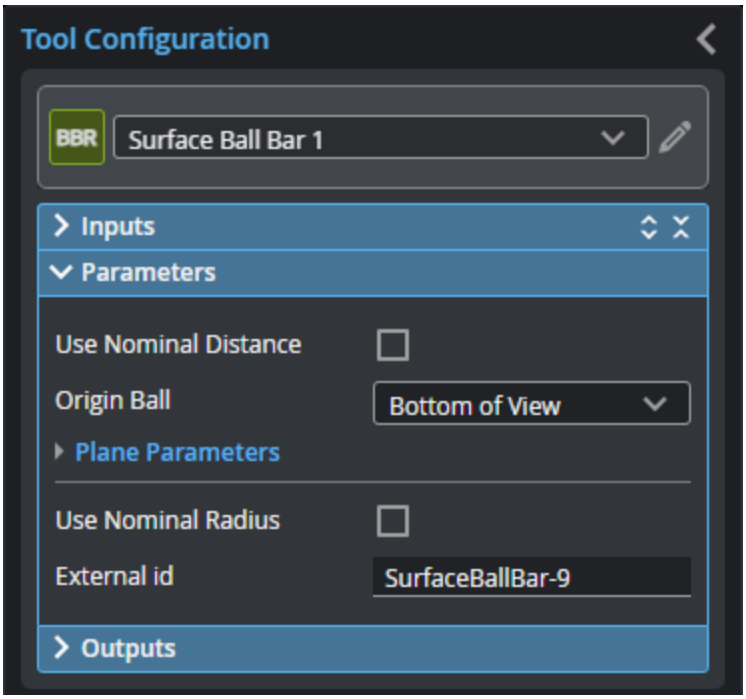


Inputs

Name	Description
Enable Batching	Leave this setting unchecked.
Surface Input	The data the tool applies measurements to or processes.

Parameters

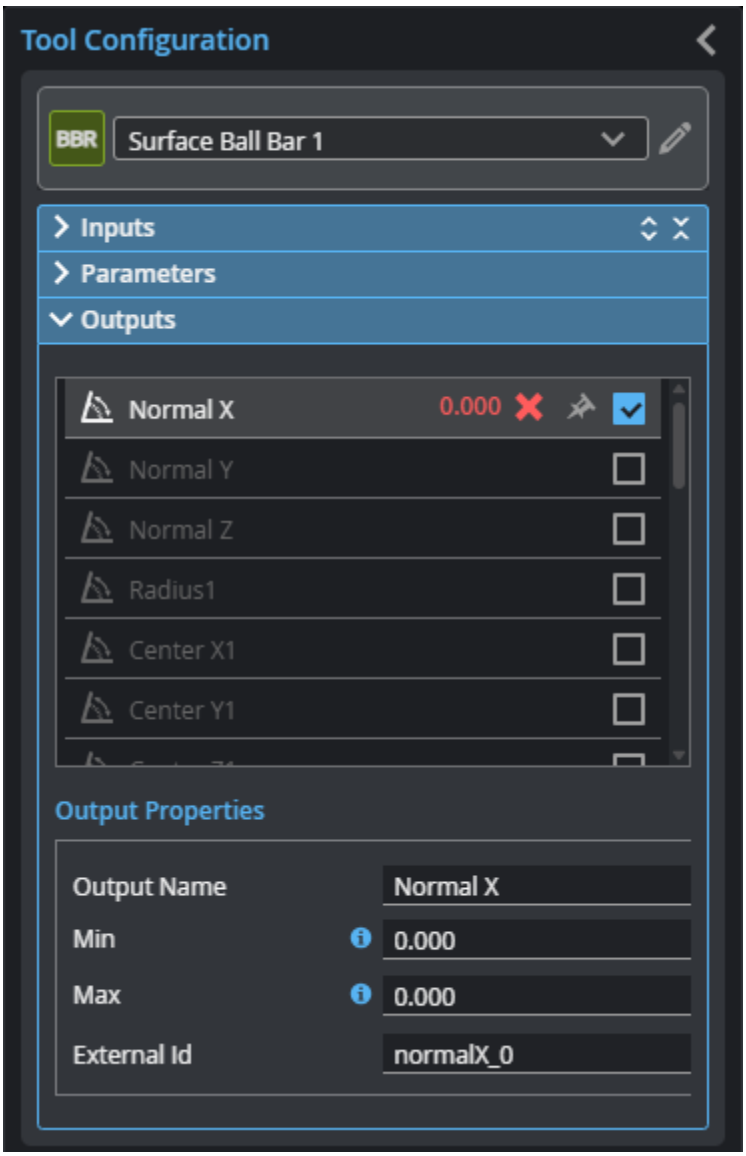
You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Use Nominal Distance	When enabled, displays **Nominal Distance** and **Distance Tolerance** settings. Set these to 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.
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 web interface.
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.
Use Nominal Radius	When enabled, displays **Nominal Radius {n}** settings. Set these to 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. The tool uses these as a starting point and will not necessarily reject targets based on these.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



Measurements

Measurement
Normal X / Y / Z These measurements return the X, Y, and Z components of the normal vector of the surface surrounding the calibration target.
Radius1 **Radius2** These measurements return the radius of each ball.
Center X1 / Y1 / Z1 **Center X2 / Y2 / Z2**

Measurement
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.)
Distance 3D The direct distance between the centers of the spheres fitted to the balls.
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.

Data

Type	Description
Difference Surface	The difference between the scan data and the provided nominal dimensions. Used for diagnostics.

Surface Blob

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 a minor tilt of the target surface (up to 10 degrees); this lets you detect low or shallow defects that would otherwise not be detectable due to a tilt.

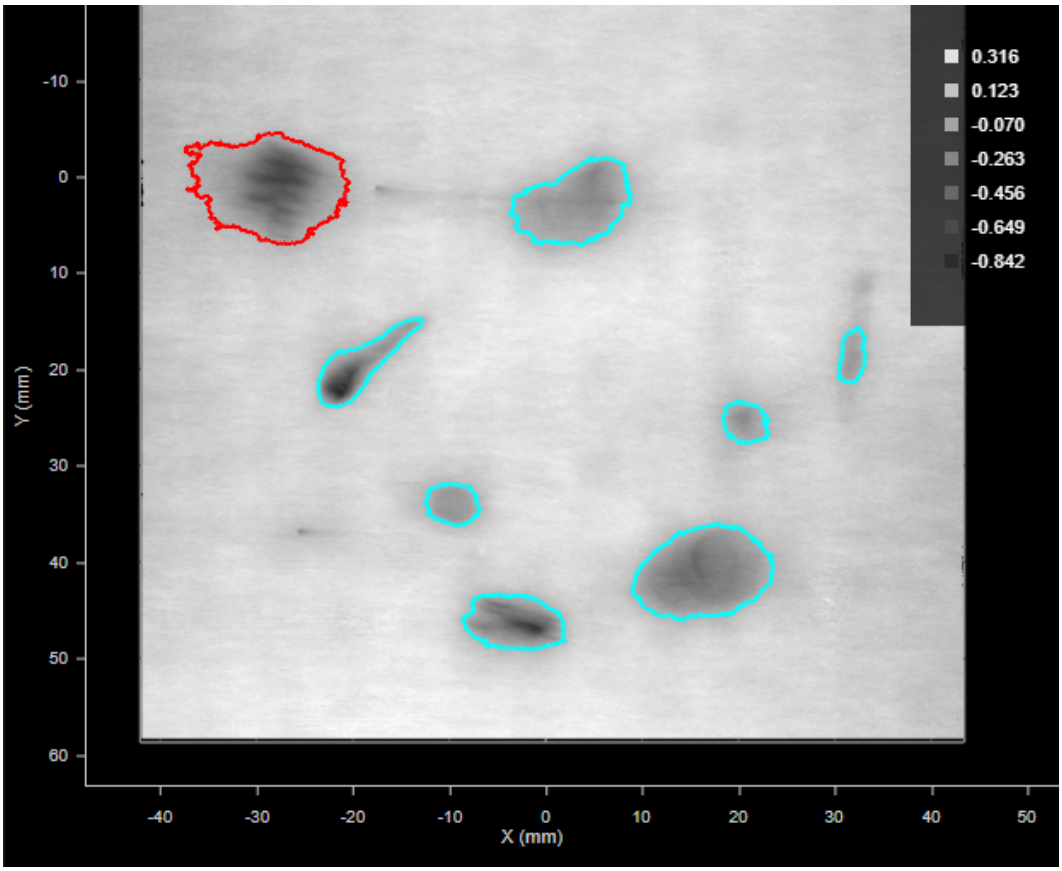


The Surface Blob tool provides functionality similar to the Surface Segmentation tool. For information on the Surface Segmentation tool, see *Surface Segmentation* on page 951.

For a comparison of these tools and the part detection capabilities you can configure on the **Acquire** page, see *Isolating Parts from Surface Data* on page 712.

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.



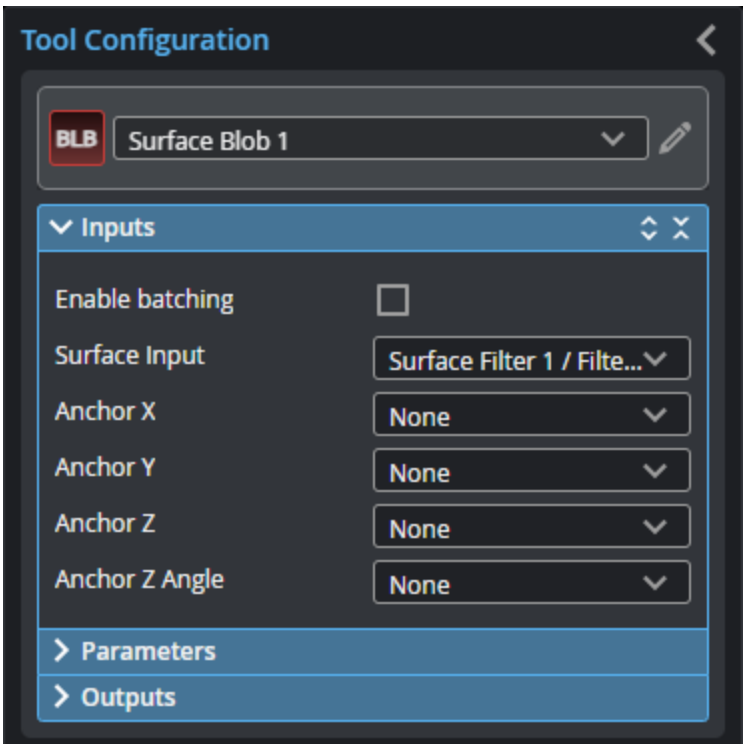
Several dents outlined by the tool on a surface in cyan. The currently selected blob is outlined in red. (Grayscale heightmap mode is used to better see the outlines.)

Note that knowing the rough size and shape of the kinds of defects you expect is important when you are configuring the open and close kernels and the tool's filters.

> When **Enable Array Output** is checked, the tool outputs the blobs in an array called Blob Surface. Be *sure* to enable Blob Surface in the **Outputs** section after enabling this parameter.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs



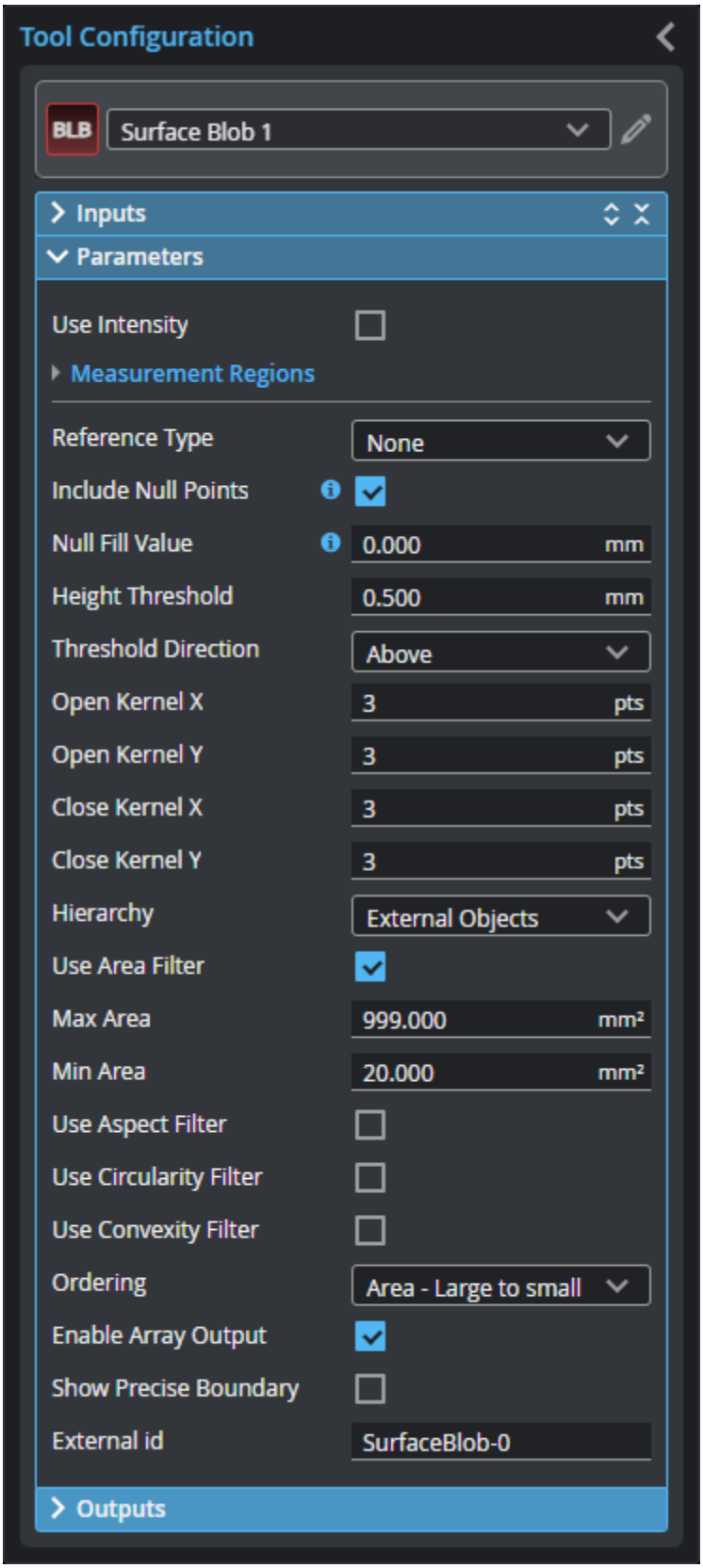
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

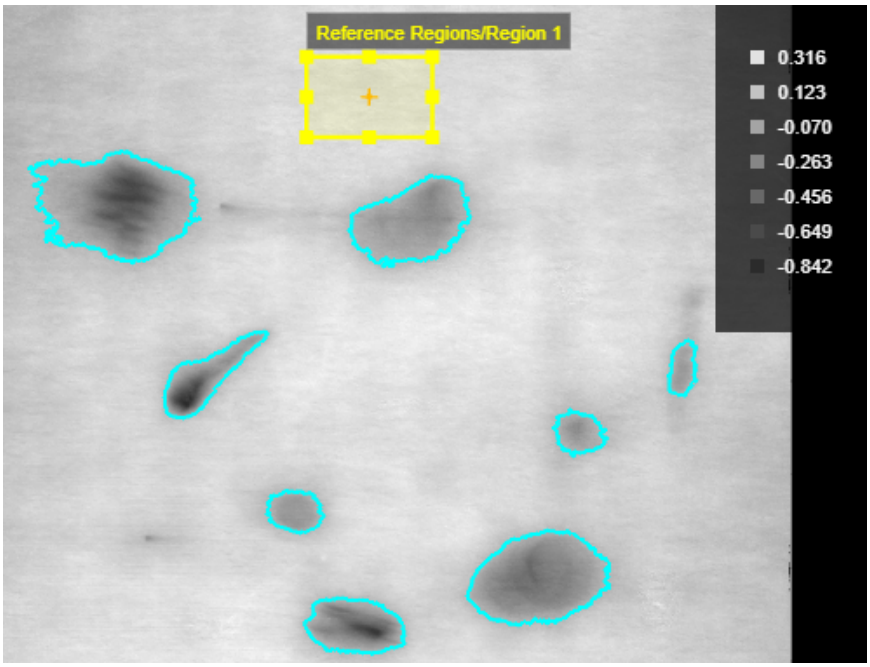
Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. This setting is hidden when the tool's **Enable Array Output** setting is checked. For more information, see *Enable Array Output* on page 769. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	The X or Y measurement of another tool that this tool uses as a positional anchor. Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.
Plane	The Plane geometric feature the tool uses when **Reference Type** is set to Reference Plane.

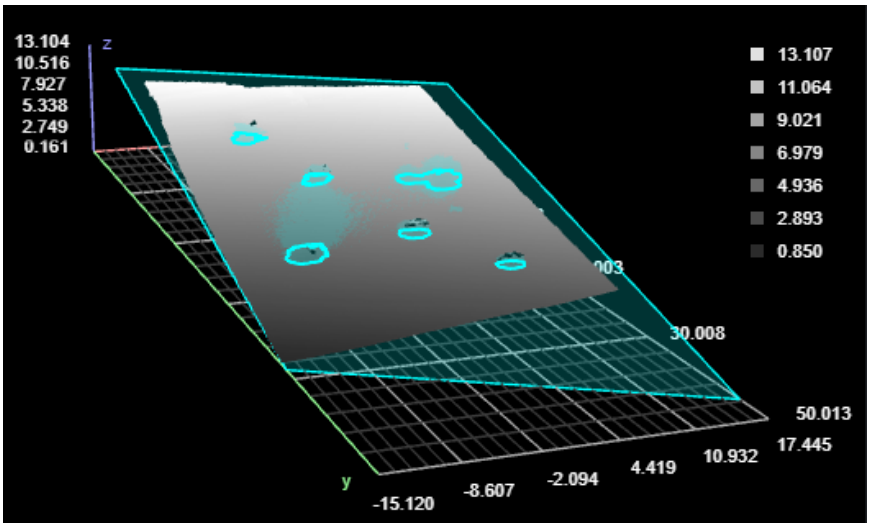
Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

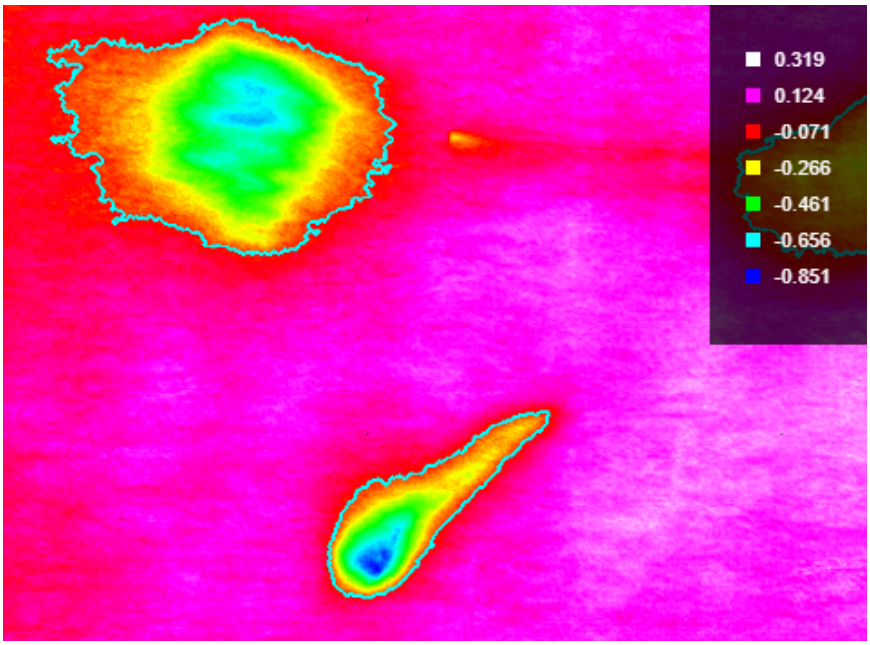
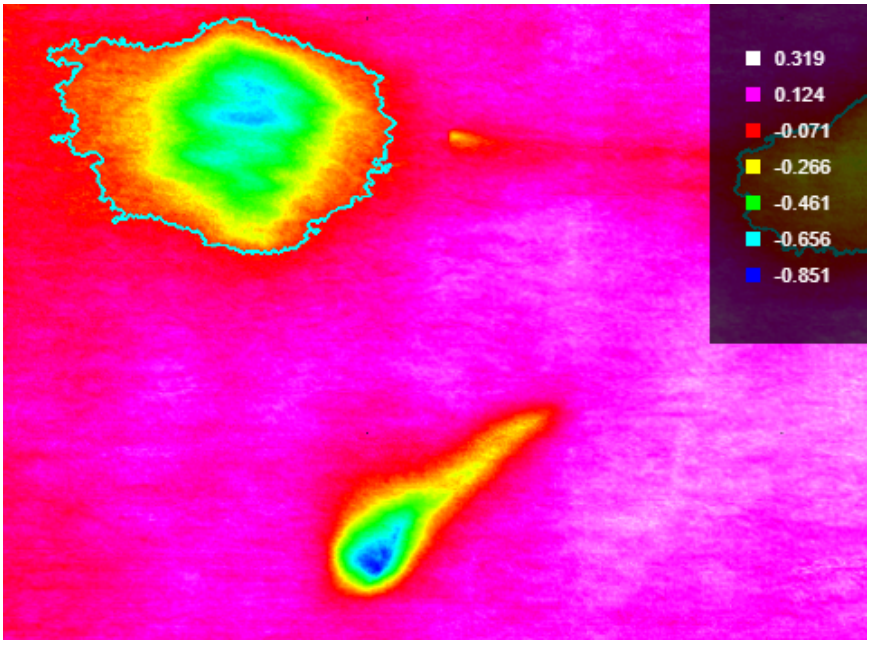

Tool Configuration
BLB
Surface Blob 1
Inputs
Parameters
Use Intensity
Measurement Regions
Reference Type
None
Include Null Points
Null Fill Value
0.000
mm
Height Threshold
0.500
mm
Threshold Direction
Above
Open Kernel X
3
pts
Open Kernel Y
3
pts
Close Kernel X
3
pts
Close Kernel Y
3
pts
Hierarchy
External Objects
Use Area Filter
Max Area
999.000
mm²
Min Area
20.000
mm²
Use Aspect Filter
Use Circularity Filter
Use Convexity Filter
Ordering
Area - Large to small
Enable Array Output
Show Precise Boundary
External id
SurfaceBlob-0
Outputs

Parameters

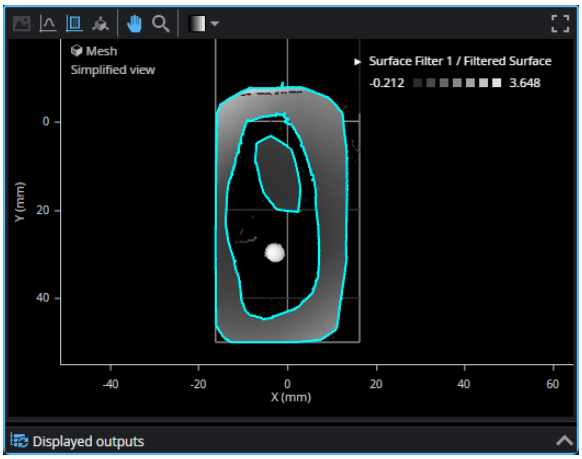
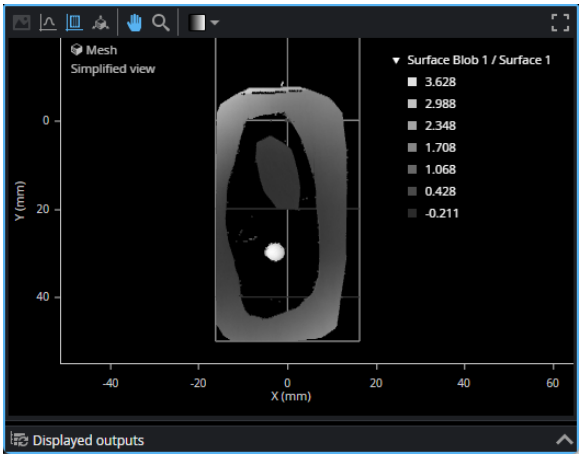
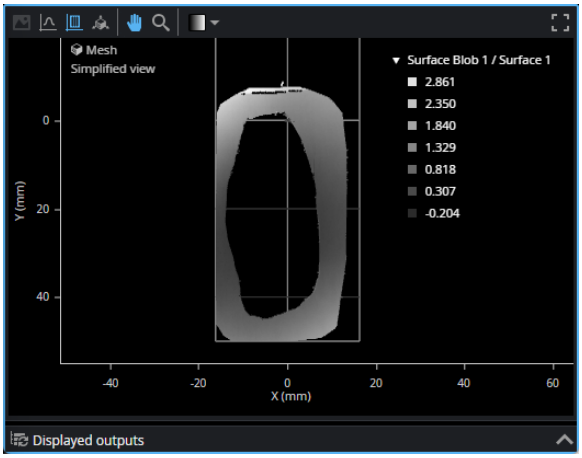
Parameter	Description
Use Intensity	If enabled, the tool uses intensity data instead of heightmap data. Only available if **Acquire intensity** is enabled on the **Acquire** page, in the **Scan mode** section during scanning; for more information, see *Scan Modes and Intensity* on page 139.
Measurement Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
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 Reference Region and Reference Plane options, see the descriptions of the **Reference Region Type** and **Reference Plane** parameters below.
Reference Region Type	If you set **Reference Type** (see above) to Reference Region, the tool displays a drop-down that lets you configure the reference region. (For details, see *Flexible Regions* on page 217.) 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 214.

Parameter	Description
Reference Plane	If you set **Reference Type** (see above) to Reference Plane, the tool uses the specified plane geometric feature to correct for a tilt of the target. Note however that using a reference plane to correct the tilt distorts the scan data: it sheers the data by the same angle as the tilt. The maximum tilt angle with which you can use the tool therefore depends on how much sheer angle you can tolerate in your application (which can effect the tool's ability to detect blobs). Typically, you add and configure a Surface Plane tool to generate a plane (for more information, see *Surface Plane* on page 923). For information on geometric features, see *Geometric Features* on page 227. For applications where sheer distortion can't be tolerated, use Surface Transform to correct the tilt (see *Surface Transform* on page 993), and use the latter tool's output as the input for Surface Blob. 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 **Intensity Threshold**. Otherwise, it is named **Height Threshold**.
Threshold Direction	Determines whether data above or below the threshold is considered as being a

Parameter	Description
	blob. **Below**: The **Height Threshold** value is the maximum that will be considered as part of a blob (for example, a dent below the surrounding surface). **Above**: The **Height Threshold** value is the minimum that will be considered as part of a blob (so a raised feature).
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 objects only or all objects. **External Objects** Use this option to ignore smaller objects in larger objects. Only the outermost objects are returned. **All objects** Use this option to include smaller objects in larger objects. **Holes** Use this option to get holes inside objects. **All Contours** Use this option to get all detected contours. When you choose this option, an **Include Inner** checkbox is displayed (see below).
Use Area Filter	If **Use Area Filter** is enabled, the tool applies an area filter to potential blobs using the values in **Max Area** and **Min Area**.
Use Aspect Filter	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 lowest dent in a surface is included as a blob if the aspect values are set to 1 and 0.350, respectively (the rotated bounding box encapsulating would be 13.461 mm x 4.782 mm).

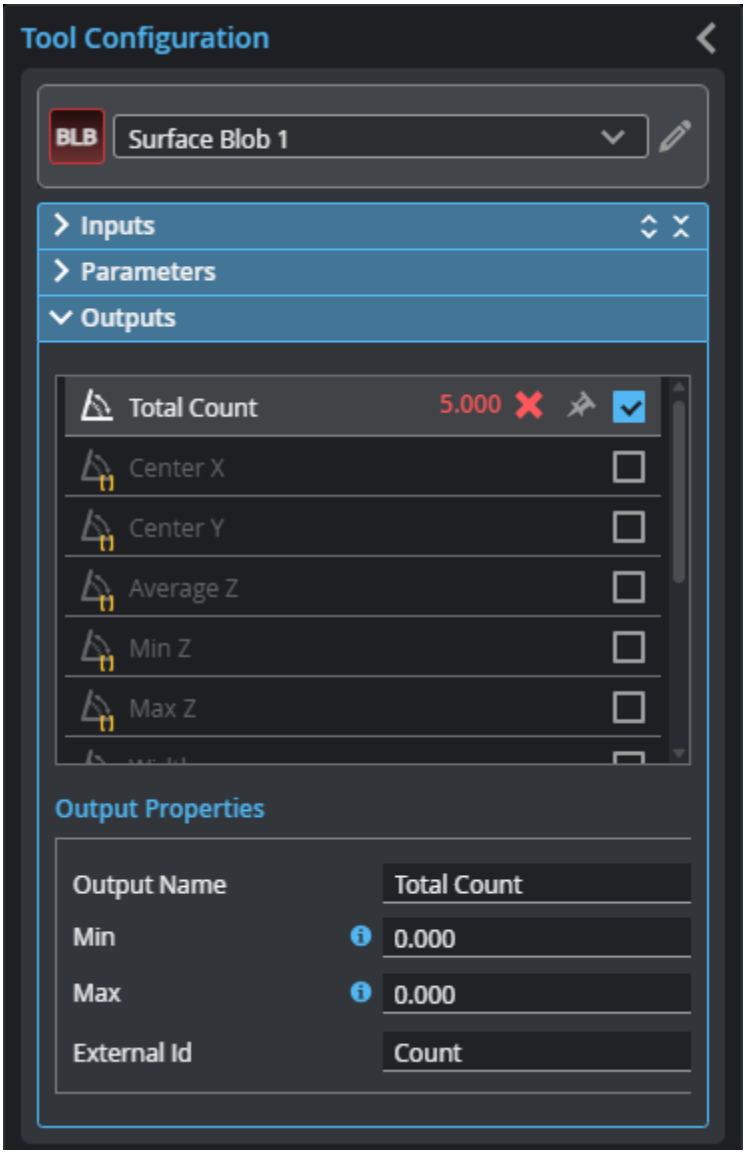
Parameter	Description
	 In the following, the same dent is excluded if **Min Aspect** is set to a value greater than 0.354. 
Use Circularity Filter	If **Use Circularity Filter** is enabled, the tool applies a circularity filter to potential blobs to measure how close to a circle the blob is, using the 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.

Parameter	Description
Use Convexity Filter	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. One of the following: • Position - X increasing • Position - X decreasing • Position - Y increasing • Position - Y decreasing • Area - Large to small • Area - Small to large
Enable Array Output	When **Enable Array Output** is unchecked, the tool outputs each blob as an individual Surface output. The tool can only take an array as input when this is unchecked. For more information, see *Enable Batching* on page 762. When **Enable Array Output** is checked, the tool outputs the blobs in an array called Blob Surface. Be *sure* to enable Blob Surface in the **Outputs** section after enabling this parameter. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
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. If **Enable Array Output** is checked, this parameter is hidden.
Show Precise Boundary	Displays the boundaries of the blobs so that they appear on the Surface data, rather than floating above the Surface. This makes it easier to evaluate the blob boundaries in relation to the Surface data. This setting does not affect measurements or other outputs. Disable this setting for maximum performance.
Include Inner	Uncheck this to exclude inner Surface data in the extracted blobs. Only available when **Hierarchy** is set to All Contours. For example, the following contours have been found by the tool:

Parameter	Description
	 When **Include Inner** is enabled, the Surface data inside the large ring is included in the output Surface data (whether that data is itself considered a blob or not):  When **Include Inner** is disabled, the inner data is excluded: 
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

If **Enable Array Output** is checked, the tool outputs arrays for the center position, length, width, and area measurements, as well as the Center Point output.

If **Enable Array Output** is unchecked, the tool outputs individual, numbered measurements. Use the **Number of Blob Outputs** setting to determine the number of measurements listed in the **Measurements** tab.

Measurements

Measurement
Total Count Returns the total number of blobs identified, based on the tool's parameters.
Center X {n} / Center X **Center Y {n} / Center Y** The X and Y positions of the center of mass of a blob extracted from the surface.
Average Z {n} / Average Z The averaged Z positions of a blob extracted from the surface. Average Z is calculated from all valid blob points, whereas Center.Z is calculated from a relatively small rectangle region around the center (7 by 7 data points); for this reason they are usually not consistent.
Min Z {n} / Min Z **Max Z {n} / Max Z** The minimum and maximum Z positions of a blob extracted from the surface.
Width {n} / Width **Length {n} / Length** 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.
Area {n} / Area 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 1014.

Features

Type	Description
Center Point {n} / Center Point	The point representing the center of a blob.



For more information on geometric features, see *Geometric Features* on page 227.

Data

Type	Description
Diagnostic Surface	Surface data you can use to evaluate the impact of the tool's parameters, after filters are applied, to properly separate the areas corresponding to the defects or targets you need to detect.
Surface {n} / Blob Surface	Surface data corresponding to each blob. When the tool is configured to output arrays, the blob surfaces are in Blob Surface.

Surface Bounding Box

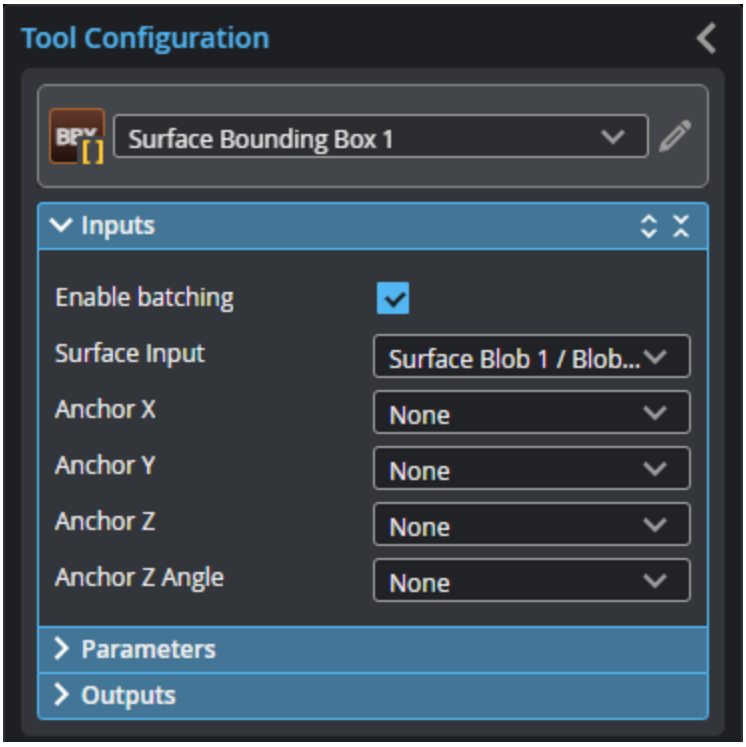
The Bounding Box tool provides measurements related to the smallest box that contains the *scan data* from a part (for example, X position, Y position, width, length, etc.). The tool also lets you get the height of bounding box relative to the Z origin (typically the conveyor on which the target is sitting). This lets you determine, for example, the height of a box or other container on the conveyor as part of a product packaging process. Various settings 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.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

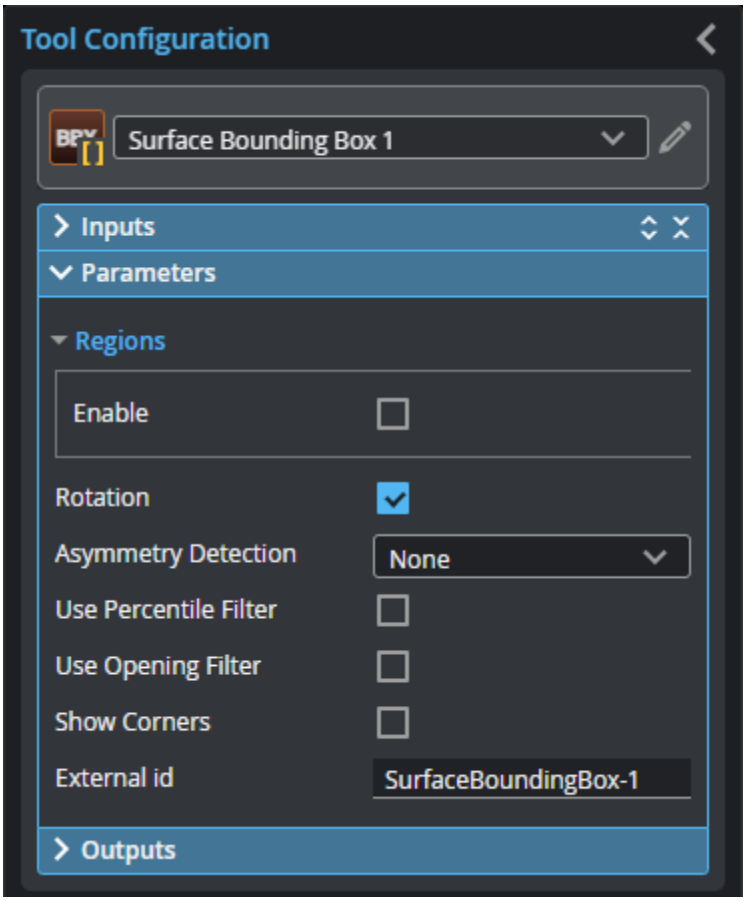
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes

Name	Description
	each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

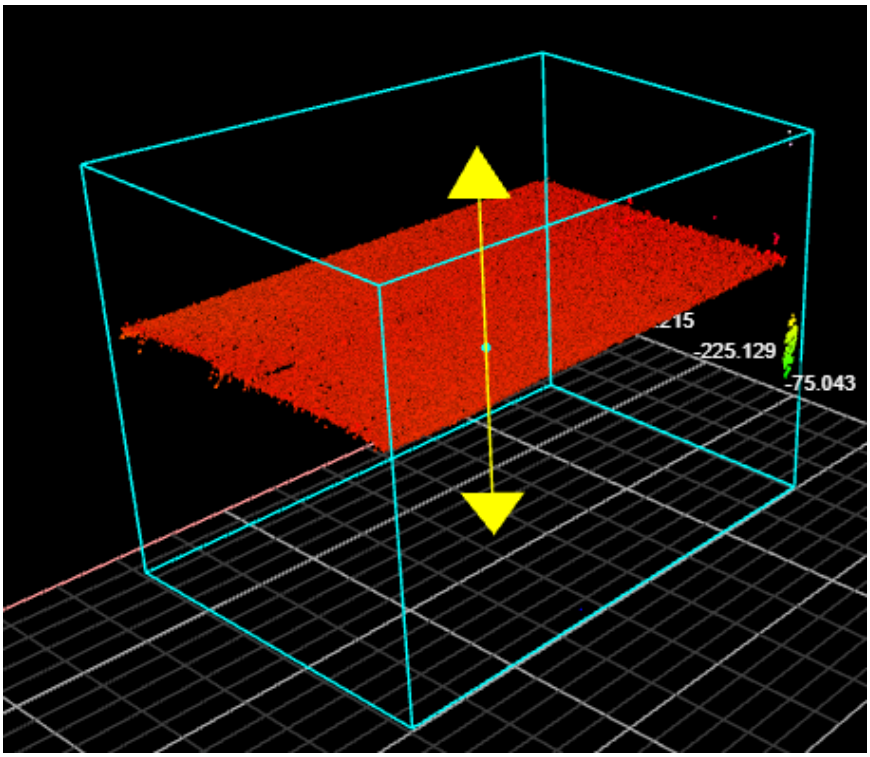
Parameters

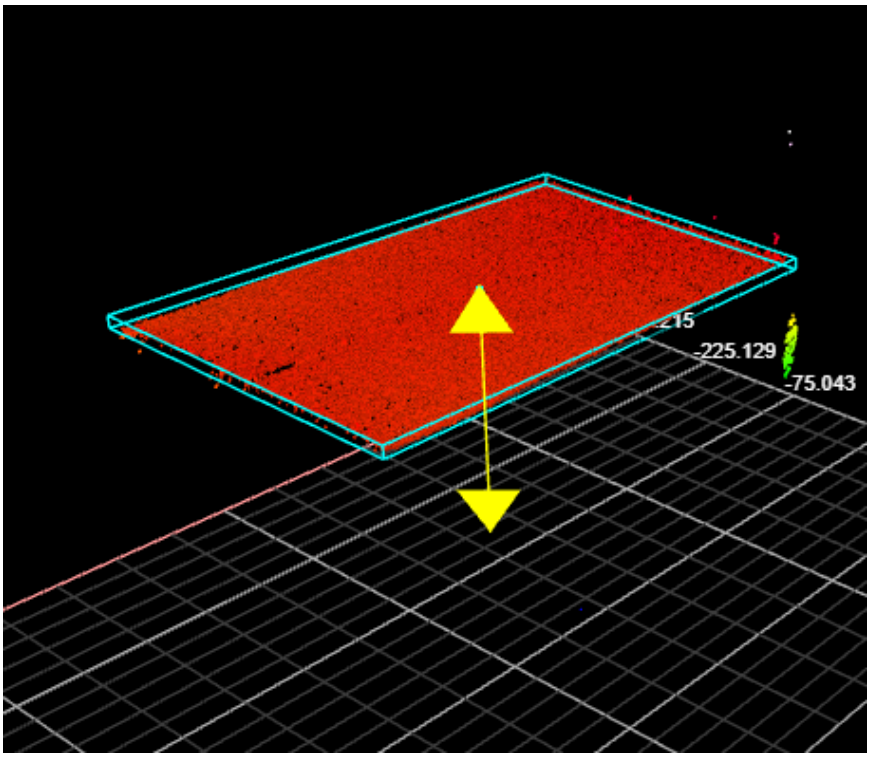
You configure the tool's parameters in the expandable **Parameters** section.

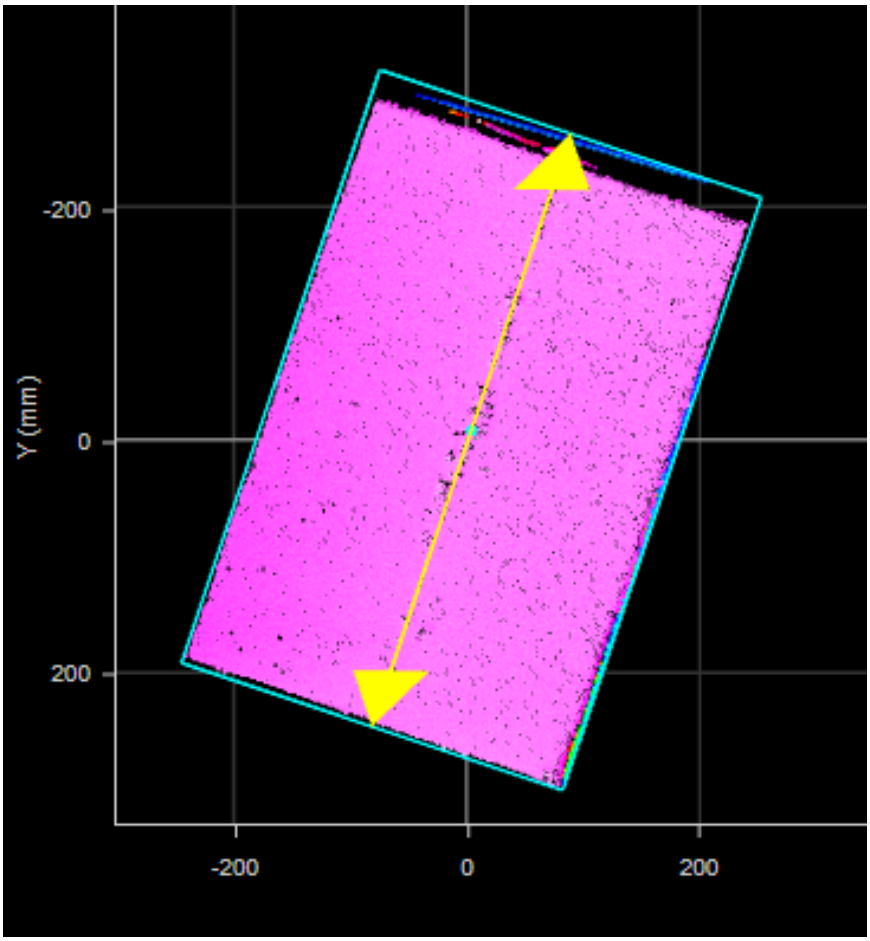


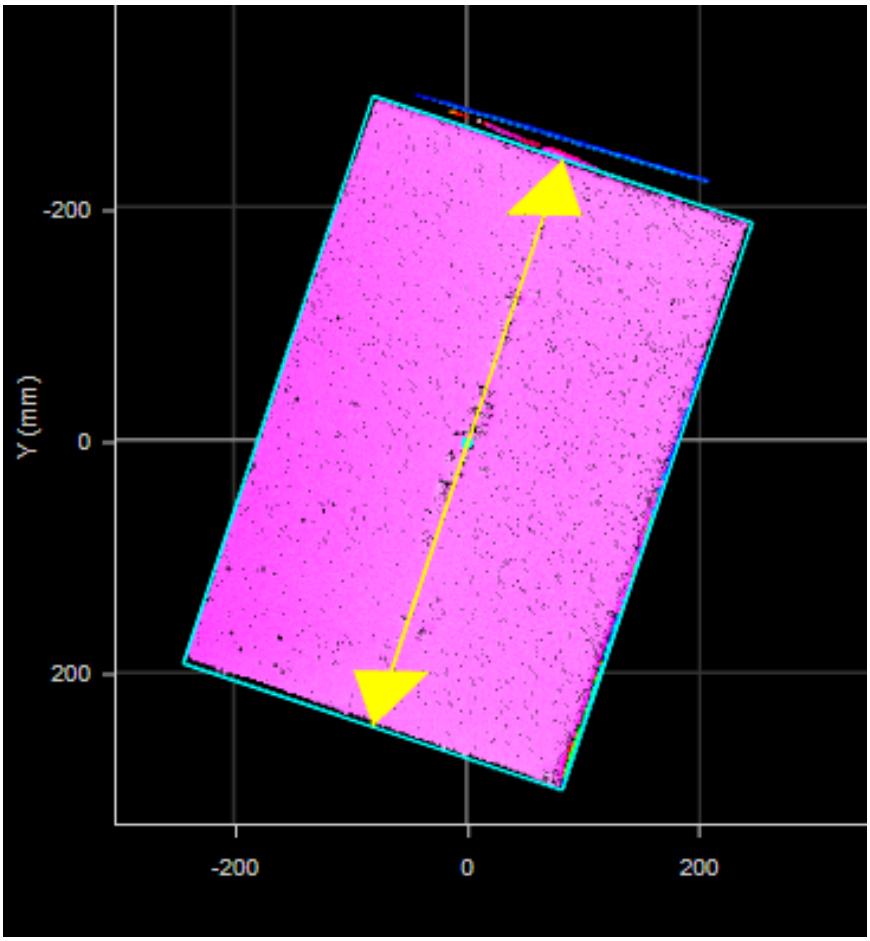
Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Region Type 1 Region 1	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
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. When this setting is enabled, an **Asymmetry detection** setting is displayed.
Asymmetry detection	Determines how the Z angle orientation of an object is returned. The possible values are: None Along Major Axis

Parameter	Description
	Along Minor Axis This setting is only visible if **Rotation** is checked.
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 (from the Z origin) of the box of 406.457 mm. (The following screenshots are from a G2 sensor.)  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.

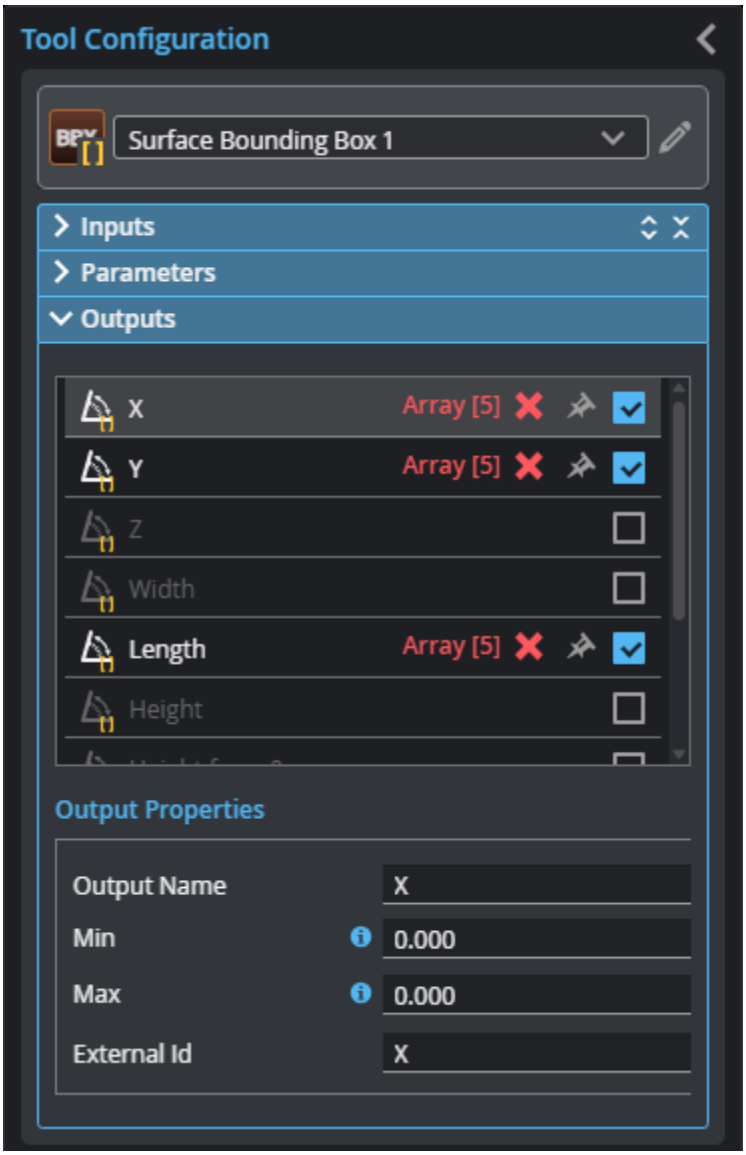
Parameter	Description
	
High Percentile Low Percentile	See **Use Percentile Filter** above.
Use Opening Filter	When enabled, this setting lets you set the value of **Kernel Size** for an *opening* morphological operation applied to the scan data on the XY plane, letting you achieve "robust" width and 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. (The following screenshots are from a G2 sensor.)

Parameter	Description
	 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.
Kernel Size	See **Use Opening Filter** above.
Show Corners	When this setting is enabled, the tool outputs a Point geometric feature for each corner of the box.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

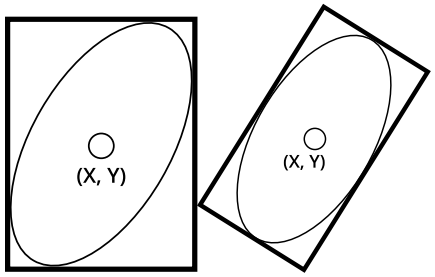
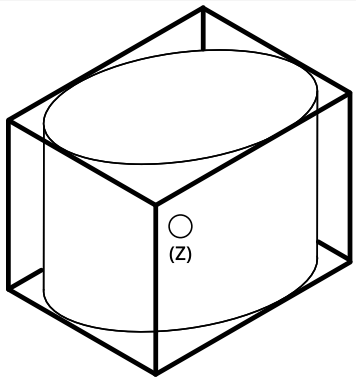
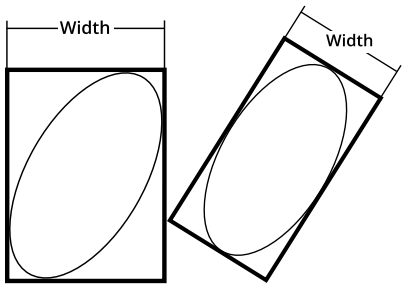
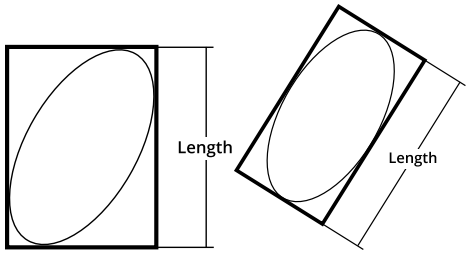
Most tools provide measurements, geometric features, or data as outputs.

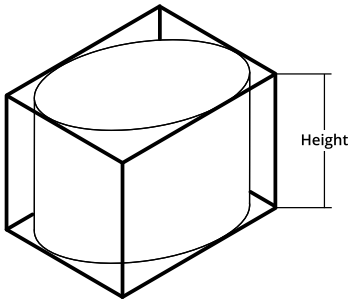
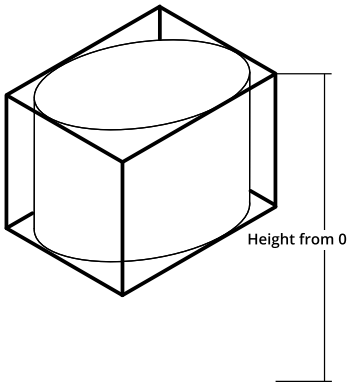
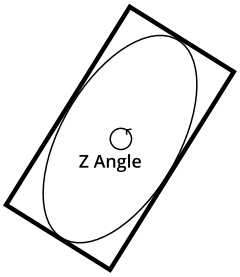


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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*.	
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.	

Measurement	Illustration
Height Determines the height of the bounding box that contains the part.	
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 209.	

Features

Type	Description
Center Point	The center point of the bounding box.
Box Axis Line	The axis of the bounding box.
Corner {n}	Point geometric features representing the corners of the box. Only displayed when **Show Corners** is checked.

For more information on geometric features, see *Geometric Features* on page 227.

Surface 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.

The tool also returns minimum and maximum error points to either side of the circle and gives the error distance measurement of minimum/ maximum points to the circle.

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)

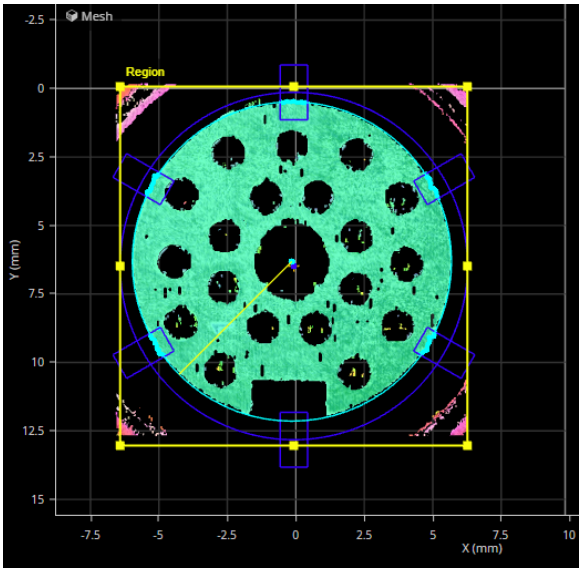
The tool can also generate circle and center point geometric features that Feature tools can take as input for measurement. For more information on Feature tools, see *Feature Tools* on page 252.

Some of the tool parameters are hidden unless **Show Advanced Parameters** is checked.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

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

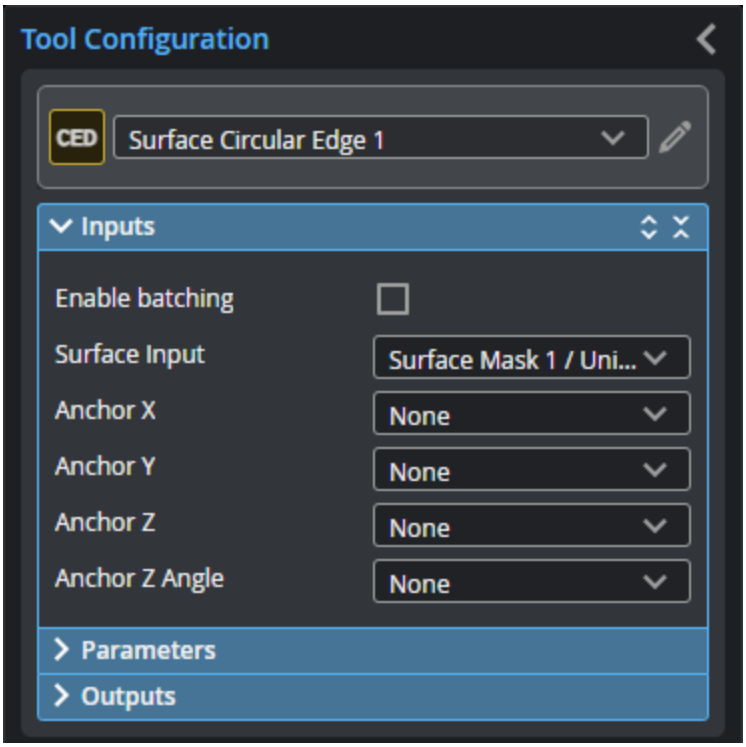
The circular path can optionally be partial, and starts at a defined orientation around the Z axis. The circular path can be as short as 1/4 of a circle, letting it work with rounded corners. Calipers extend vertically to fill the entire region of interest.

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.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



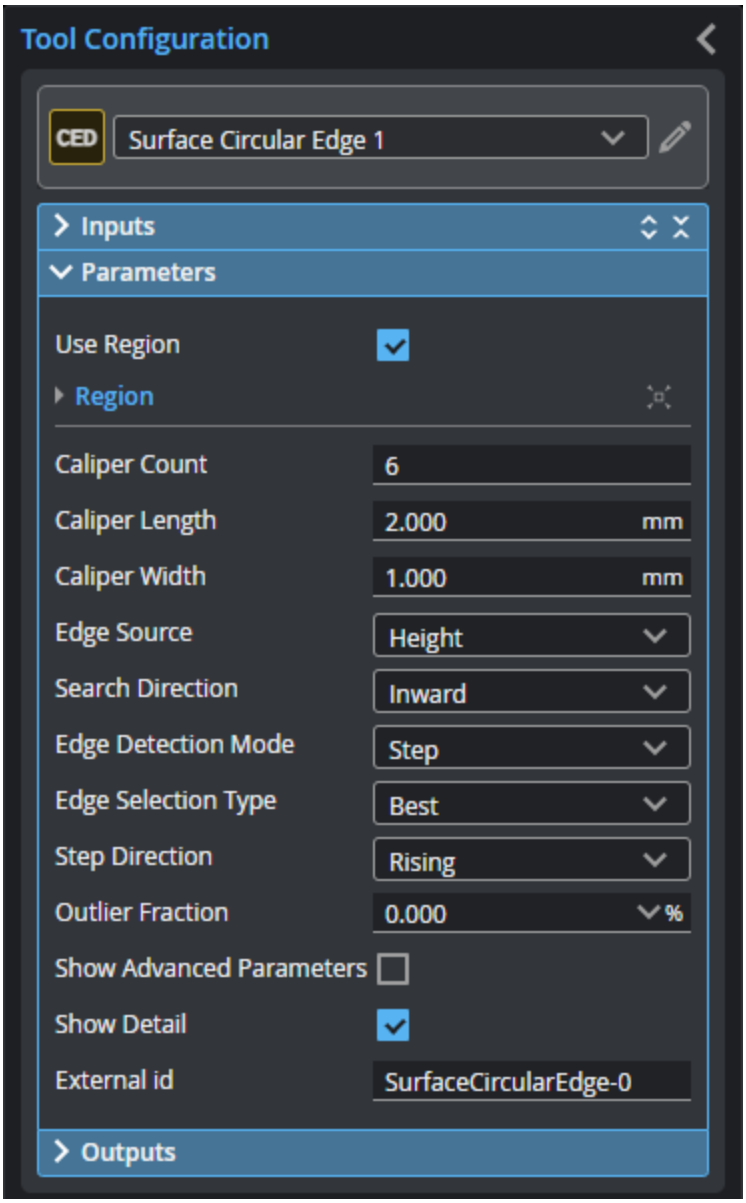
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.
Reference Plane	Uses the feature output of a Surface Plane tool as a reference plane. Useful to correct the scan data if the target is slightly tilted.

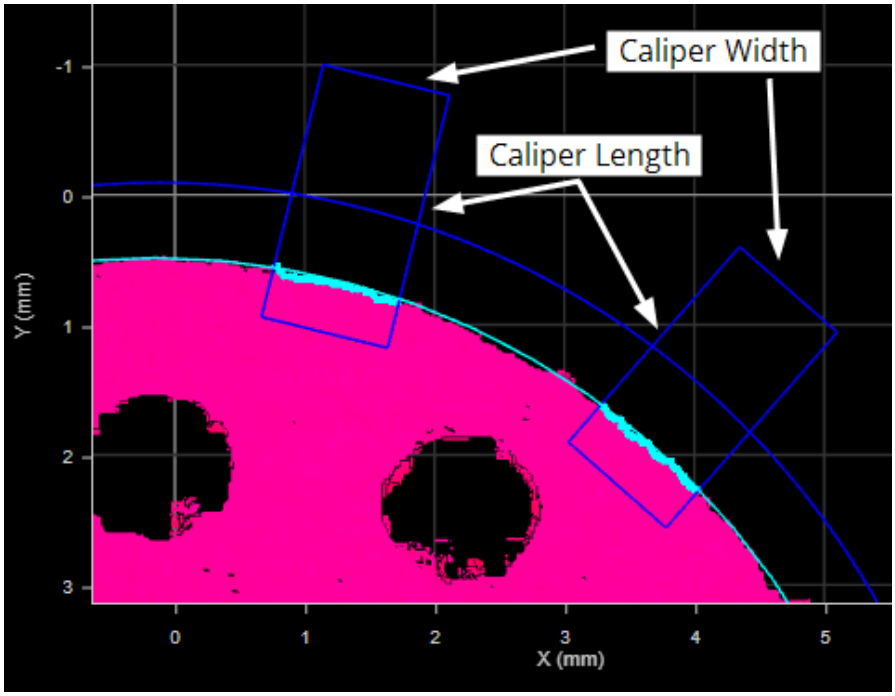
Parameters

You configure the tool's parameters in the expandable **Parameters** section.

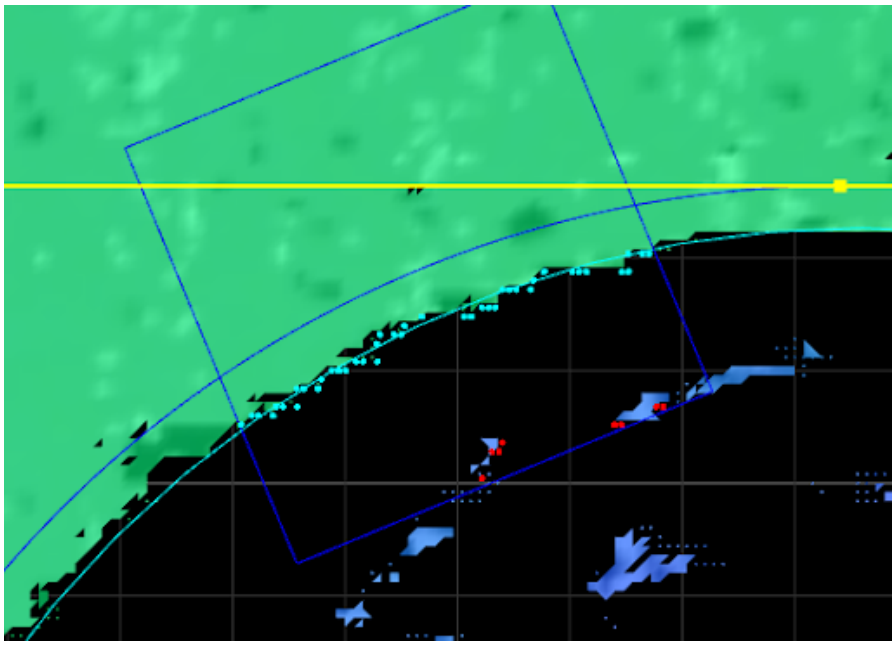


Parameters

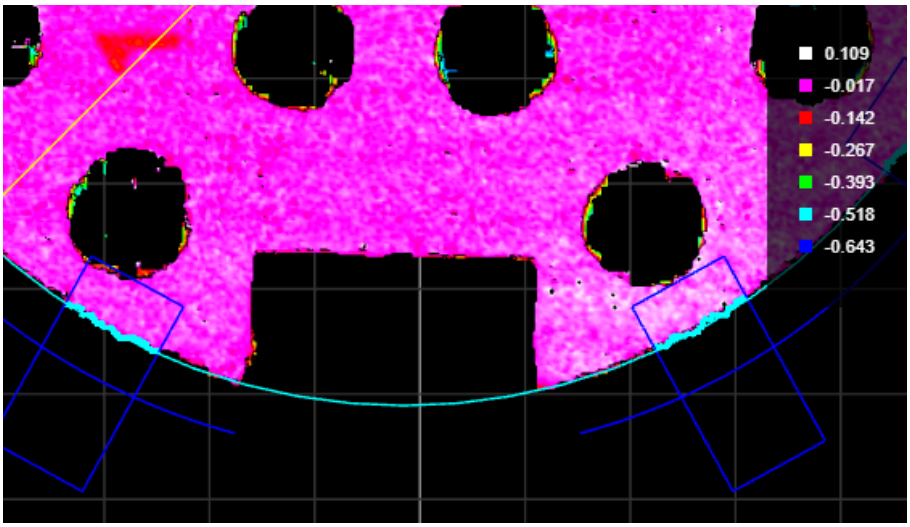
Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.

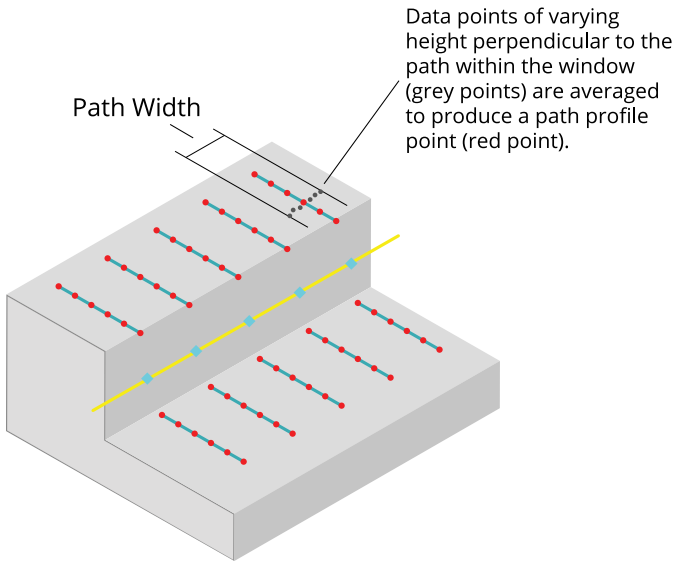
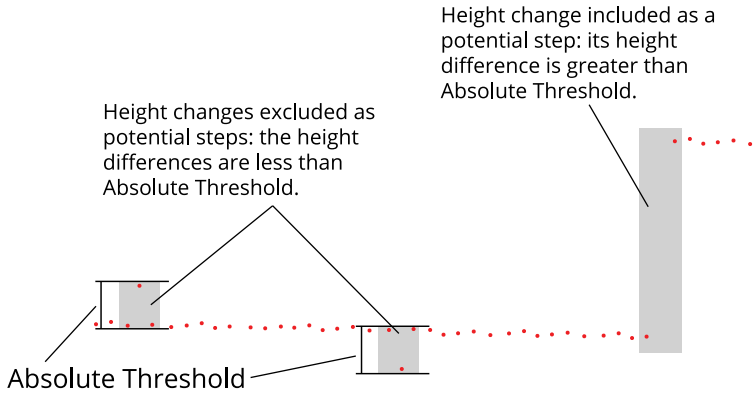
Parameter	Description
Caliper Count	The number of calipers the tool places along the circular path. Using a higher number of calipers increases the amount of data available to the tool, but also increases the amount of time the tool takes to run. Choose a balance between the runtime of the tool and the 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.
Search Direction	Specifies the search direction along the calipers. Either Inward (toward the center of the region of interest) or Outward.

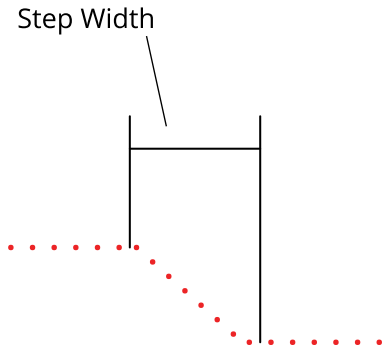
Parameter	Description
Edge Detection Mode	One of the following: Step or Corner. **Step**: Searches for steps on each path profile. **Corner**: Searches for slopes on each path profile. When you choose this mode, several of the tool's parameters are hidden.
Edge Selection Type	Determines which step the 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.
Corner Type	Determines which corner in the search direction on each profile the tool selects. One of the following: **Best**: Selects the greatest corner. **First**: Selects the first corner. **Last**: Selects the last corner. **Top**: Selects the top corner. **Bottom**: Selects the bottom corner.
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.

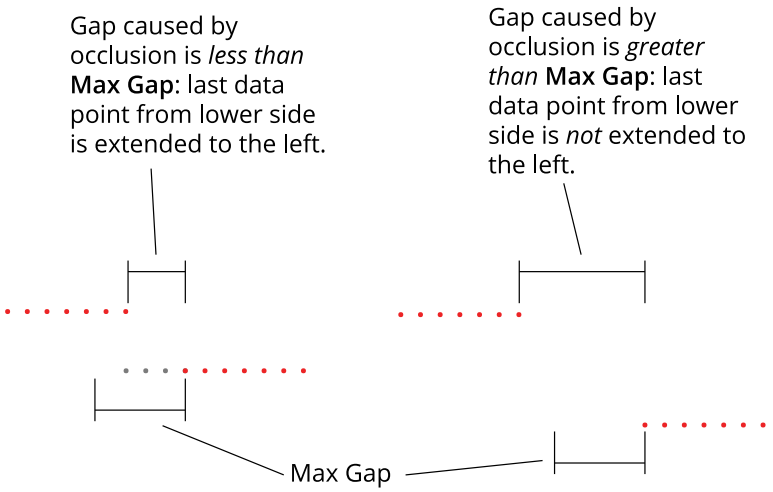
Parameter	Description
Outlier Fraction	The percentage of outlier points to exclude. Setting this to a small value can help the tool fit the circle better to the edge. 
Show Advanced Parameters	When enabled, displays advanced settings. Note that most of these settings are applied *even when they are hidden*. 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.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

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.

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.

Parameter	Description
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).
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.

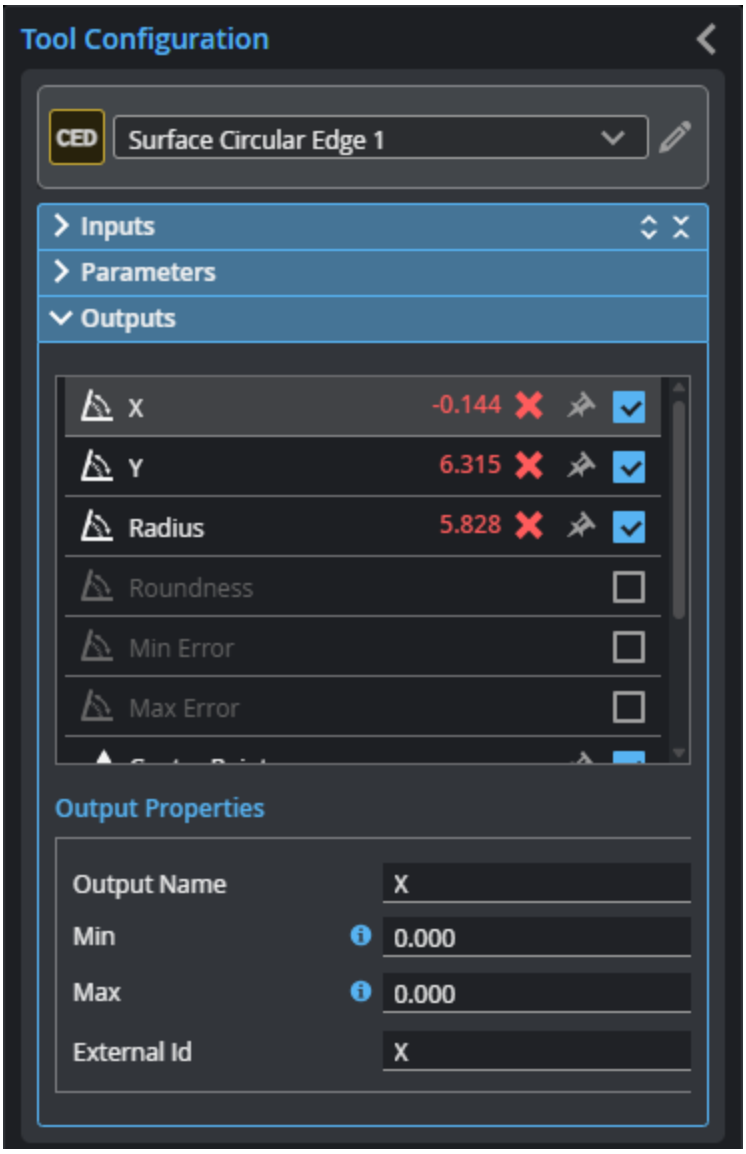
Parameter	Description
Relative Threshold	The value for the relative threshold. The tool calculates a relative threshold by scaling the greatest height or intensity difference found on the path profiles by the percentage in **Relative Threshold**. This lets you configure the tool without knowing the actual 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).
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.

Parameter	Description
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**. 
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 the part detection of an object sitting on a flat background (for more information on part detection, see *Profile Part Detection* on page 603). 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.

Parameter	Description
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.
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.
Which Circle	Tells the tool which circle (Inner or Outer) to use when Minimum Zone Circle is the fit method in **Fit Type**.

Outputs

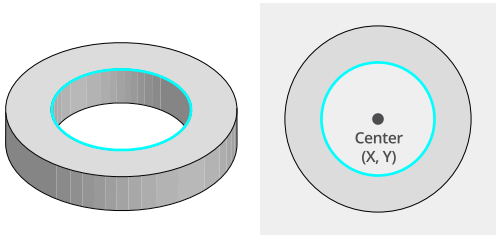
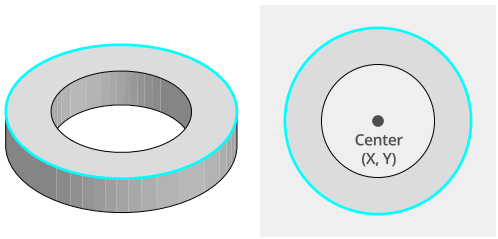
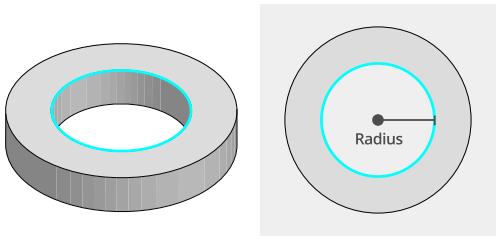
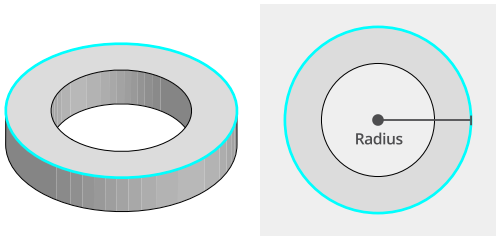
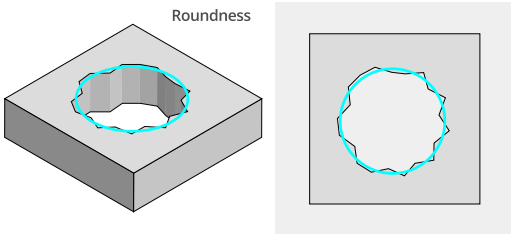
Most tools provide measurements, geometric features, or data as outputs.

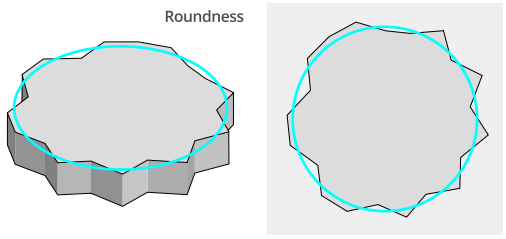


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurement	Illustration
X **Y** Returns the X and Y position of the center of the fitted circle, respectively.	 
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**.	

Measurement	Illustration
	
Min Error **Max Error** These measurements return information on the points furthest inside and outside the fitted circle, respectively.	

Features

Type	Description
Center Point	The center of the fitted circle.
Circle	The fitted circle.
Min Error Point	The point of minimum error which is furthest inside of the circle.
Max Error Point	The point of maximum error which is furthest outside of the circle.

For more information on geometric features, see *Geometric Features* on page 227.

Surface Countersunk Hole

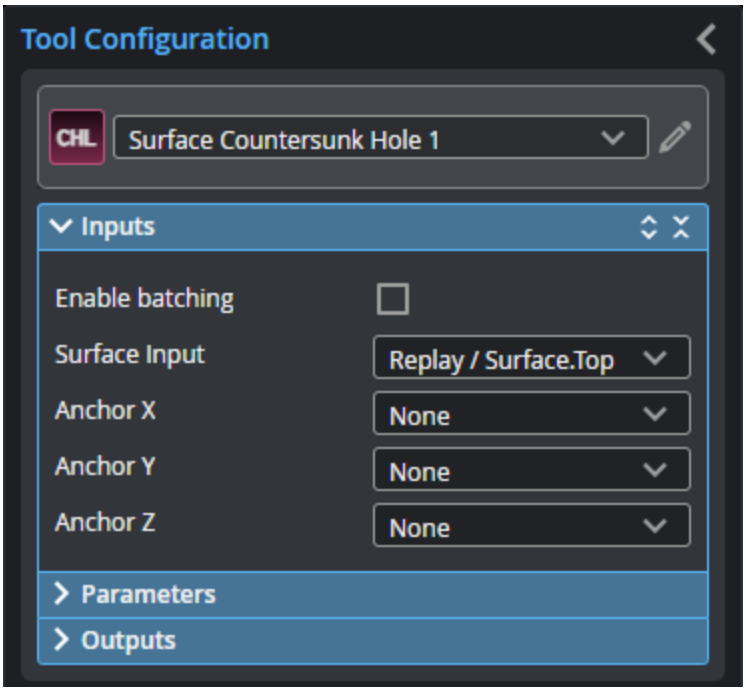
The Countersunk Hole tool locates a countersunk circular opening within a region of interest on the surface and provides measurements to evaluate characteristics of countersunk holes, including the position (X, Y, and Z) of the center of the hole, outside radius of the hole, hole bevel angle, and the 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.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



> To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

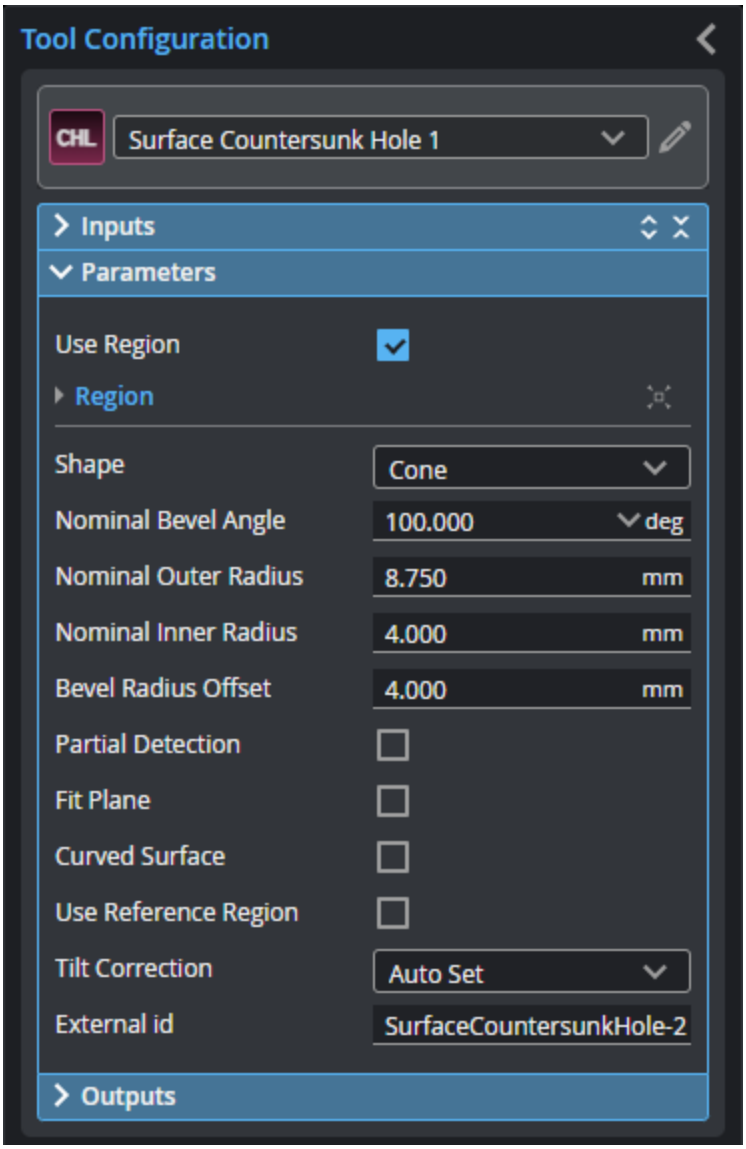
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array,

Name	Description
	other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X	Positional anchors are optional.
Anchor Y	
Anchor Z	

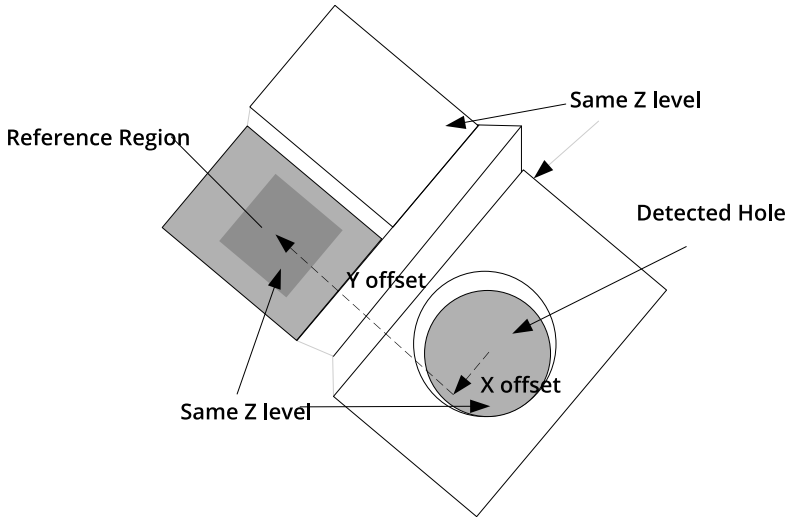
Parameters

You configure the tool's parameters in the expandable **Parameters** section.



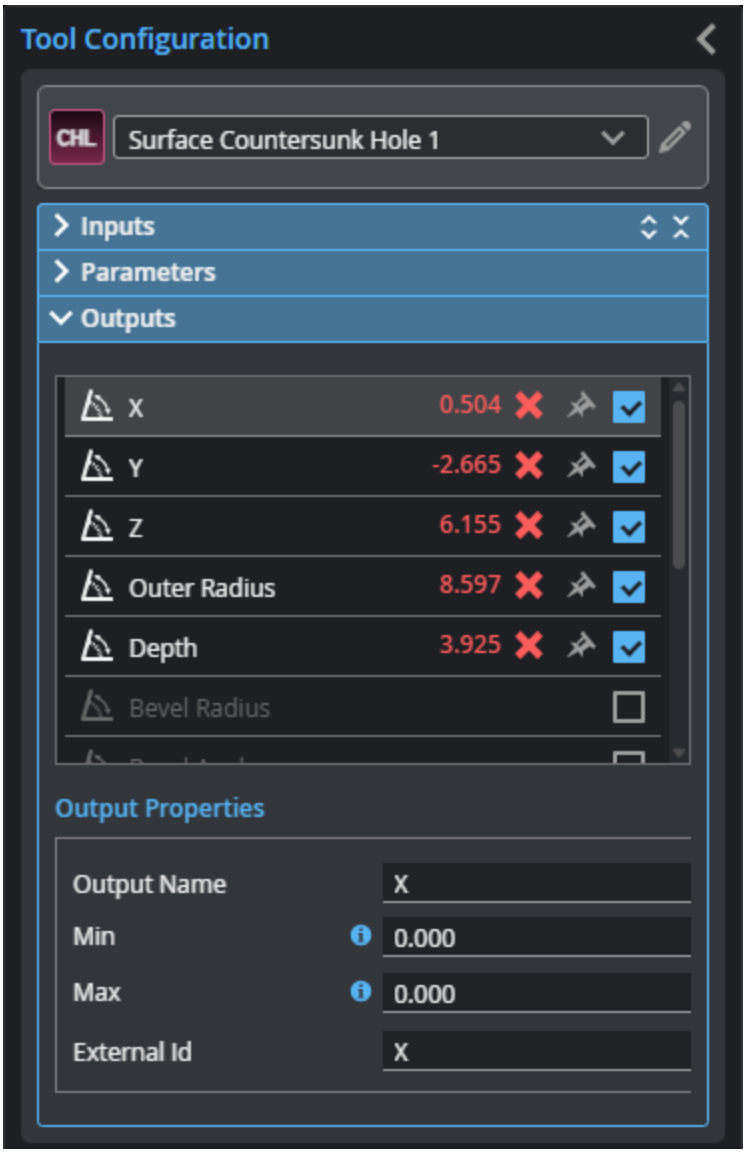
Parameters

Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
Shape	The shape of the countersunk hole. (See illustrations below.) 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.
Fit Plane	When enabled, excludes data beyond the distance from the plane surrounding the hole, using the value you provide in the **Plane Fit Range** parameter. 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.
Plane Fit Range	The distance from the plane surrounding the hole beyond which data is excluded.
Curved Surface	Whether the surface that the countersunk hole is on is curved.
Curve Orientation	The orientation of the curvature in degrees. Only visible when **Curved Surface** is enabled.
Use Reference Region	When enabled, displays settings you use to configure reference regions.
Reference Type	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.

Parameter	Description
	 When this option is set to **Autoset**, the algorithm automatically determines the reference region. You can also set this option to one or two regions, which you manually set. Note that the location of manual reference regions 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.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

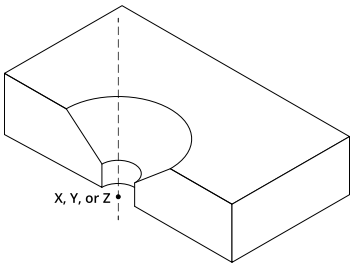
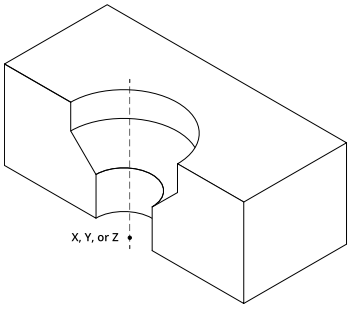
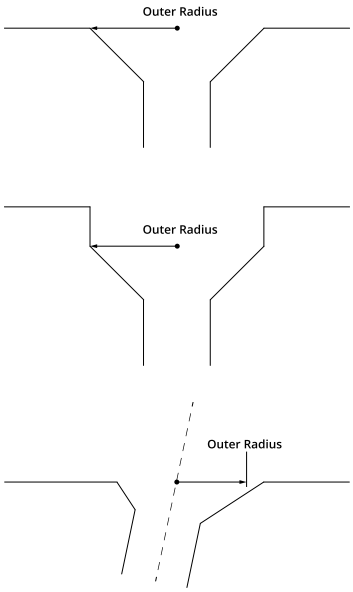
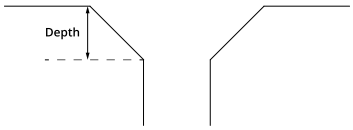
Most tools provide measurements, geometric features, or data as outputs.

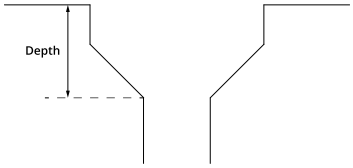
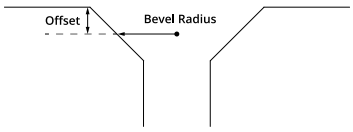
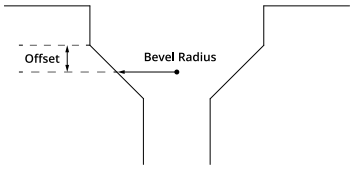
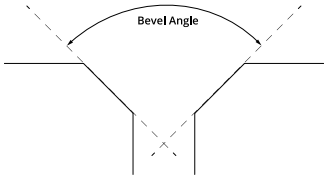
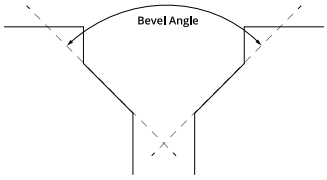
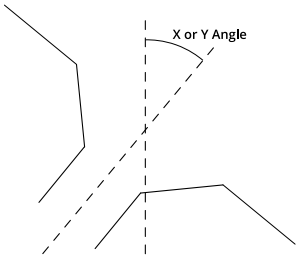


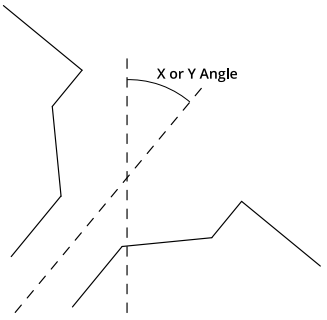
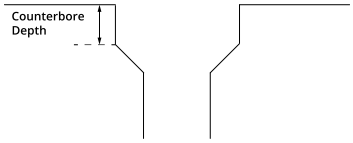
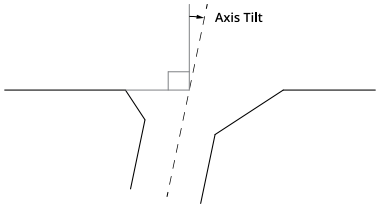
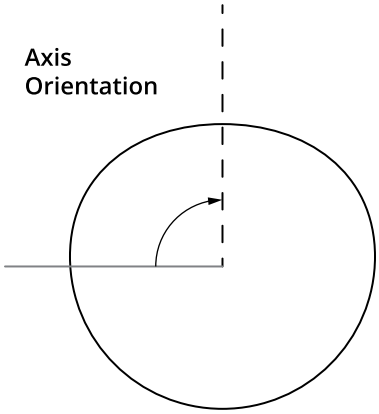
To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.	
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, use an instance of the Measurement Formula too with the Formula field set to M1*2, for example. For more information, see *Measurement Formula Tool* on page 489.	
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 (**Bevel Radius Offset** setting) relative to the surface that the countersunk hole is on. To convert the radius to a diameter, use an instance of the Measurement Formula too with the Formula field set to M1*2, for example. For more information, see *Measurement Formula Tool* on page 489.	 
Bevel Angle Determines the angle of the hole's bevel.	 *Cone*  *Counterbore*
X Angle Determines the angle the hole relative to the X axis.	 *Cone*

Measurement	Illustration
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**.	
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.

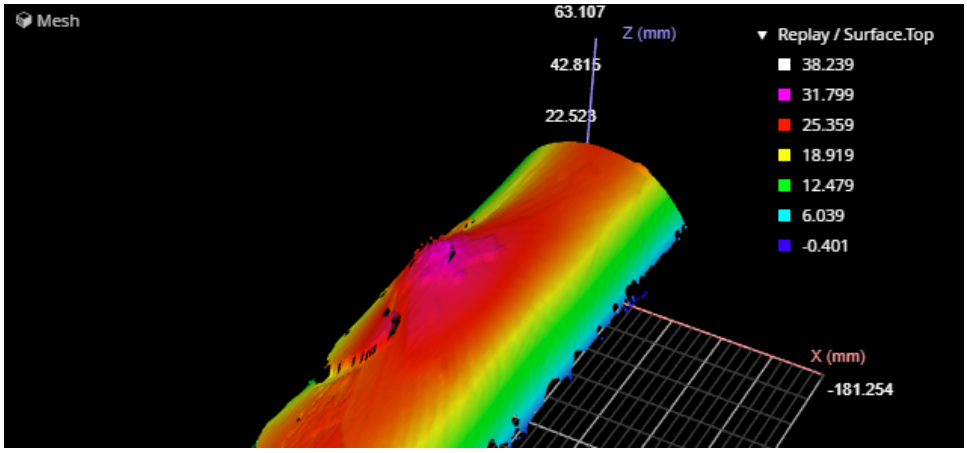
For more information on geometric features, see *Geometric Features* on page 227.

Surface Curvature Correction

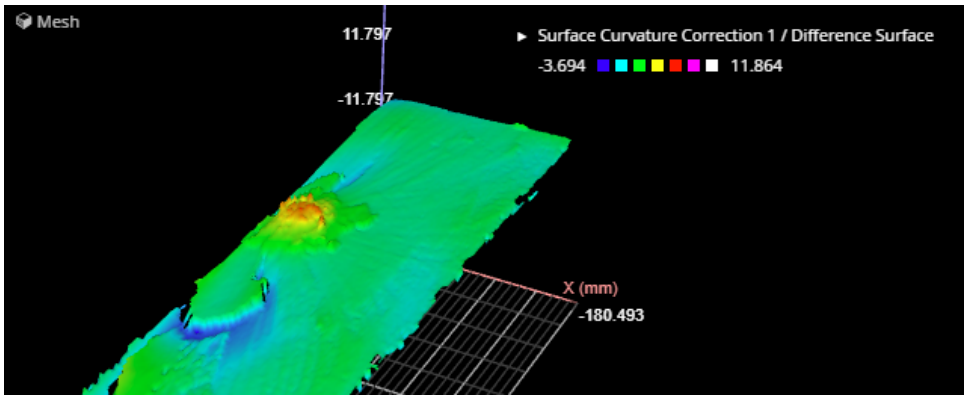
The Surface Curvature Correction tool removes curvature from curved surfaces while preserving surface features or defects, using a configurable polynomial order (the tool performs a 2D polynomial fit on X and Y to process surfaces). You can then use the tool's output as input to other measurement tools to 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.

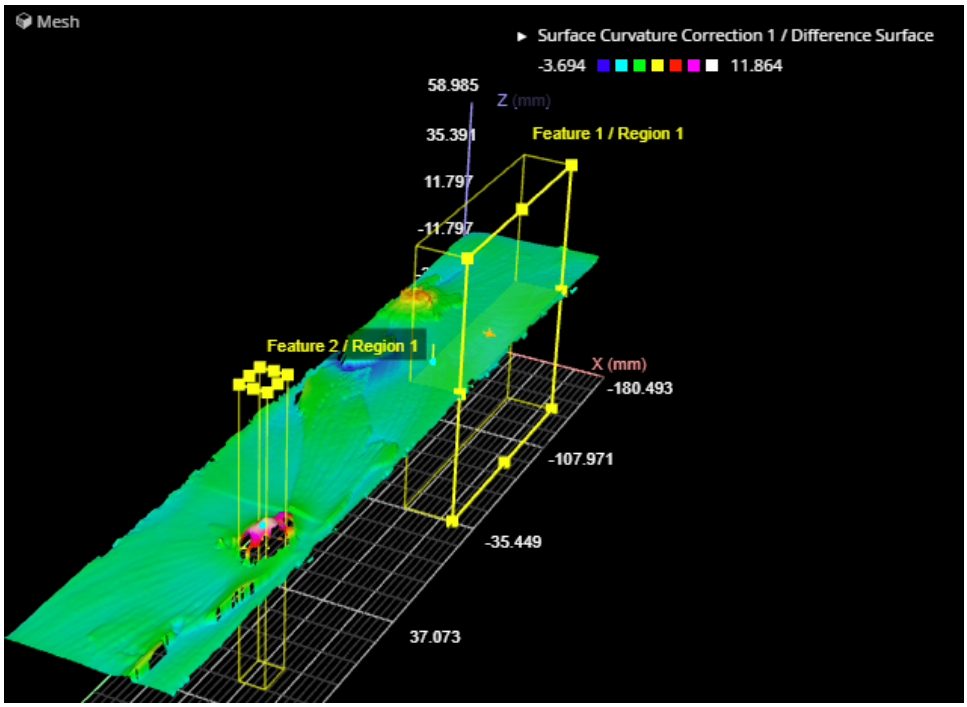


The original, curved scan of a target.



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:

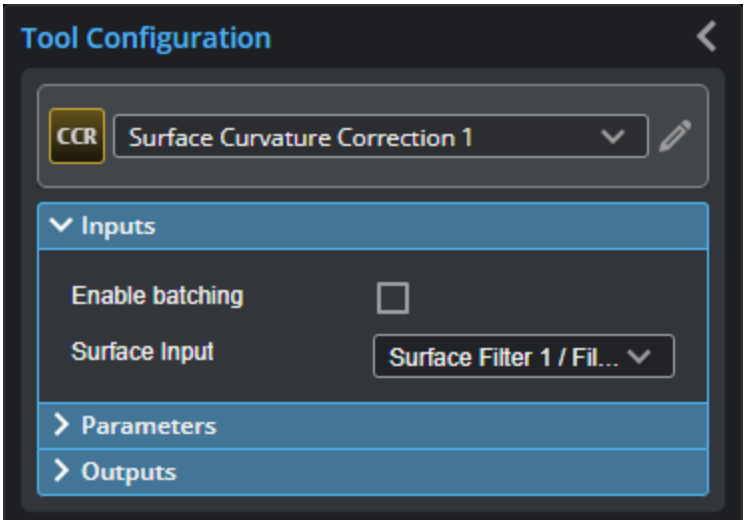


Height of a raised feature relative to the flattened surface.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



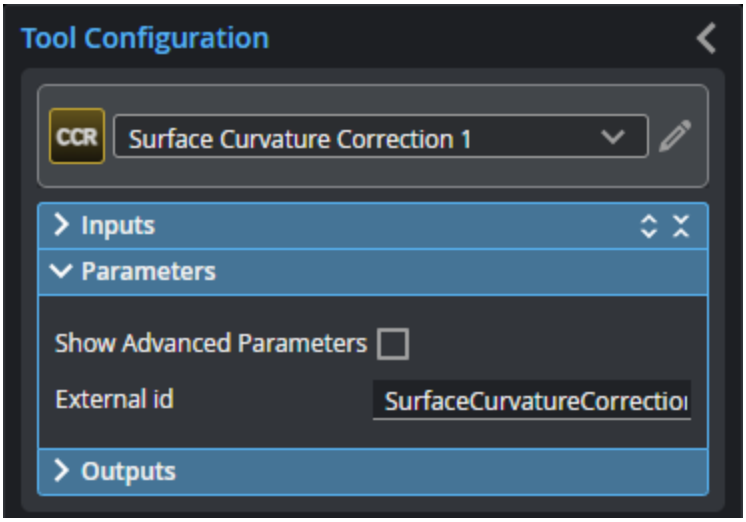
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes

Name	Description
	each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

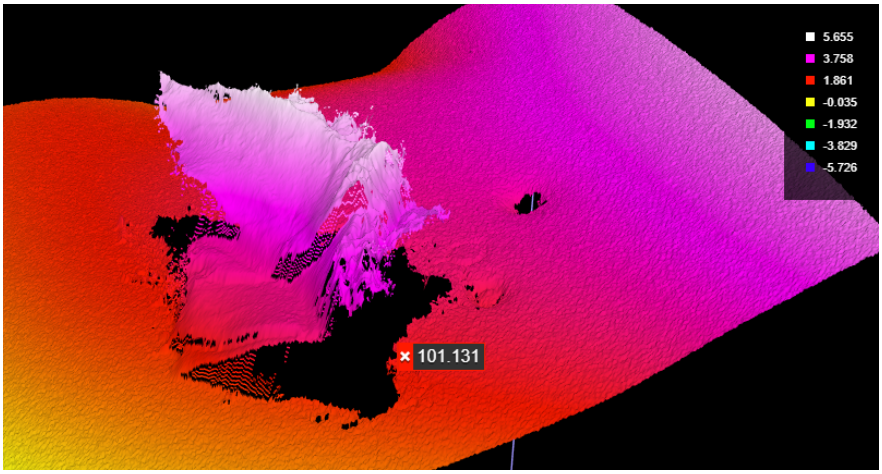
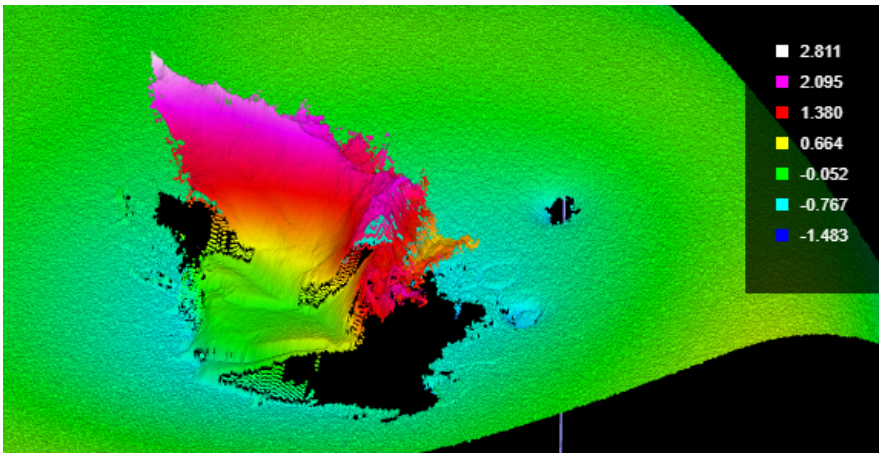


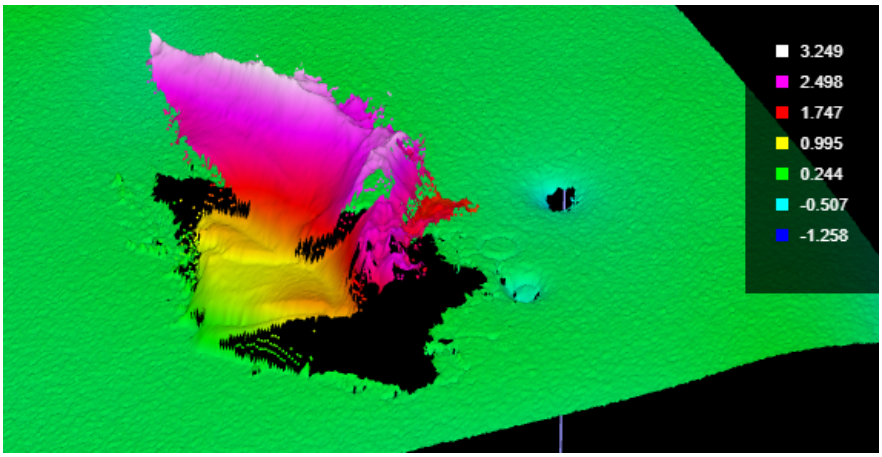
Parameters

Parameter	Description
Show Advanced Parameters	Enables a set of advanced parameters. (See *Advanced Parameters* below.)
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Advanced Parameters

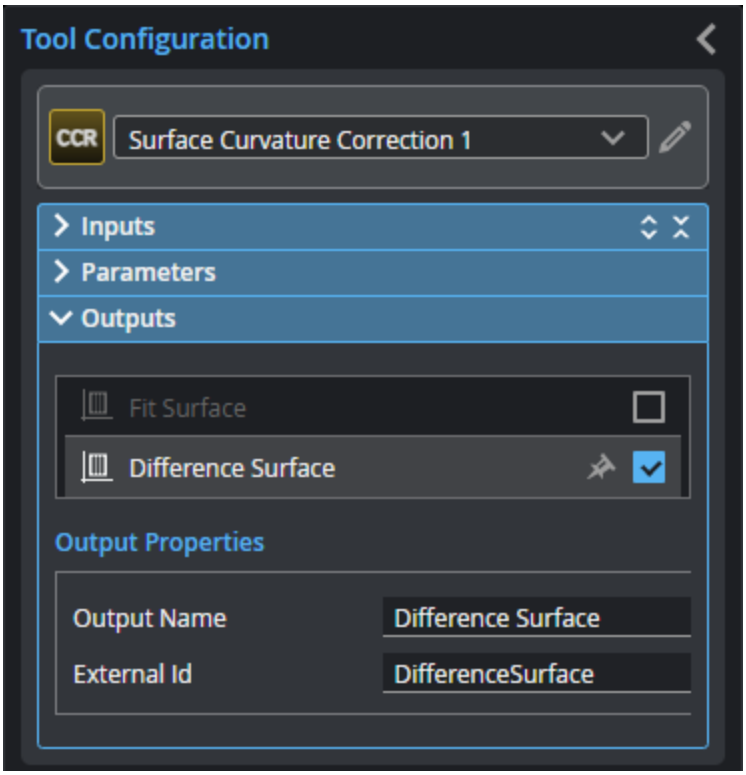
Parameter	Description
Number of Regions	Lets you specify and configure one or more regions the tool will process. Use this parameter to limit the tool to specific areas on the target.
Exclude Features	Lets you exclude features or surface details from the polynomial fit. This can allow you to get a better 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

Parameter	Description
	surface.  If the large feature to the left is *not* excluded for the polynomial fit, 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
	
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.
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.
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 bottom up (**Negative area**) and from the top down (**Positive area**), respectively.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

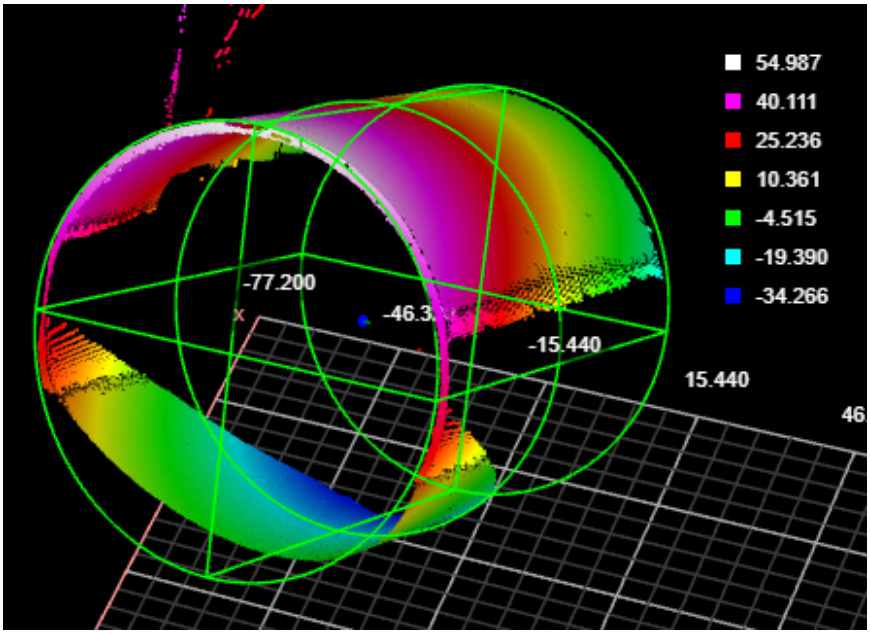


Data

Type	Description
Fit Surface	The fitted polynomial the tool uses to flatten the original surface. Use this as a diagnostic to better fit the polynomial.
Difference Surface	The “flattened” surface: this is the original surface with the fitted polynomial removed.

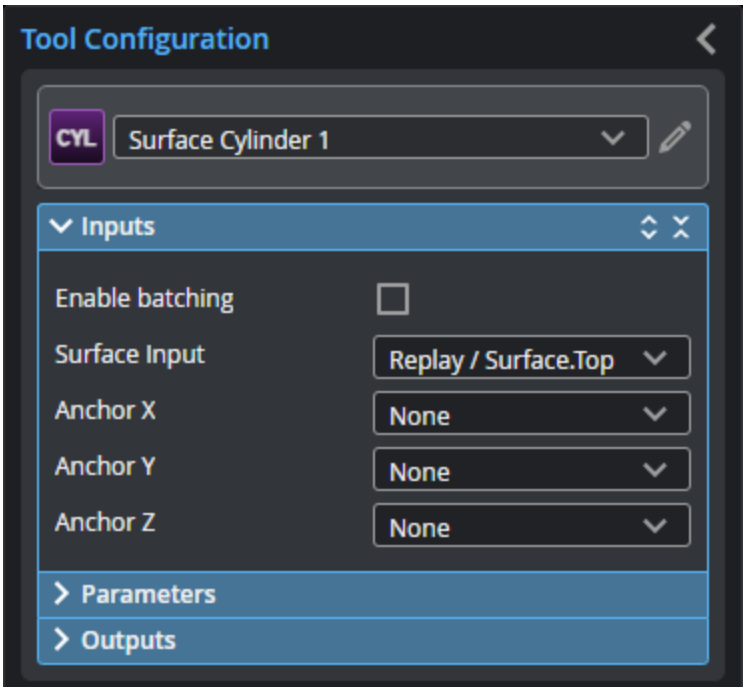
Surface Cylinder

The Surface Cylinder fits a cylinder to scan data and returns measurements and geometric features related to the fitted cylinder. Unlike the Surface Stud tool, the Surface Cylinder tool does not rely on a flat surface perpendicular to the cylindrical object. Examples uses of the tool include fitting to the outside of a cylinder and fitting inside a drilled hole.



Inputs

You configure the tool's inputs in the expandable **Inputs** section.

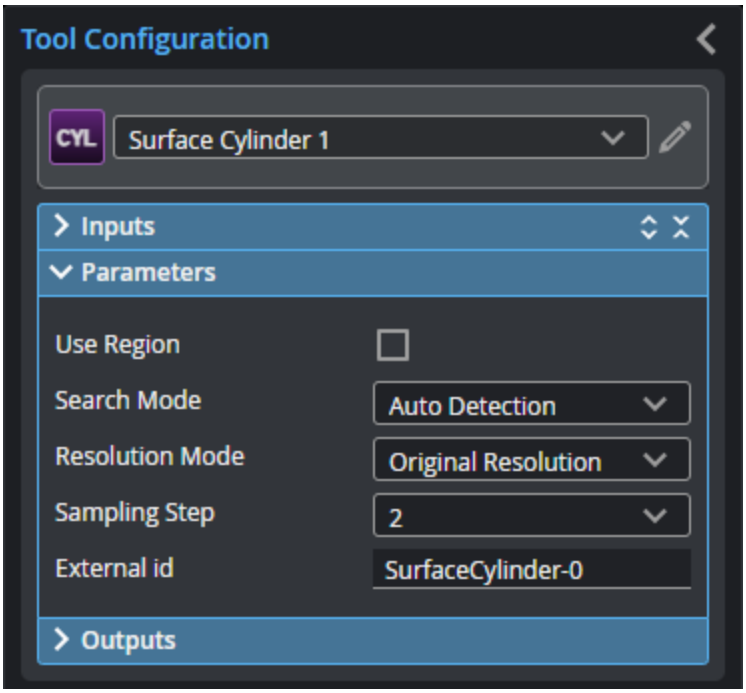


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

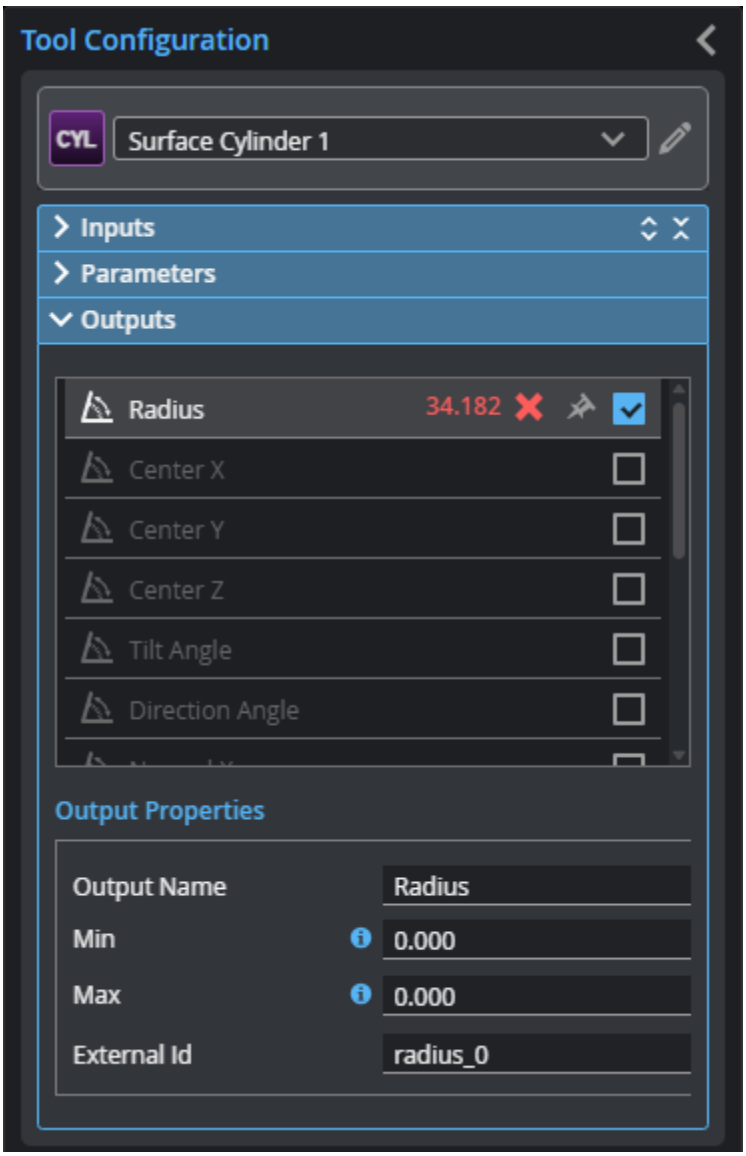


Parameters

Parameter	Description
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
Search Mode	Indicates the expected orientation of the cylindrical target's axis around the Z axis.

Parameter	Description
	One of the following: **Auto Detection** – The cylindrical target can be in any orientation. **Axis in X Direction** or **Axis in Y Direction** – The cylindrical target's axis is expected to be roughly parallel to the X or the Y axis, respectively. Variation typically must be less than +/- 3 or 4 degrees. **Advanced Detection** – As with Auto. Detection, the cylindrical target can be in any orientation. This mode can handle some complex scenarios, such as when trying to fit a cylinder to the inside of a hole. Processing time is greater with this search mode.
Resolution Mode	On G3 sensors, leave this set to the default **Original Resolution**. Determines whether the tool scales the X or Y resolution so that they are the same (a 1:1 ratio), or leaves the X and Y resolutions as the original. One of the following. • **Optimal (uniform)** Brings the X/Y resolution ratio to 1:1 while preserving the pixel area. Best for random rotation around Z. Provides a balance between the highest and lowest possible resolutions, requiring an average amount of memory and processing time compared to the **High Oriented (uniform)** or **Low Oriented (uniform)** options. • **High Oriented (uniform)** Interpolates the lower resolution to match the higher resolution (between X and Y) in the input. Choose this option when increased resolution is preferred over speed and low memory usage. (This can result in a very high resolution output, creating a lot of data for subsequent tools to process. This can in turn result in slower processing.) • **Low Oriented (uniform)** Decimates the higher resolution to match the lower resolution (between X and Y) in the input. Choose this option when speed and low memory usage is preferred over resolution. (It can result in significant data quality reduction with large Z rotations if the X and Y resolutions of the input are very different.) • **Original Resolution** Keeps the original X and Y resolution of the scan. Use this option only when you expect little or no Z rotation. Otherwise, with X/Y resolution ratios that are not 1:1, large rotation around Z results in severe data quality reduction.
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.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement
Radius Returns the radius of the fitted cylinder.
Center X **Center Y** **Center Z** The X, Y, and Z position of the center of a circle place in the middle of the fitted cylinder
Tilt Angle The angle of the cylinder relative to the XY plane. A cylinder parallel to the XY plane has an angle of 90 degrees.
Direction Angle The angle of the cylinder's axis around the Z axis. An angle of 0 degrees is parallel to the X axis.
Normal X **Normal Y** **Normal Z** These measurements return the X, Y, and Z components of the direction vector of the cylindrical target.

Features

Type	Description
Point	A point representing the center of a circle at the midpoint of the fitted cylinder
Line	A line representing the axis of the fitted cylinder.

For more information on geometric features, see *Geometric Features* on page 227.

Data

Type	Description
Fit Surface	The Surface data the tool uses to fit the cylinder (for example, excluding data outside the defined region). Use this output, for example, to confirm that the tool is using the correct part of the target to fit the cylinder.
Difference Surface	Deviations in the Surface from the fitted cylinder. Use this output, for example, to evaluate if blemishes or distortions in the Surface data are preventing the tool from properly fitting the cylinder.

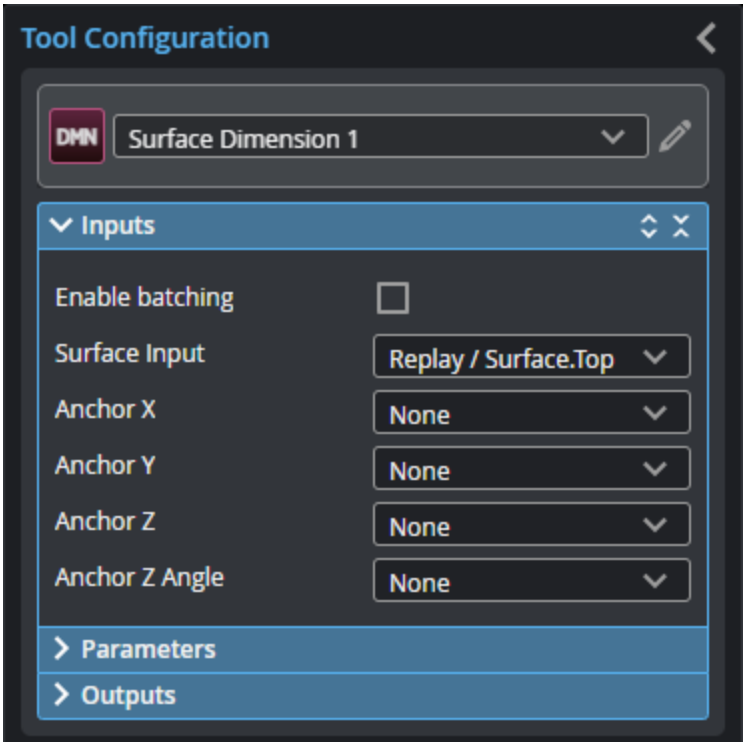
Surface Dimension

The Dimension tool returns various dimensional measurements of a part. You must specify two feature types (see below).

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

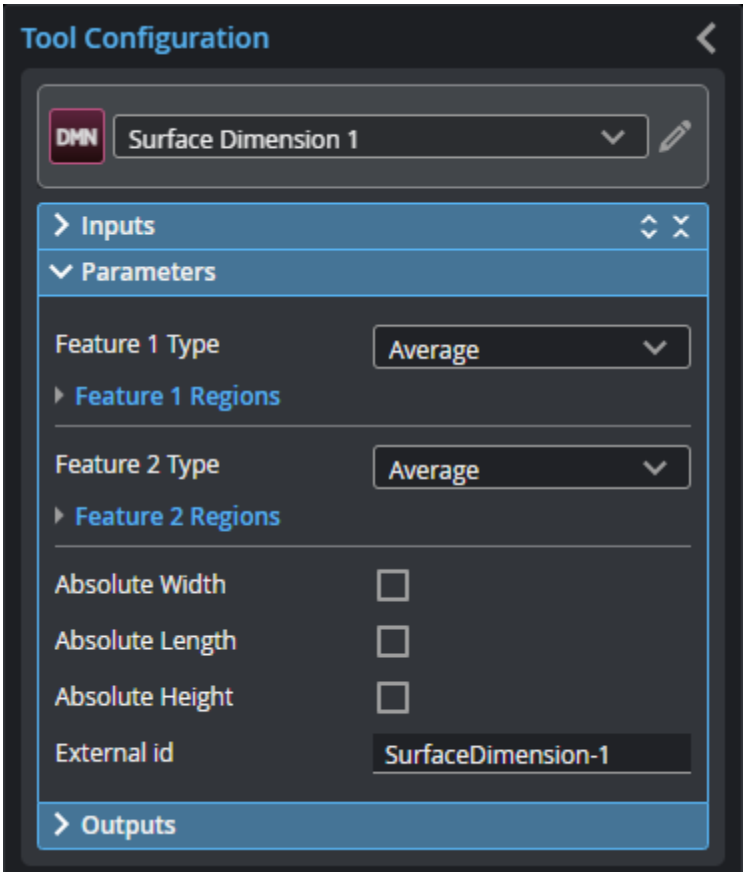
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X	Positional anchors are optional.

Name	Description
Anchor Y	
Anchor Z	
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



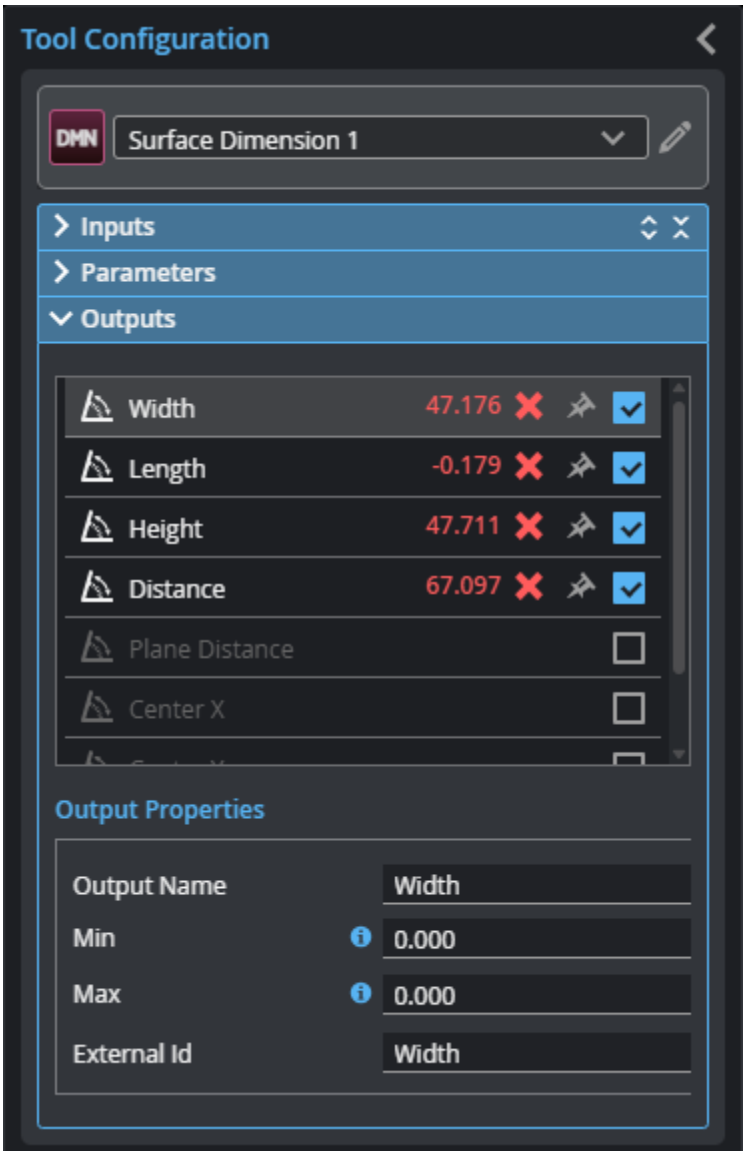
Parameters

Parameter	Description
Feature 1 Type Feature 2 Type	The feature point types the tool uses. For each, one of the following (for more information, see *Feature Points* on page 224): • Average • Median • Centroid • Max X • Min X

Parameter	Description
	• Max Y • Min Y • Max Z • Min Z When more than one point is at minimum Y or maximum Y (and the feature is set to Min Y or Max Y, respectively), the rightmost point (the one at greater positive X) is selected by the tool. When more than one point is at minimum X or maximum X (and the feature is set to Min X or Max X, respectively), the bottom point (the one at greater positive Y) is selected by the tool.
Feature 1 Regions Feature 2 Regions	When these are expanded, they display the region- and mask-related settings associated with the two features the tool uses to perform measurements. The feature parameters associated with these regions are **Feature 1** and **Feature 2**, respectively; for more information, see below.
Enable	When enabled, the region- and mask-related settings are displayed.
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Absolute Width Absolute Length Absolute Height	Enable these parameters to return an absolute value, regardless of the relative positions of the first and second regions of interest. (Dimensions are calculated by subtracting the position of the feature point in region 1 from that of the feature point in region 2.)
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

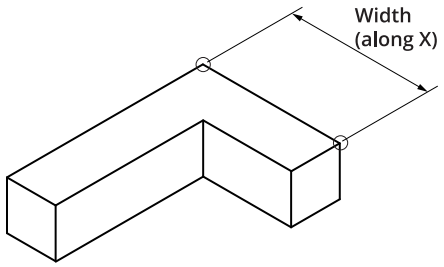
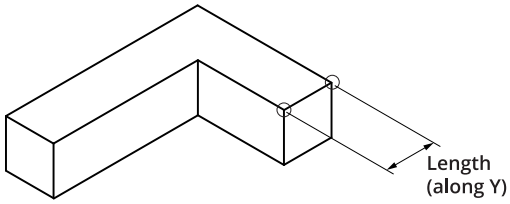
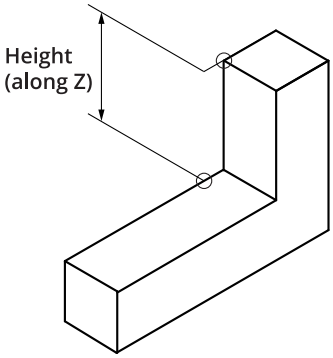
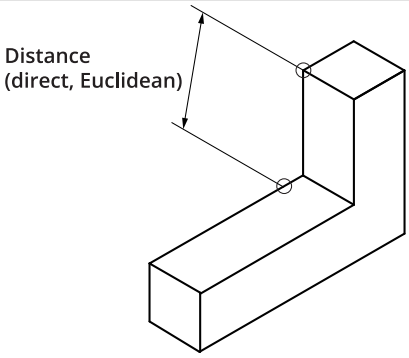
Most tools provide measurements, geometric features, or data as outputs.

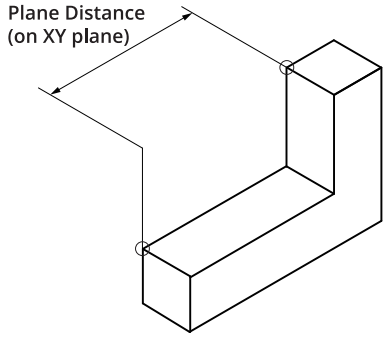
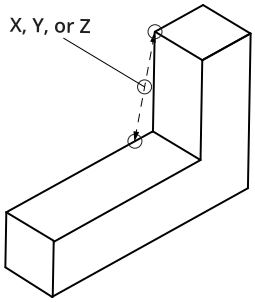


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

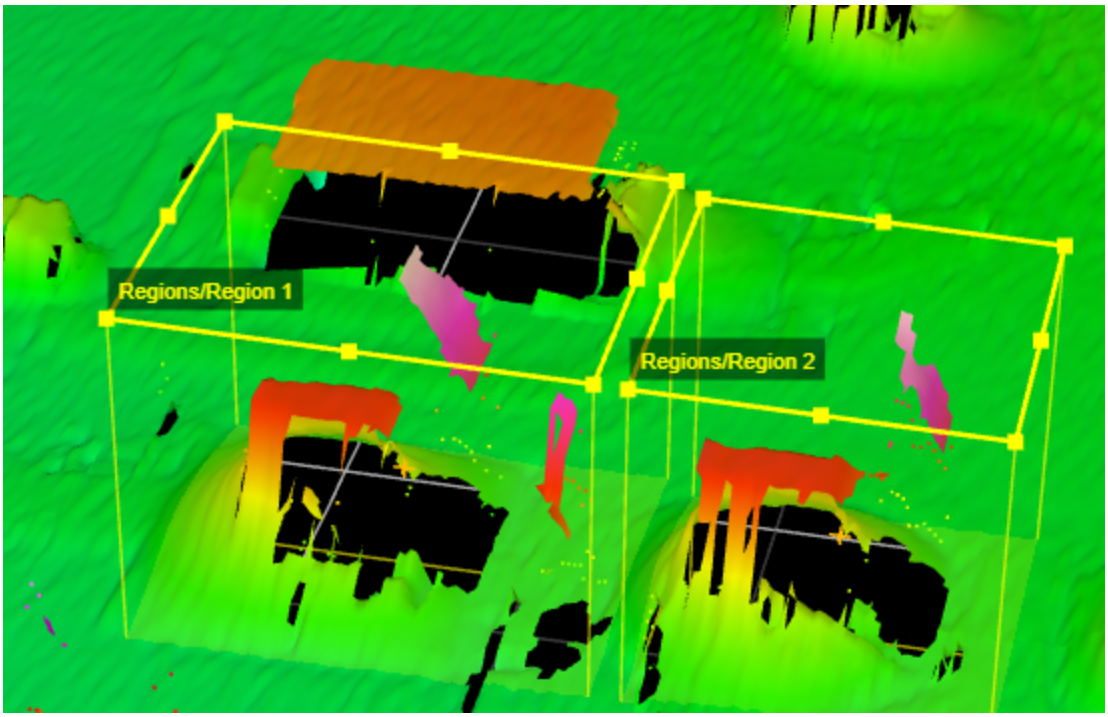
Measurement	Illustration
Width Determines the distance between the selected features along the X axis.	
Length Determines the distance between the selected features along the Y axis.	
Height Determines the distance between the selected features along the Z axis.	
Distance Determines the direct, Euclidean distance between the selected features.	

Measurement	Illustration
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 **Center Y** **Center Z** These measurements determine the X, Y, or Z position, respectively, of the center point between the selected features.	

Surface Direction Filter

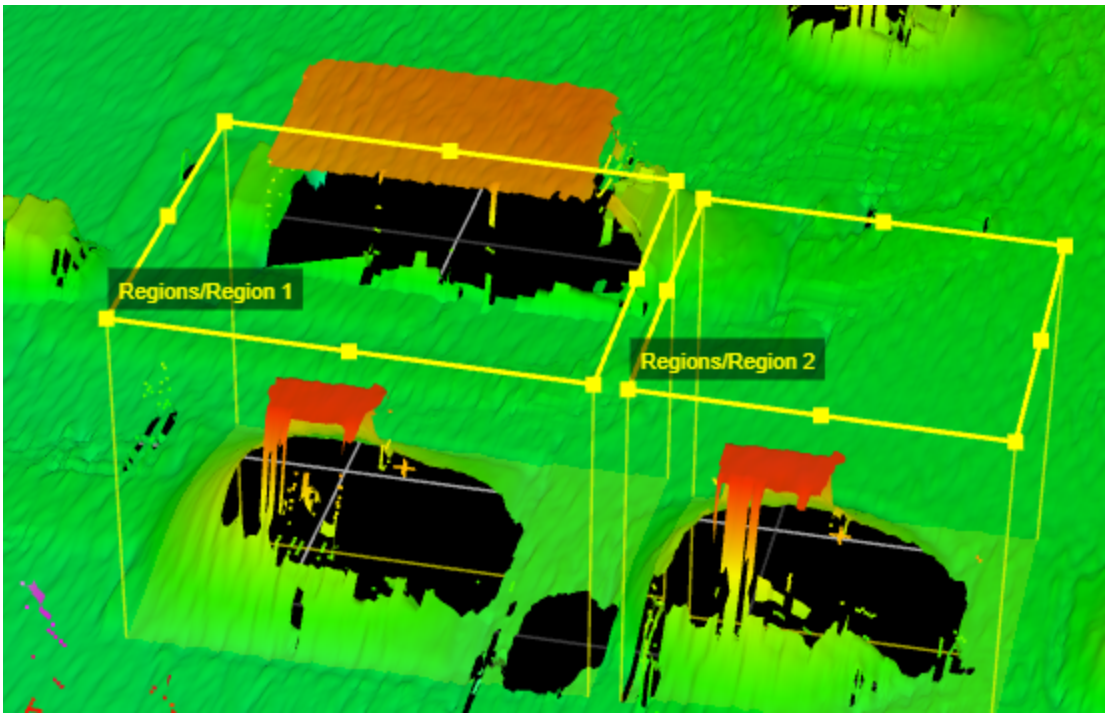
The Surface Direction Filter helps exclude unwanted data points based on their “orientation” (the polar angle relative to surrounding data points) in 3D space, for example, data points resulting from reflections. The tool can provide better results than median or height based filters. The tool lets you define up to 16 regions to which the filtering is applied.

For example, in the following scan data, noise appears to the right of two surface mount components on a PCB (pink areas). In this case, the "direction" 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.



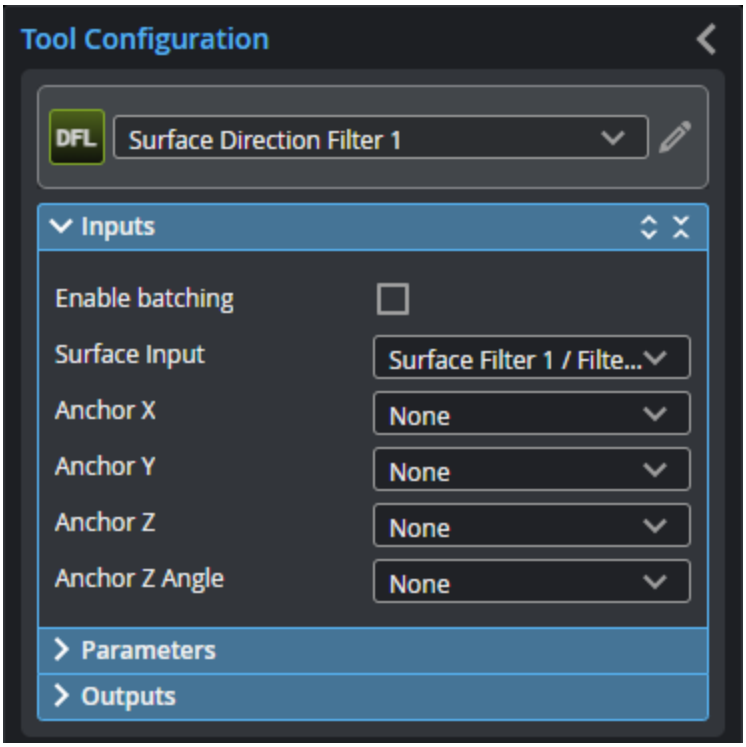
Surface after direction filtering.

Note that the tool's filtering parameters apply to all enabled regions. To apply different filtering parameters, you can add additional instances of the tool and chain them together, setting the input of subsequent instances to the output of the preceding instance.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



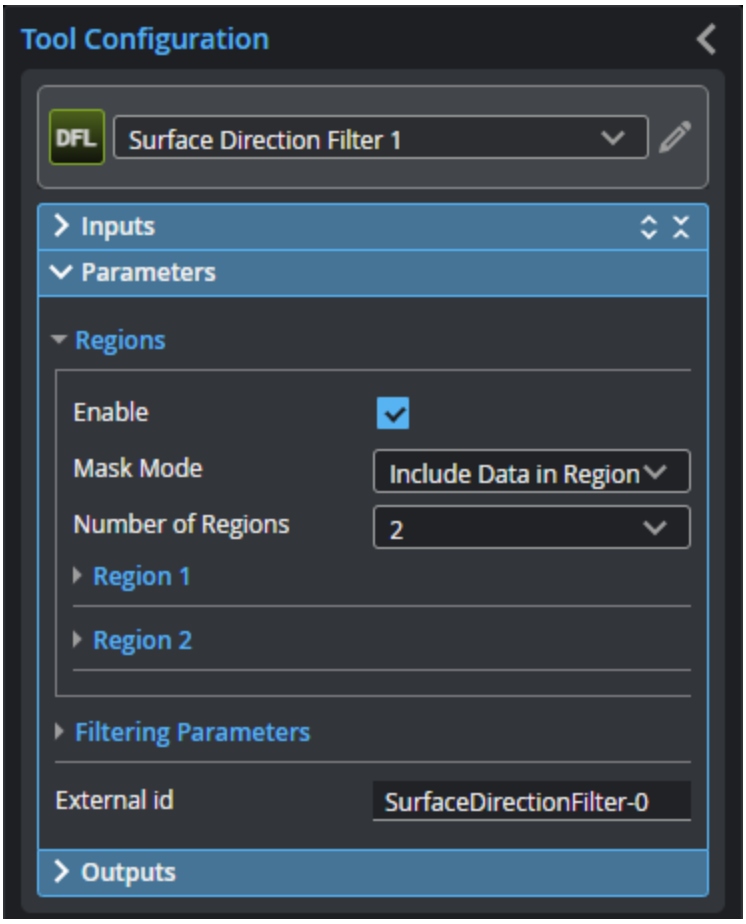
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

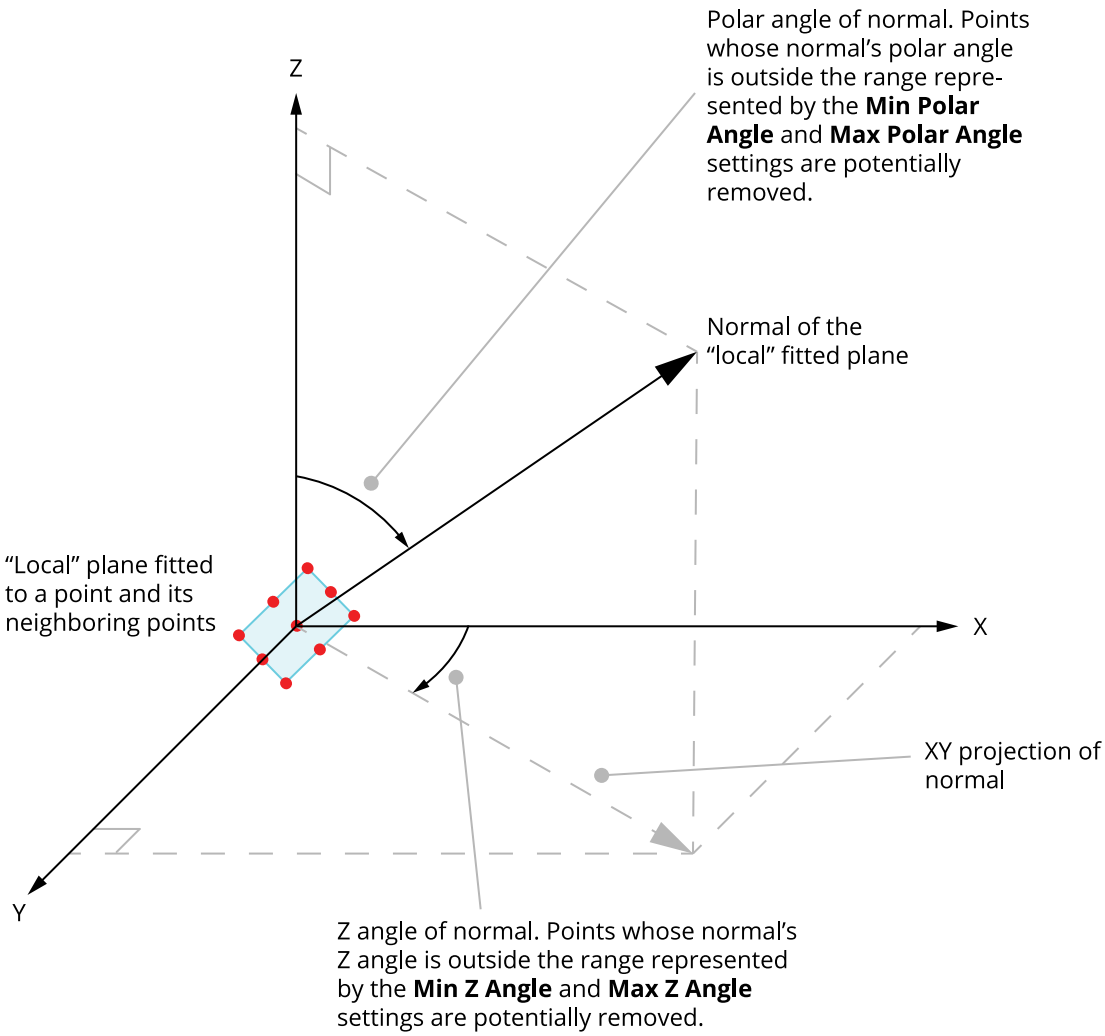
Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



The following illustrates the angle parameters that control which data points are excluded in scan data:



The number of neighboring points shown above is for illustrative purposes only.

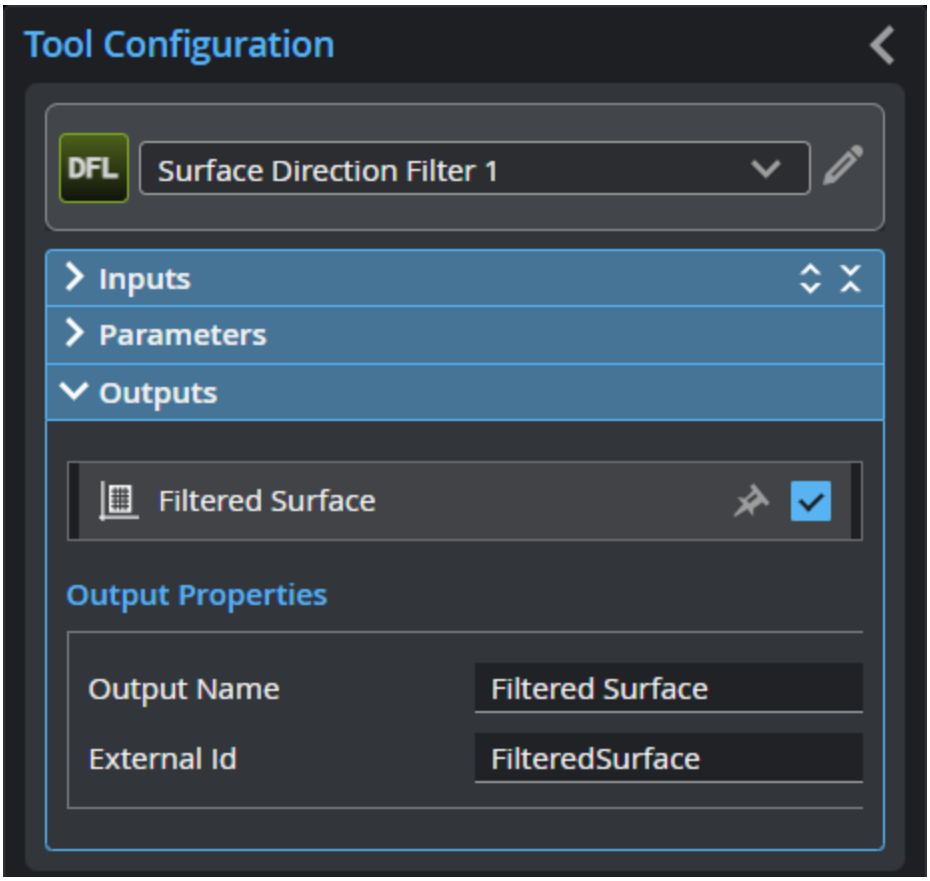
Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Filtering Parameters	When expanded, displays the filtering parameters the tool uses.
Min Z Angle 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.

Parameter	Description
Min Polar Angle 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.
Smooth Size	A mean filter applied to the surface data before calculating the normals in order to avoid abrupt normal changes due to noise.
Noise Removal	Eliminates noise that can be introduced by the tool's normal calculation.
Preserve Data Outside Region	When enabled, data outside the regions is preserved in the output. Otherwise, only the data in the regions is output.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Data

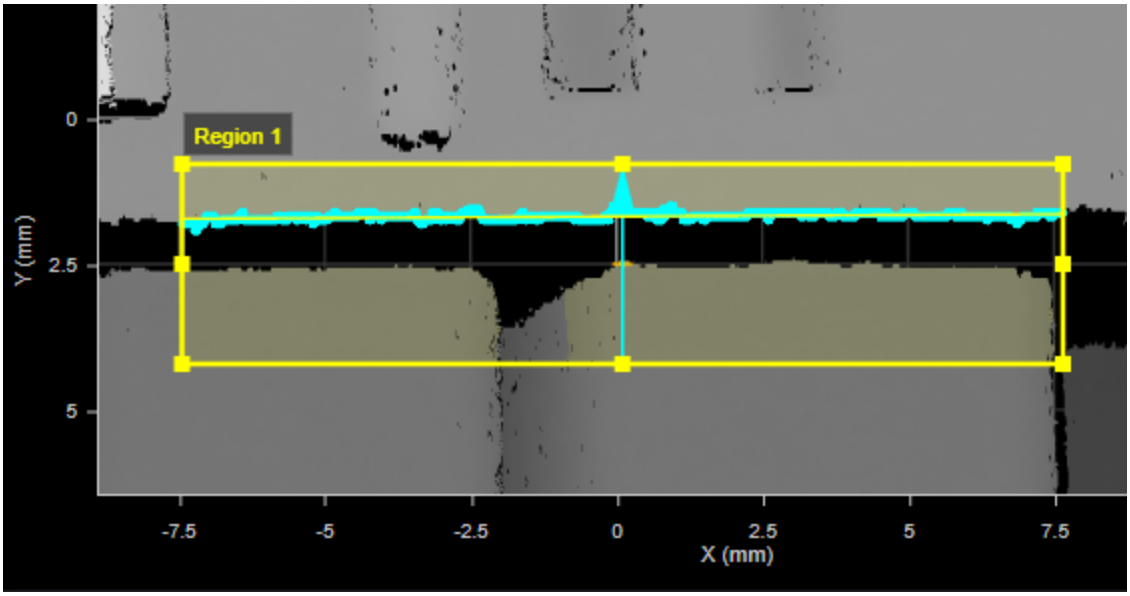
Type	Description
Filtered Surface	The surface after filtering.

Surface Edge

The Edge tool fits a line to a straight edge in the scan data, using either height map or intensity data. The tool can search for an edge using either a step (an abrupt change in the data) or a corner (a contiguous change in the shape of surface). 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 209.

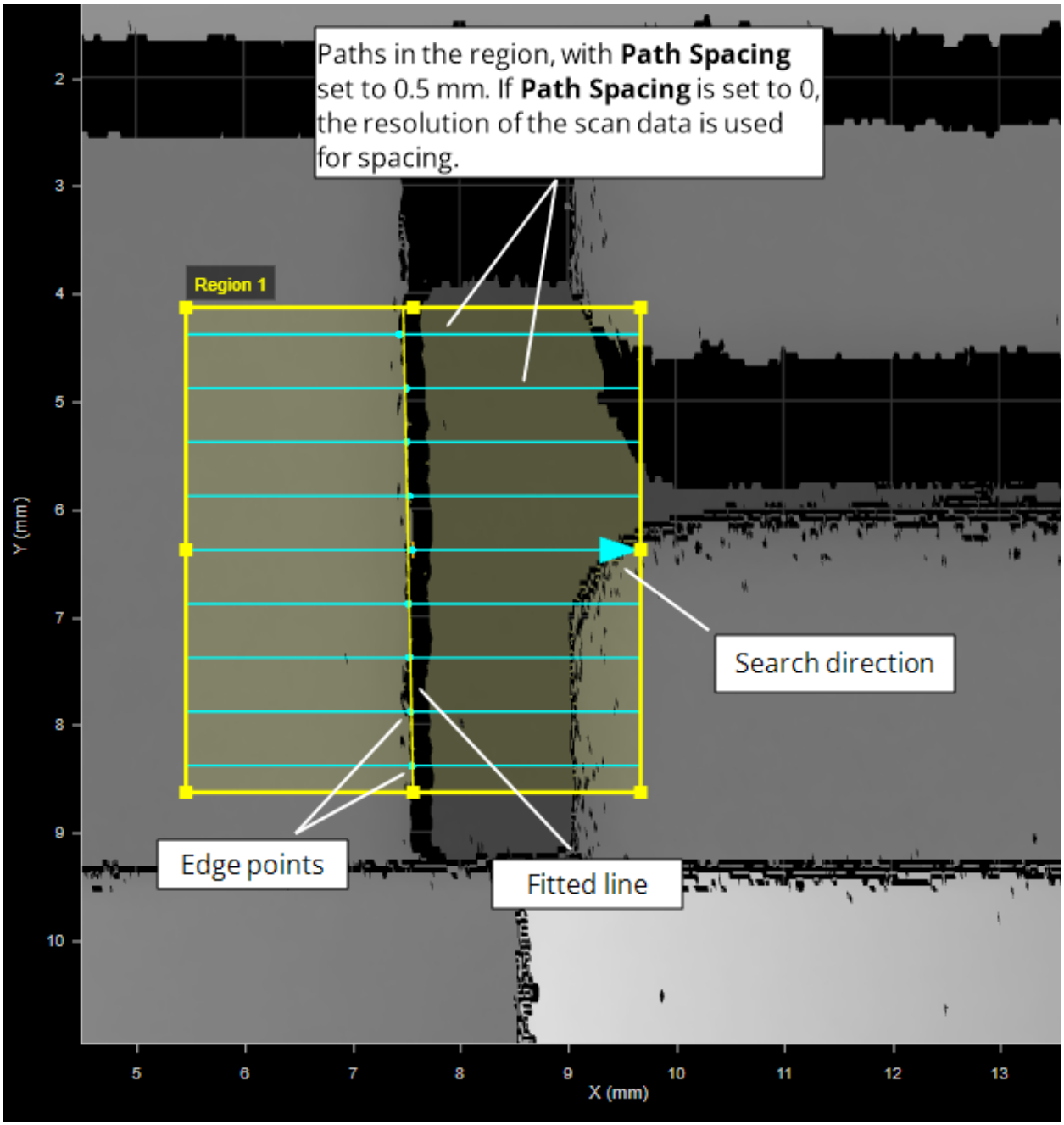
The tool can also generate edge line and center point geometric features that Feature tools can take as input for measurement. For more information on Feature tools, see *Feature Tools* on page 252.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

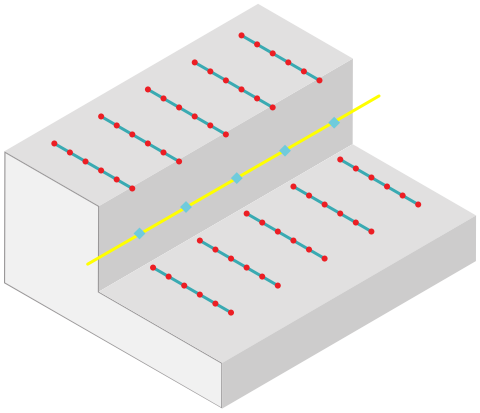
Paths and Path Profiles

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. (Note that the paths are only displayed if **Path Spacing** is greater than 0.)



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).



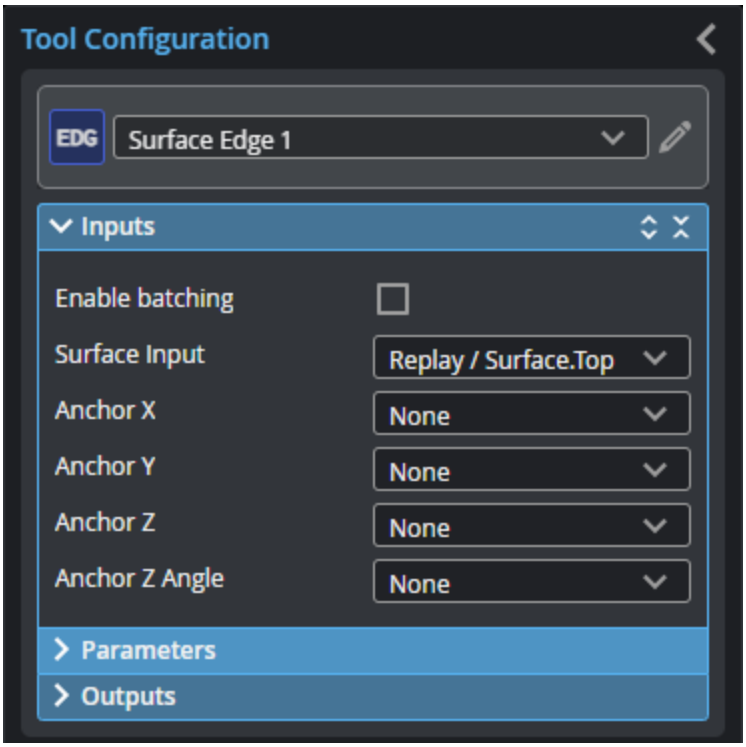
A single path profile extracted from a path.



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.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



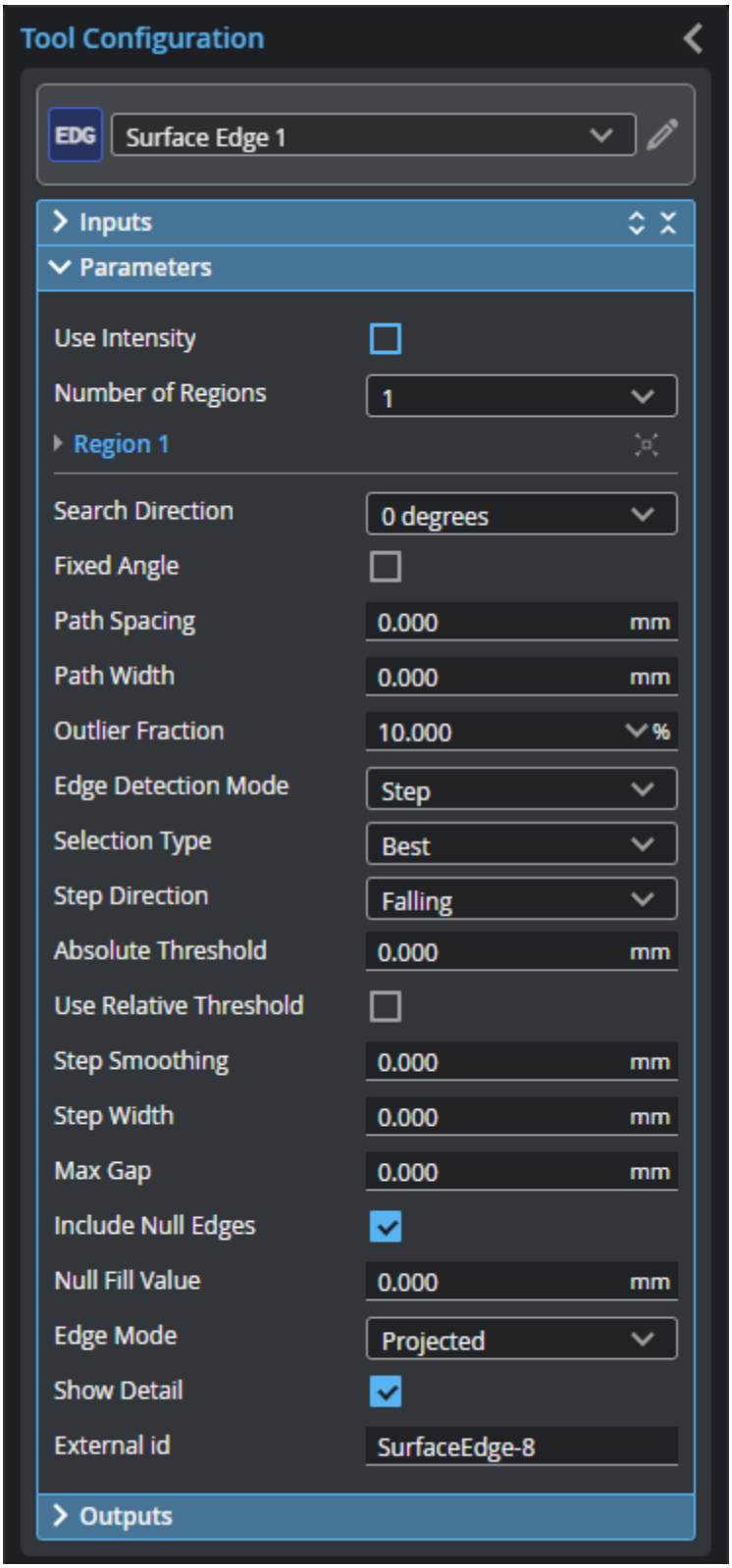
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

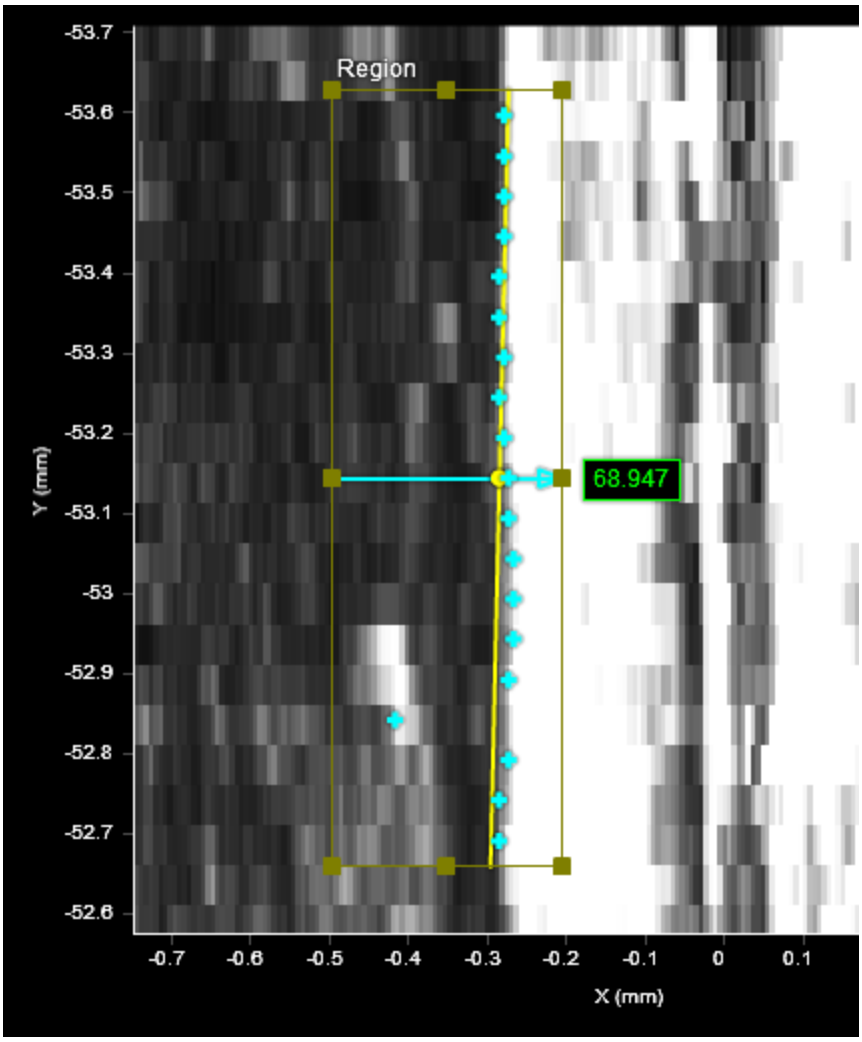
Parameters

You configure the tool's parameters in the expandable **Parameters** section.

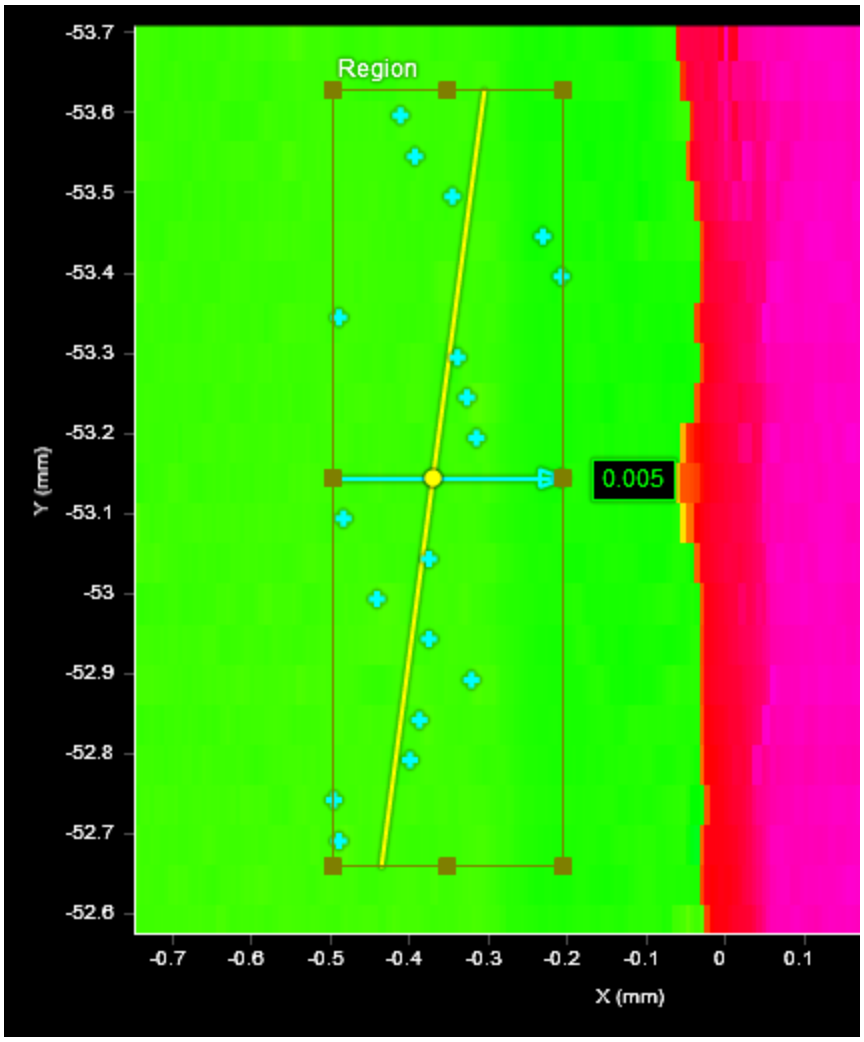

Tool Configuration
EDG
Surface Edge 1
Inputs
Parameters
Use Intensity
Number of Regions
1
Region 1
Search Direction
0 degrees
Fixed Angle
Path Spacing
0.000
mm
Path Width
0.000
mm
Outlier Fraction
10.000
%
Edge Detection Mode
Step
Selection Type
Best
Step Direction
Falling
Absolute Threshold
0.000
mm
Use Relative Threshold
Step Smoothing
0.000
mm
Step Width
0.000
mm
Max Gap
0.000
mm
Include Null Edges
Null Fill Value
0.000
mm
Edge Mode
Projected
Show Detail
External id
SurfaceEdge-8
Outputs

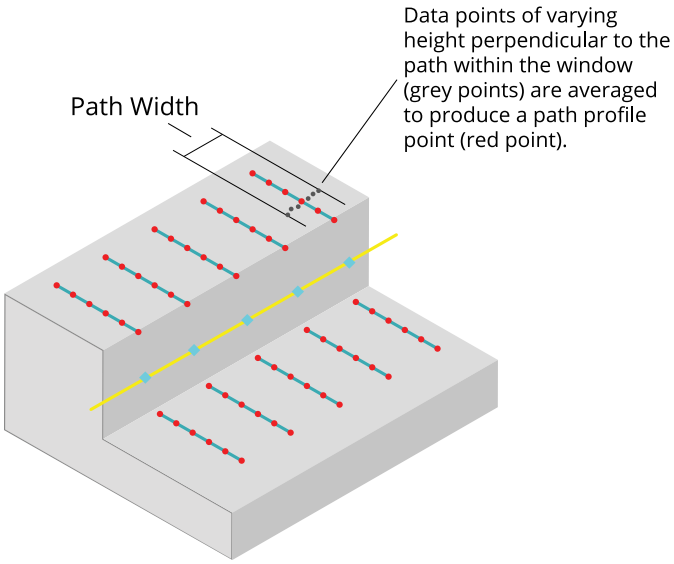
Parameters

Parameter	Description
Use Intensity	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 214. 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).
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

Parameter	Description
	roughly perpendicular to the edge on the target. The direction is indicated by a light blue arrow in the data viewer.
Fixed Angle *Dependent settings* Fixed Angle Value	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.
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).
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.
Edge Detection Mode	One of the following: Step or Corner. **Step**: Searches for steps on each path profile. For additional settings when you choose this mode, see *Step Edge Detection Mode Parameters* on the next page. **Corner**: Searches for slopes on each path profile. When you choose this mode,

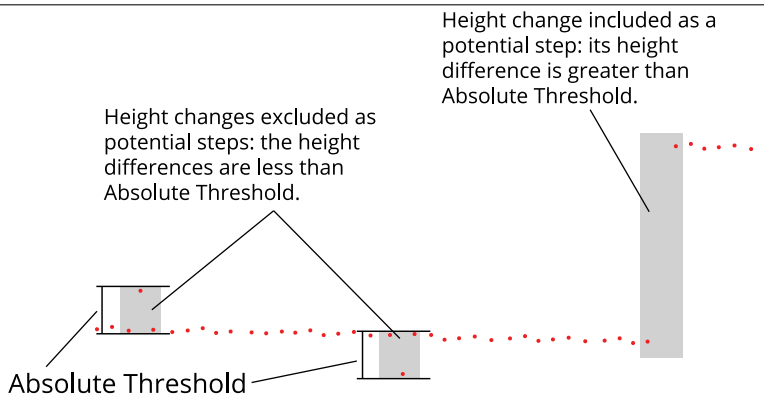
Parameter	Description
	several of the tool's parameters are hidden.
Selection Type Corner Type	Determines which step (when **Edge Detection Mode** is set to Step) or corner (when **Edge Detection Mode** is set to Corner) the tool uses on each path profile when there are multiple steps or corners in the profile. An edge point is placed on each chosen step or corner. Steps must pass the criteria of the tool's **Absolute Threshold**, **Step Direction**, and **Relative Threshold** settings (see *Step Edge Detection Mode Parameters* below). **Best**: Selects the greatest step or corner on each path profile. **First**: Selects the first step or corner on each path profile. **Last**: Selects the last step or corner on each path profile. **Index**: Selects the step at the index provided in **Edge Index** on each path profile, with 0 being the first step. (Only available when **Edge Detection Mode** is set to Step.) When **Edge Detection Mode** is set to Corner, the following additional options are available in Corner Type: **Top**: Selects the top-most corner on each path profile. **Bottom**: Selects the bottom-most corner on each path profile.
Edge Index	When **Selection Type** is set to Index, indicates which step to select; 0 is the first step from the start of the region.
Max Step Width	When **Selection Type** is set to Index, specifies the maximum width of the step, before considering the next step to be a new edge. The step width is the distance between the beginning and the end of the step.
Show Detail	When disabled, hides the light blue path lines and edge points.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.



The following parameters are only displayed if you set **Edge Detection Mode** to Step.

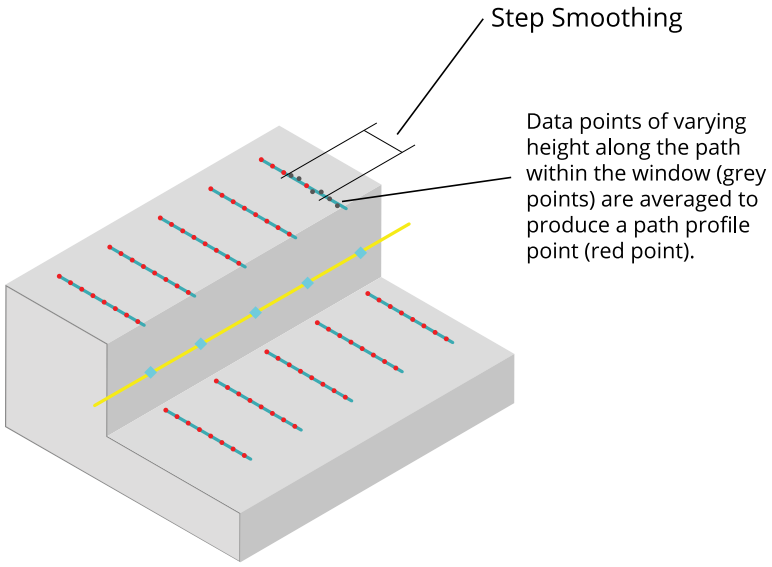
Step Edge Detection Mode Parameters

Step Direction	Determines whether the expected step rises or falls along the path, moving from left to right. Either **Rising**, **Falling**, or **Rising or Falling**.
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.

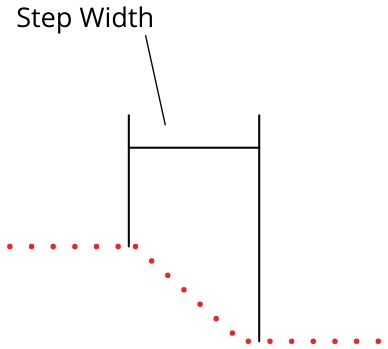


If **Step Smoothing** is set to 0, no averaging is performed (only the data point under the path is used).

For averaging perpendicular to the path, use **Path Width** (see above).

Step Width	The distance, along a path profile, separating the points used to find steps on a

path profile.

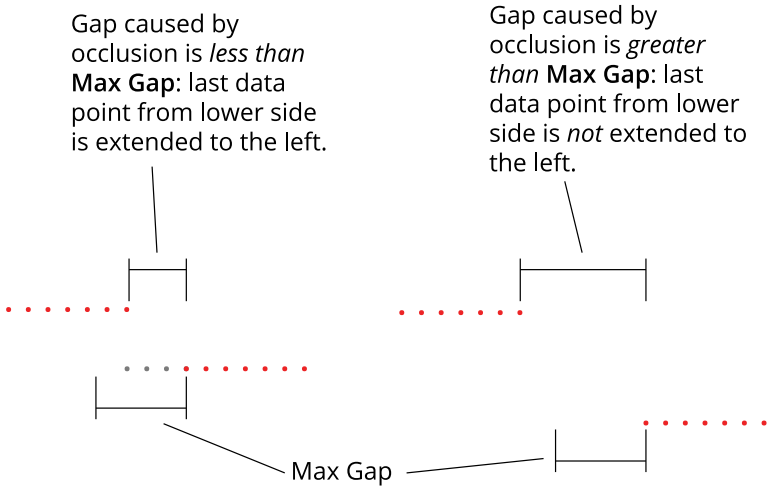


The setting is useful when you must detect a slope as an edge, rather than a sharply defined edge: setting **Step Width** to a value greater than the width of the edge ensures that the tool measures the height difference between the flat regions on either side of the edge. As a result, the height of 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**.



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 the part detection of an object sitting on a flat background (for more information on part detection, see *Profile Part Detection* on page 603). 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.
Edge Mode	One of the following: **Projected**: The line fitted to the edge is projected onto the XY plane. This mode is typically used with an edge that is parallel to the XY plane. **3D**: The line fitted to the edge follows the slope of the edge. This mode is typically used with a sloped edge.

Outputs

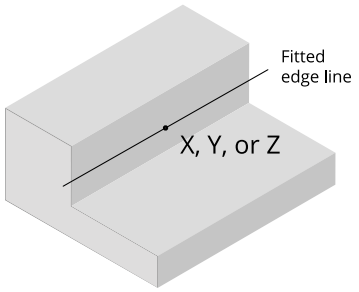
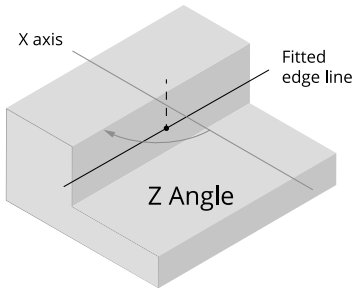
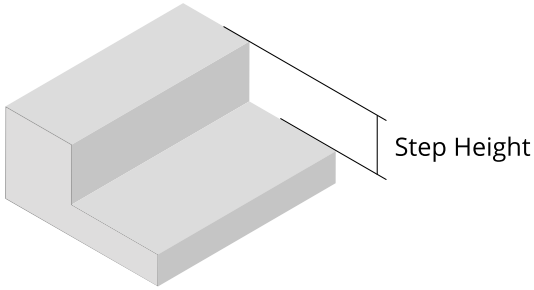
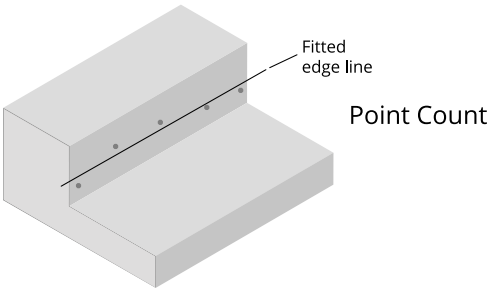
Most tools provide measurements, geometric features, or data as outputs.

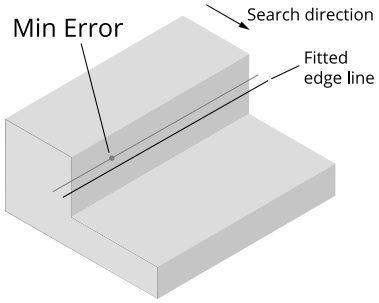
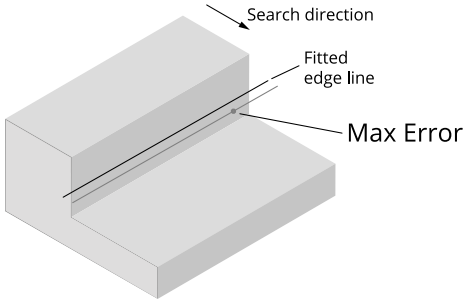


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.	
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 209.	
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.) This measurement returns Invalid when **Edge Detection Mode** is set to Corner.	
Point Count The number of points used to fit the line. Useful for determining if the number of points is above an acceptable minimum.	

Measurement	Illustration
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 **Outlier Fraction** parameter.	 

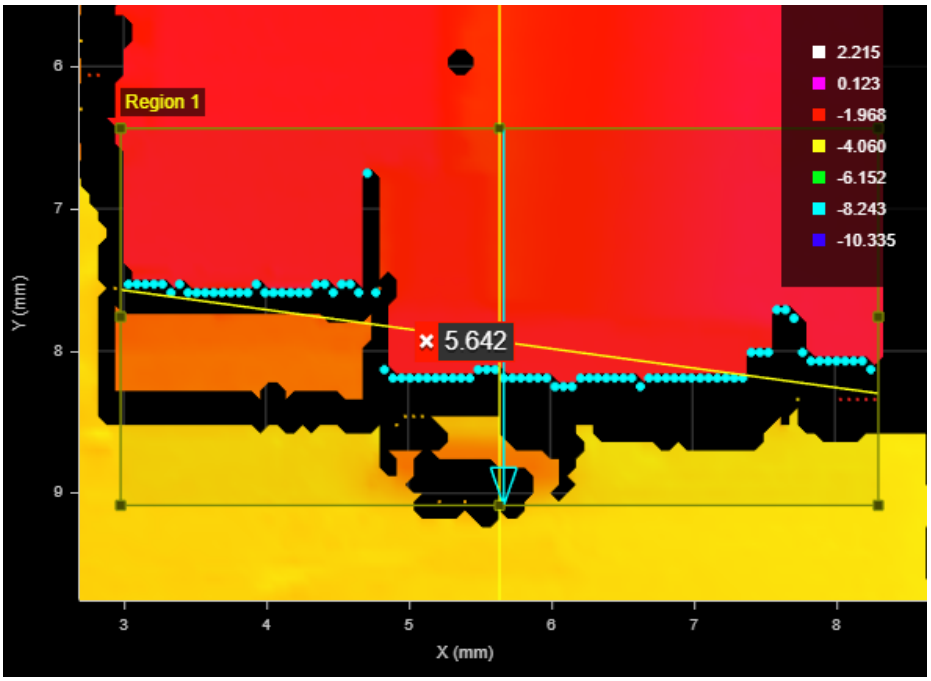
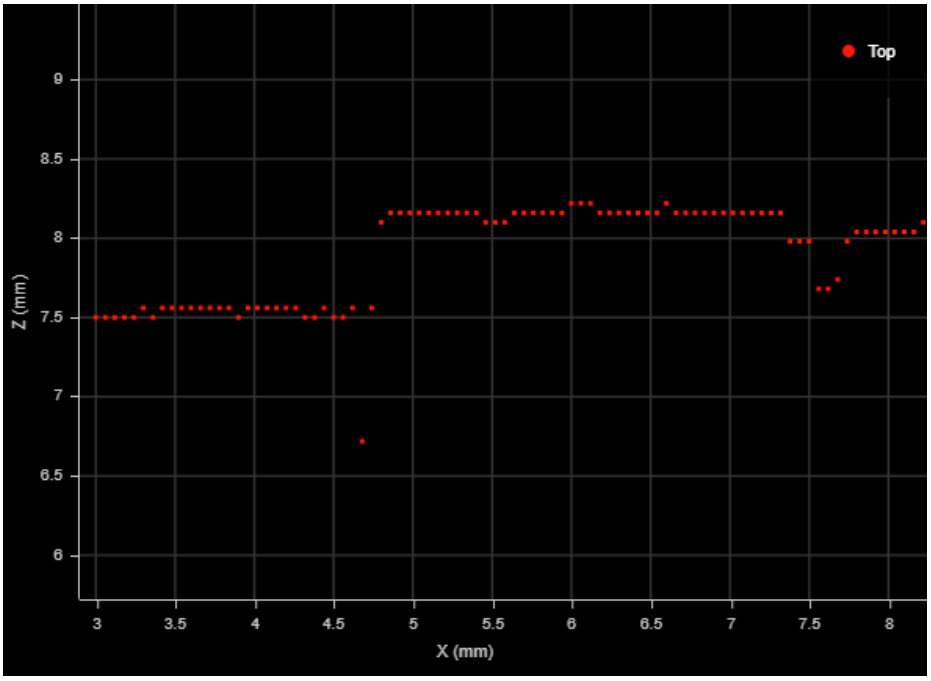
Features

Type	Description
Edge Line	The fitted edge line.
Center Point	The intersection point of the fitted edge line and the line representing the search direction through the center of the region of interest.
Edge Plane	A plane on the XZ axes at the fitted edge line.

For more information on geometric features, see *Geometric Features* on page 227.

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:

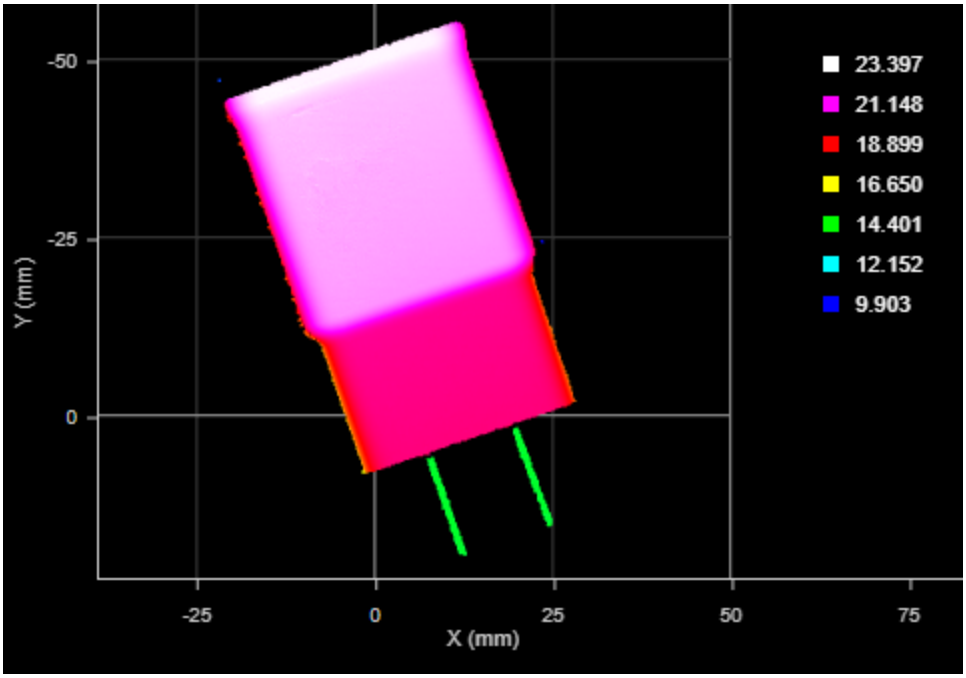
Type	Description
	 The profile is mirrored vertically when compared to the edge: Note how the single edge point toward the top of Region 1 in the surface data above is at the bottom of the extracted profile (below). 

Surface 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.

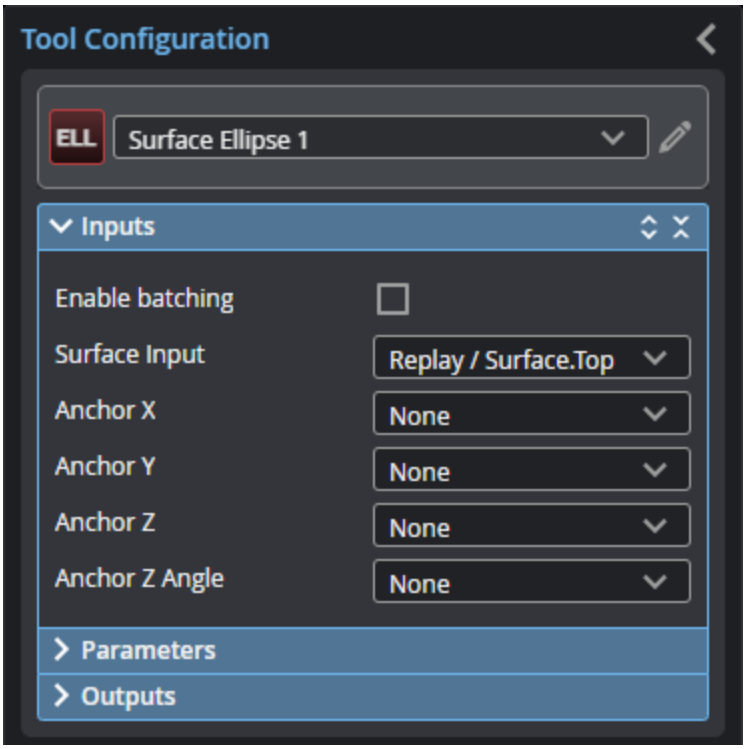
(The following screenshot is from a G2 sensor.)



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



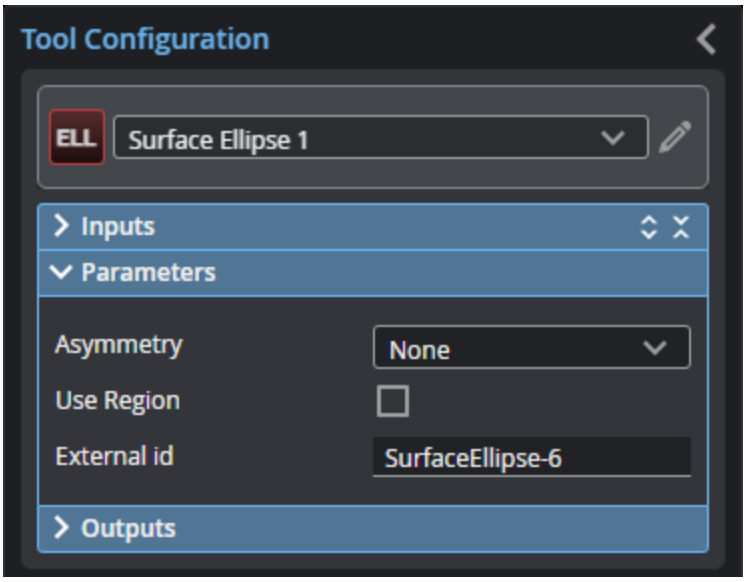
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

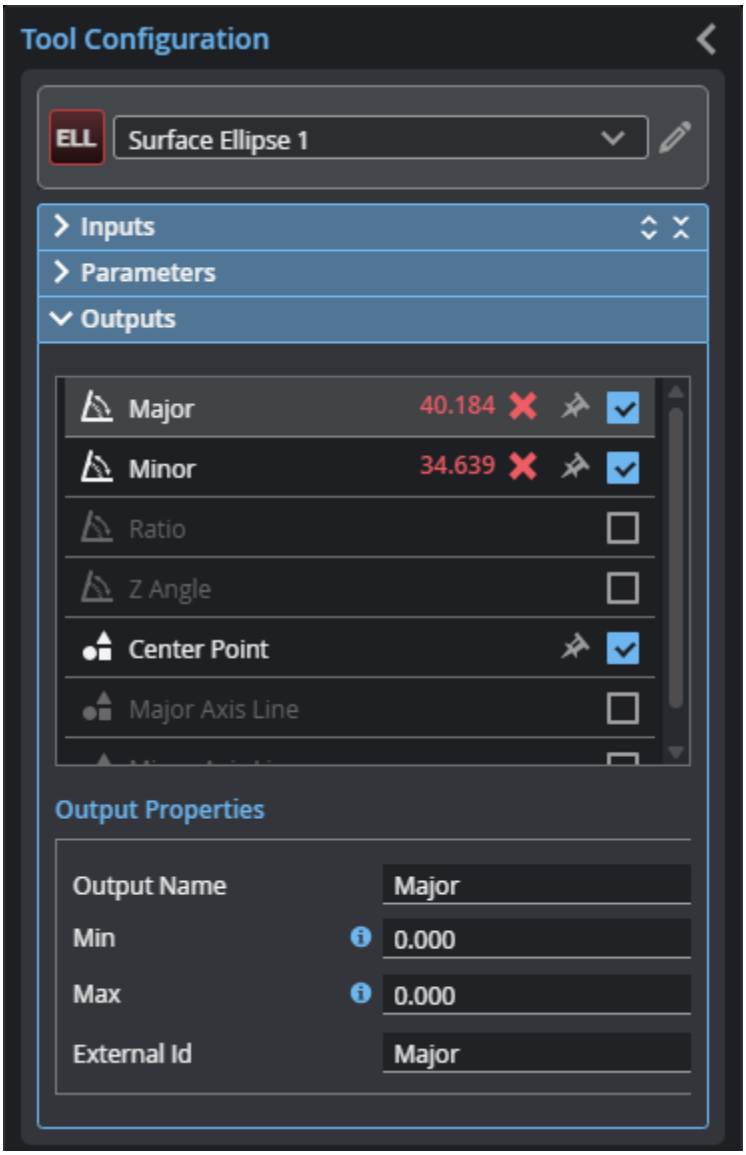


Parameters

Parameter	Description
Asymmetry	Resolves the orientation of an object over 360 degrees. The possible values are: 0 – None 1 – Along Major Axis 2 – Along Minor Axis
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

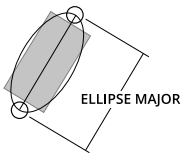
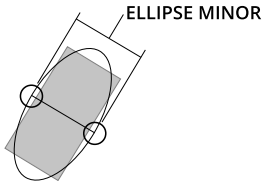
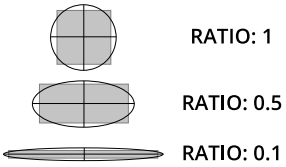
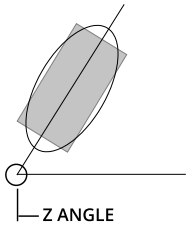


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.	

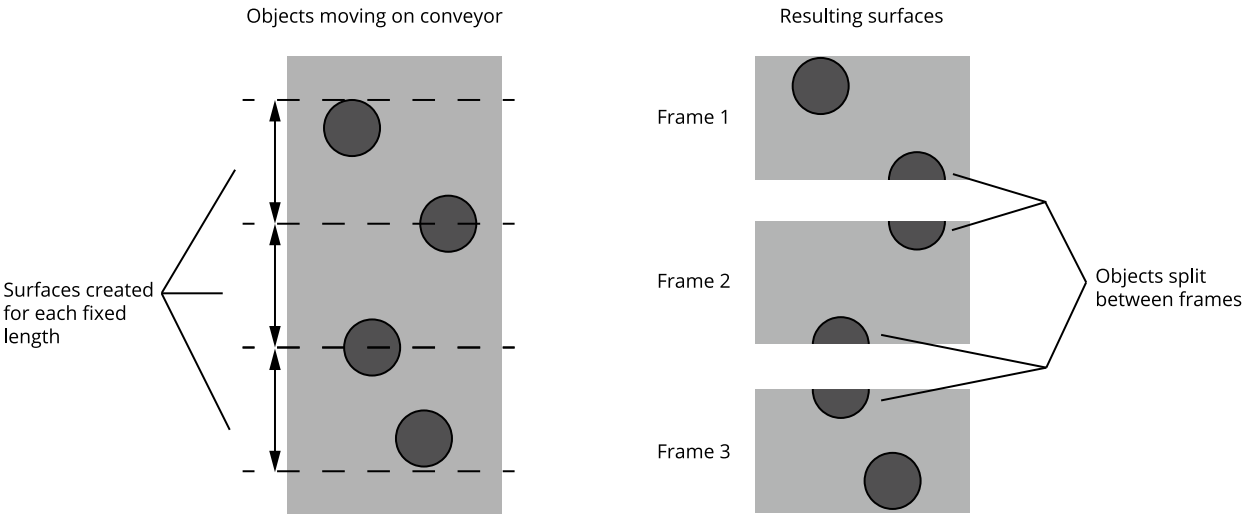
Features

Type	Description
Center Point	The center point of the fitted ellipse.
Major Axis Line	A line representing the major axis of the fitted ellipse.
Minor Axis Line	A line representing the minor axis of the fitted ellipse.

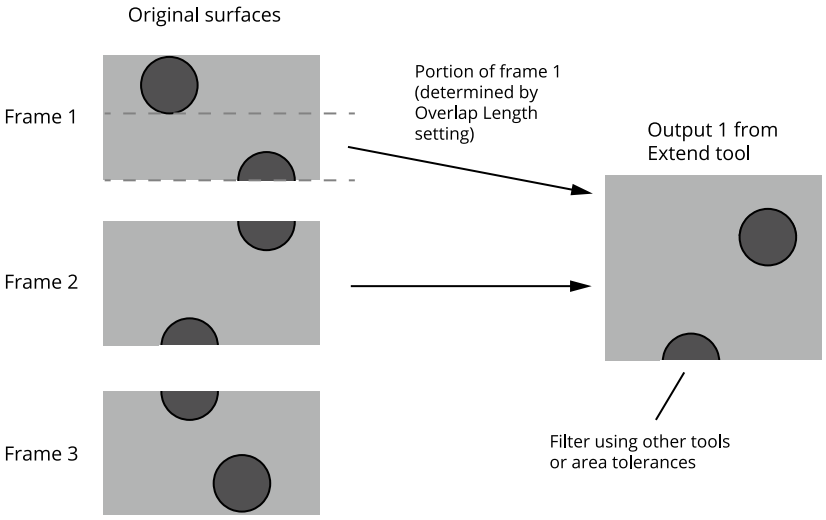
For more information on geometric features, see *Geometric Features* on page 227.

Surface Extend

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:

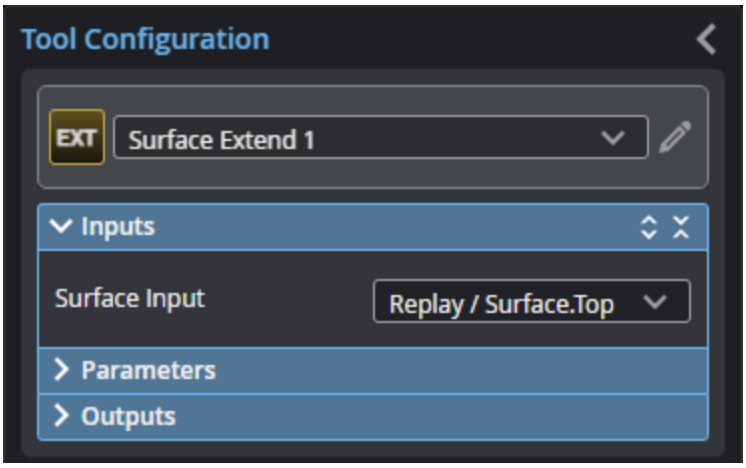


Data is only appended in one direction. Partial objects in the resulting surface output from the tool must be filtered out using downstream tools, for example, excluding them based on the expected area.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



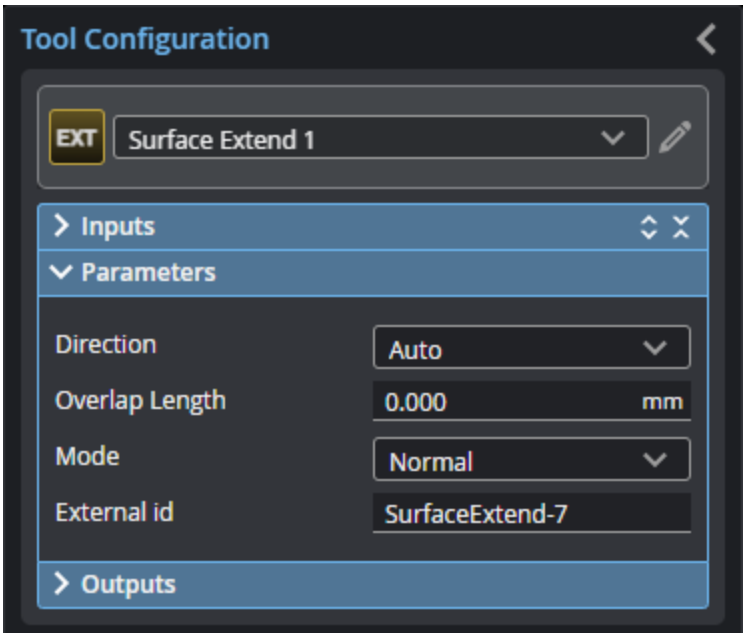
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

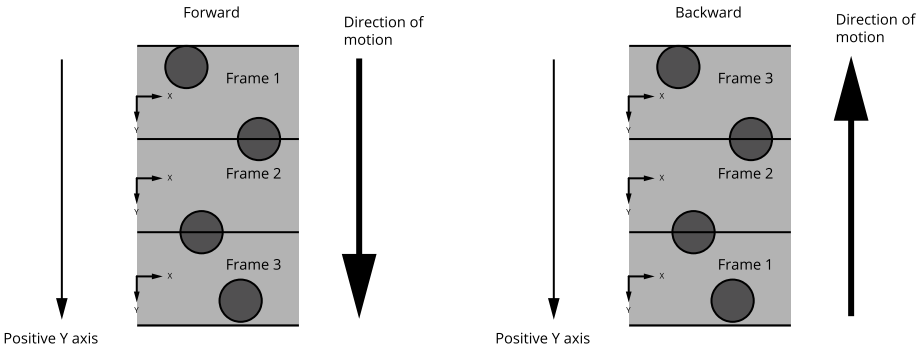
Name	Description
Surface Input	The data the tool applies measurements to or processes.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

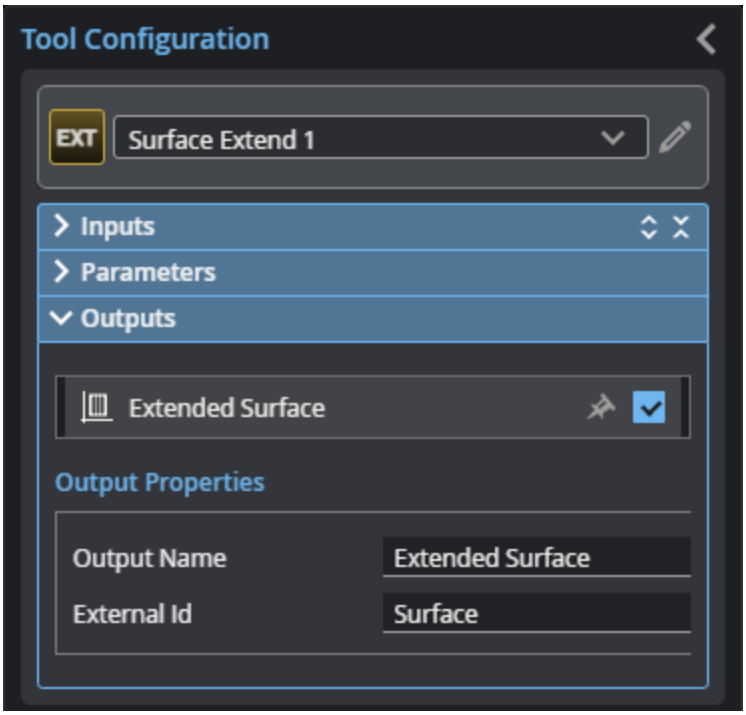


Parameters

Parameter	Description
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 *Triggers* on page 141) and the orientation of the sensor. • **Auto**: Choose this when Encoder is selected as the trigger source, in which case the tool will detect the direction of travel relative to encoder increase / decrease. • **Forward**: Choose this option when the trigger source is not set to Encoder and the direction of motion matches the sensor's positive Y axis. • **Backward**: Choose this option when the trigger source is not set to Encoder and the direction of motion is the opposite of the sensor's positive Y axis.
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's output is unlocked and functions normally. • **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.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Data

Type	Description
Extended Surface	Data containing an extended surface, available for use as input in the **Stream** drop-down in other tools.

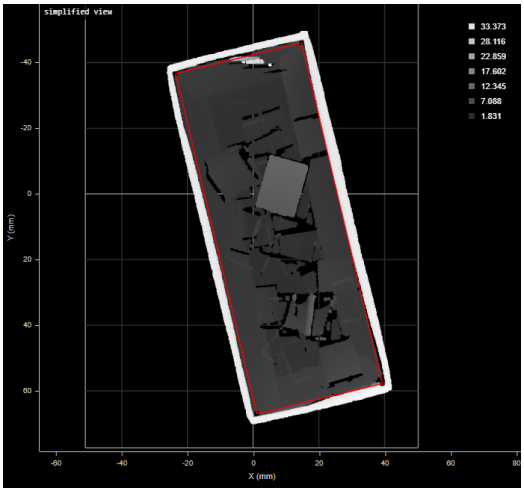
Surface Feature Mask

The Surface Feature Mask dynamically defines a polygon or circle mask using Point geometric features produced by other tools to crop a uniform surface and output it. You can use this tool in void fill applications with bins of varying sizes and varying orientations.

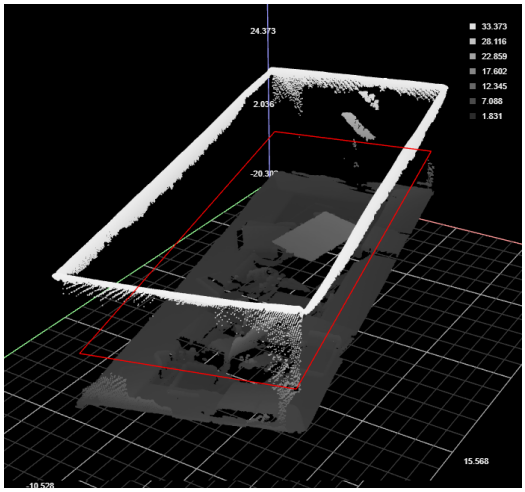
The following is a possible tool chain for calculating the fill volume of rectangular bin using Surface Feature Mask:

- Surface Bounding Box tool: Finds the orientation and position of the center of a box that must be filled. You would use the tool's X, Y, and Z Angle measurements as anchors for the other tools.
- Surface Edge tools (one for each edge of the bin): Using the Bounding Box tool's measurements as anchors, these tools find the edges of the bin.
- Feature Intersect tools: Using pairs of lines from the Surface Edge tools as inputs, these tools return Point geometric for the corners of the bin.
- Surface Feature Mask tool: Using the Point geometric features from the Feature Intersect tools as inputs, this tool outputs Surface data delimited by corners of the bin. To exclude the edges of the bin itself, use the **Padding Distance** parameter to exclude the bin walls.

When only two Point geometric features are passed as input to the tool, it creates a circular mask, where one point is the circle's center point, and the other is a point on the circle.



2D View

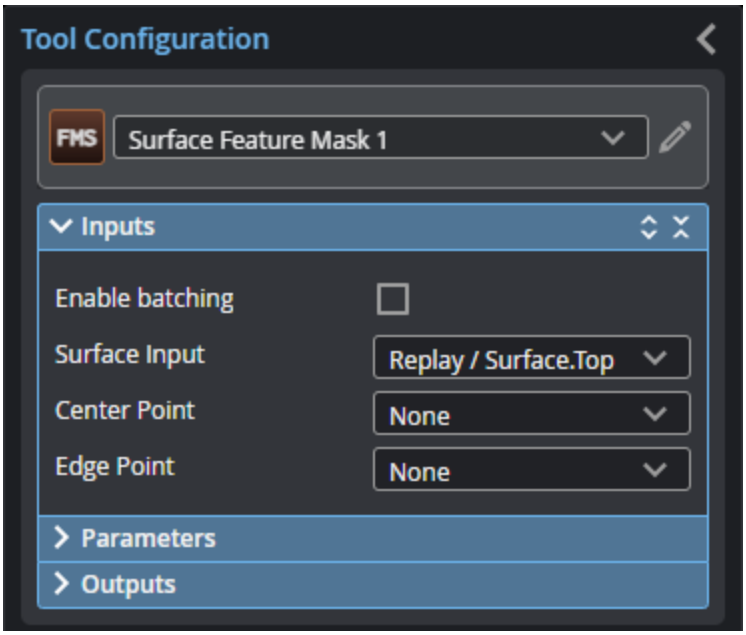


3D View

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



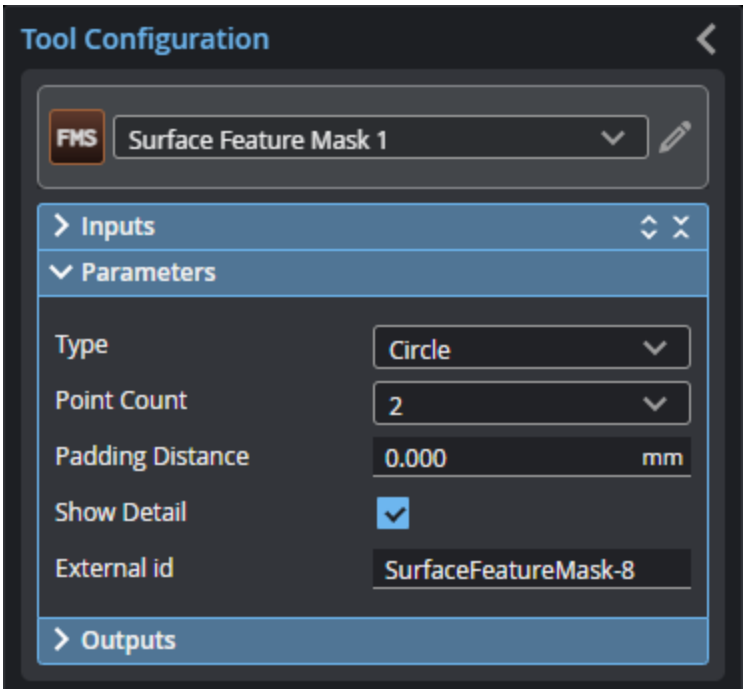
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Point {n}	The nth Point geometric feature. Only displayed if you set **Point Count** to 3 or higher.
Center Point	The center Point geometric feature of the circle. Only displayed if you set **Type** to Circle and **Point Count** to 2.
Edge Point	The Point geometric feature on the circle. Only displayed if you set **Type** to Circle and **Point Count** to 2.
Surface Input	The data the tool applies measurements to or processes.

For more information on geometric features, see *Geometric Features* on page 227.

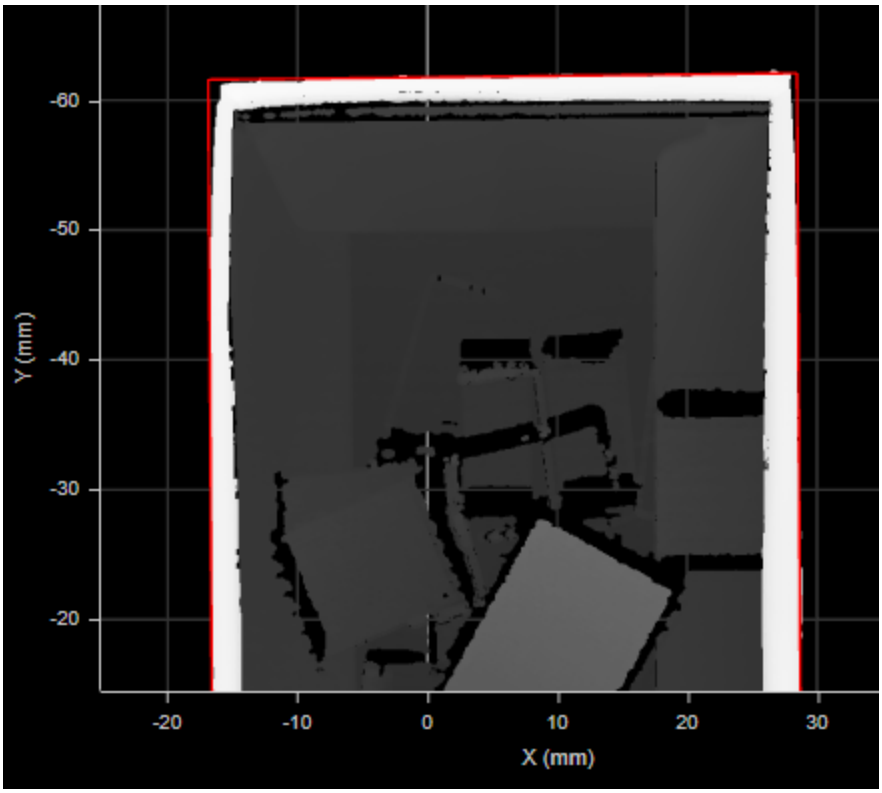
Parameters

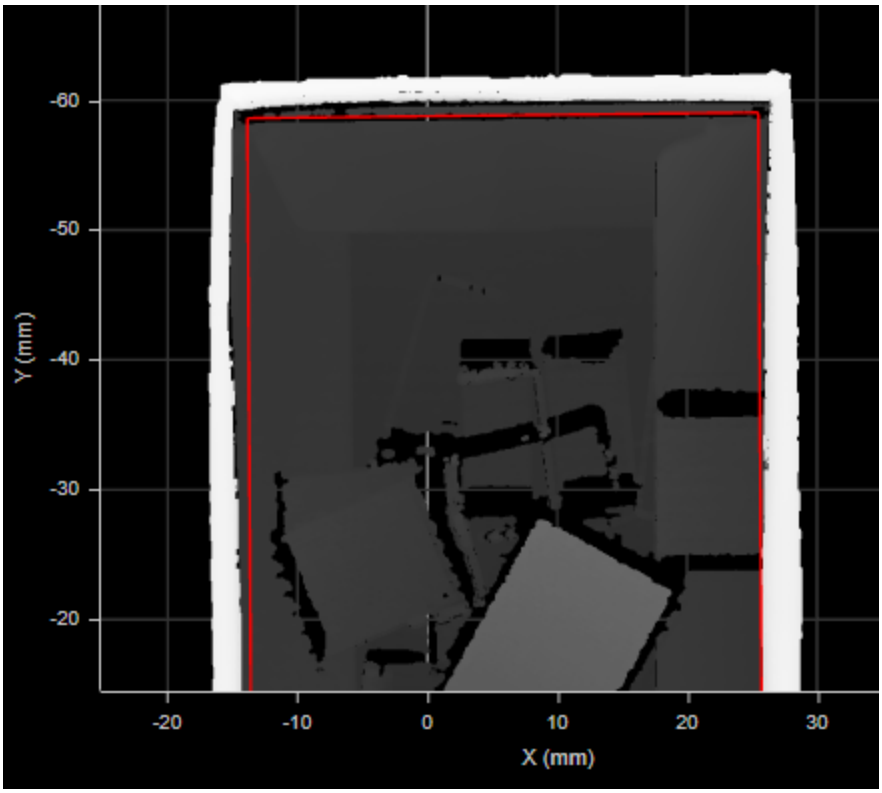
You configure the tool's parameters in the expandable **Parameters** section.

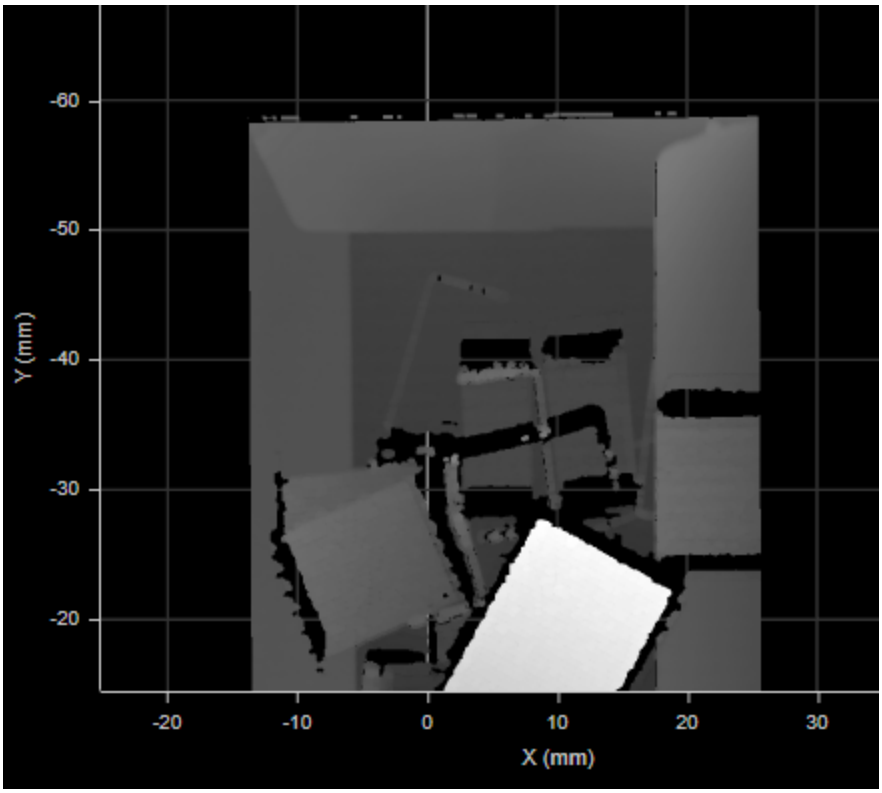


Parameters

Parameter	Description
Type	The shape type of the mask. It supports two options: **Polygon** - The tool needs at least 3 Point geometric features. **Circle** - The tool needs 2 or 3 point geometric features.
Point Count	The count of the input point features. For the Polygon **Type** option, the tool supports integers in the range of 3 to 10. For the Circle **Type** option, the tool supports integers of 2 or 3. After setting this parameter, set the point inputs in the Inputs section to the appropriate Point geometric features.
Padding Distance	The offset from the mask. When the value is greater than 0, the offset is outward. When the value is less than 0, the offset is inward. You can use this for example to exclude a bin that contains parts from the output Surface data. For example, in the following, before padding is set to a non-zero value, the bin is included in the output.

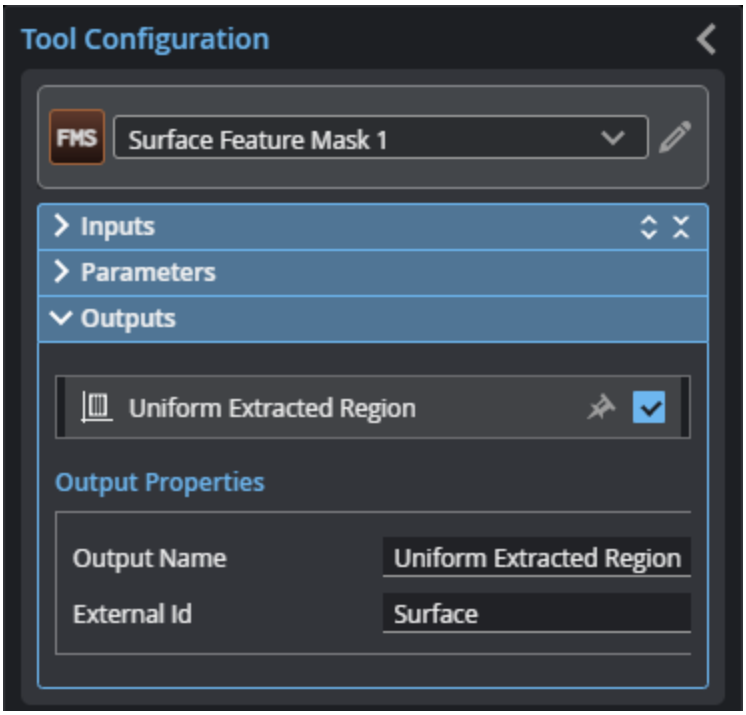
Parameter	Description
	 In the following, padding has been set to -3, which moves the mask "inward," excluding the bin.

Parameter	Description
	 The resulting Uniform Extracted Region outpu contains only the content of the bin.

Parameter	Description
	
Show Detail	Toggles whether to display the graphics of the mask region.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Data

Type	Description
Uniform Extracted Region	The cropped Surface data.

Surface Filter

The Filter tool provides several common vision processing filters that you can apply to surface data, as well as two "cropping" filters that output a subset of the surface data, letting you pre-process scan data to get more repeatable measurements. Any Surface or Feature tool can use the resulting filtered surface data as input.

For a list of the filters, see *Filters* on page 865.

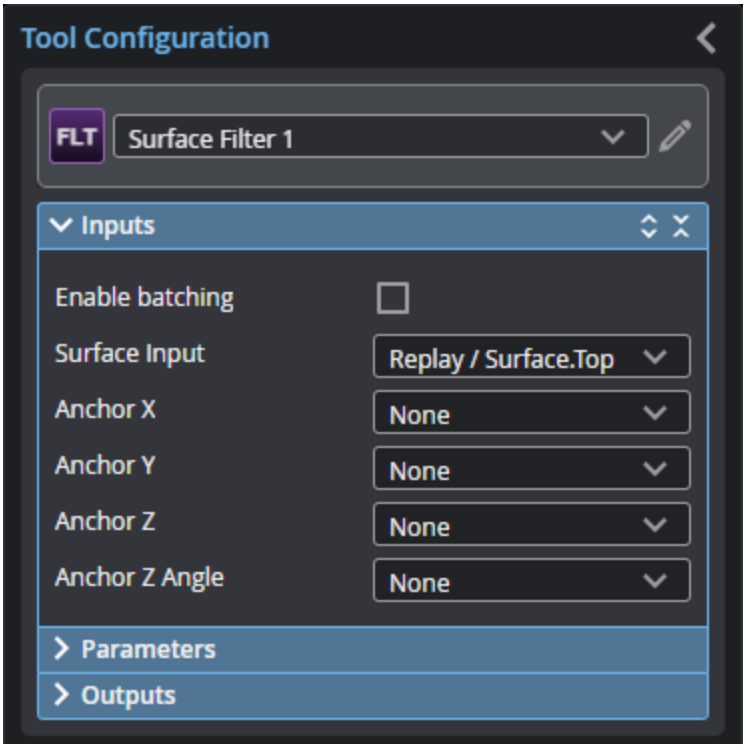
The Filter tool provides no measurements or decisions: it only outputs processed surface data.

> A limited set of filters is also available on the **Acquire** page, in the **Filters** section. These filters let you process scan data without needing to add tools. This can be useful if you are using a sensor mostly as an acquisition device.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



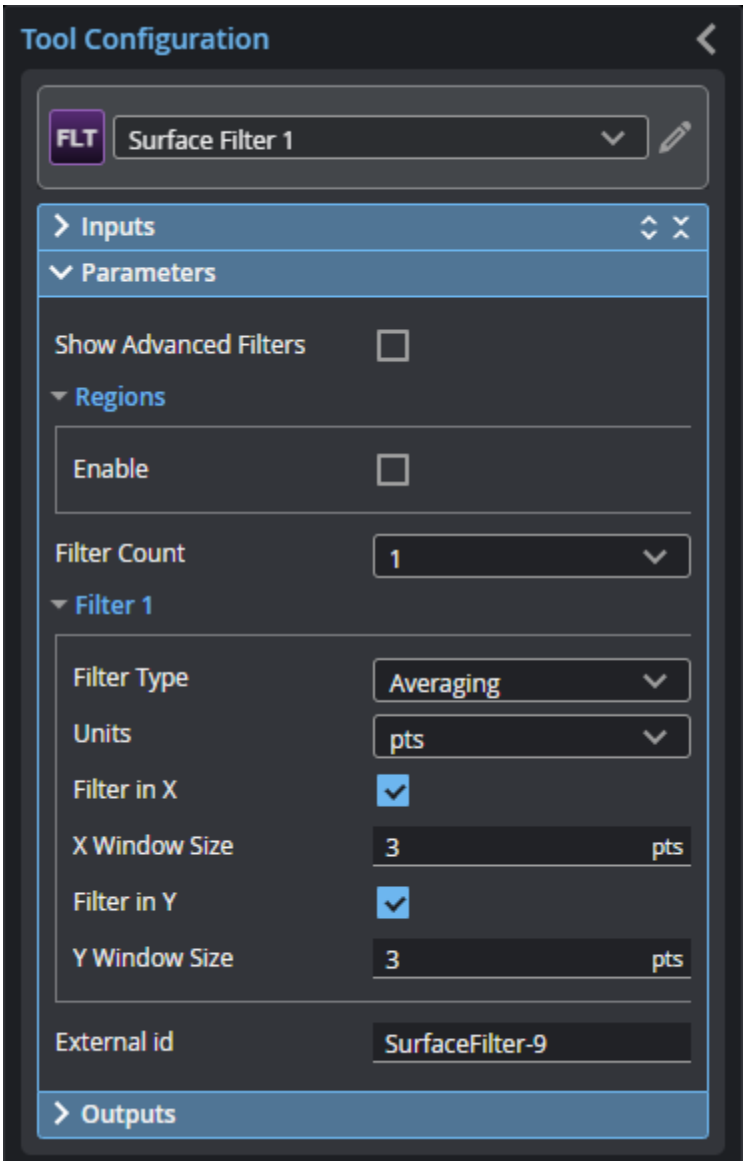
> To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
	The data the tool applies measurements to or processes.
Anchor X Anchor Y	Positional anchors are optional.
	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Show Advanced Filters	When enabled, displays advanced filters in **Filter Type** (see below), in addition to the standard filters (Gap Filling, Median, Averaging, and Decimation).
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.

Parameter	Description
Filter Count	Sets the number of filters in the tool.
Filter Type	The type of the filter. For information on the available filters, see *Filters* below.
Units	Specifies whether filters use data points (**pts**) or millimeters (**mm**). Not available with all filters.
Preserve Data Outside Region	Enable this setting when **Mask Mode** is set to **Include Data in Region** to include data outside the region in the Filtered Surface data output. Otherwise, the Filtered Surface data output only includes the data in the region. This setting is only displayed if regions are enabled.
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.)
Filter in X Filter in Y	These parameters enable filtering along the X and the Y axis, respectively. Set the corresponding window sizes in **X Window Size** and **Y Window Size**. The filter window sizes are specified in millimeters (and additionally, in the tools, in data points). To calculate the number of data points that a window covers when the units are millimeters, use the following calculation: • User-specified window size divided by the X spacing interval (that is, the number of millimeters per point) on the **Spacing** tab in the **Sensor** panel. (For more information on spacing intervals, see *Spacing on X and Y (Uniform Spacing)* on page 140.) • *With the exception of the gap filling filter*, round the result of the division to the nearest integer value. With the gap filling filter, filling is performed within the provided window size. For example, if you set the size of the filter's window to a value between 1.5 mm and 2.49 mm (inclusively), and the X spacing interval is set to 1 mm, the filter covers 2 data points. A filter window size from 2.5 mm to 3.49 mm results in a filter covering 3 data points.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

The following filters are available in the Filter tool. Filters described below as advanced are only displayed if **Show Advanced Filters** is enabled.

Filters

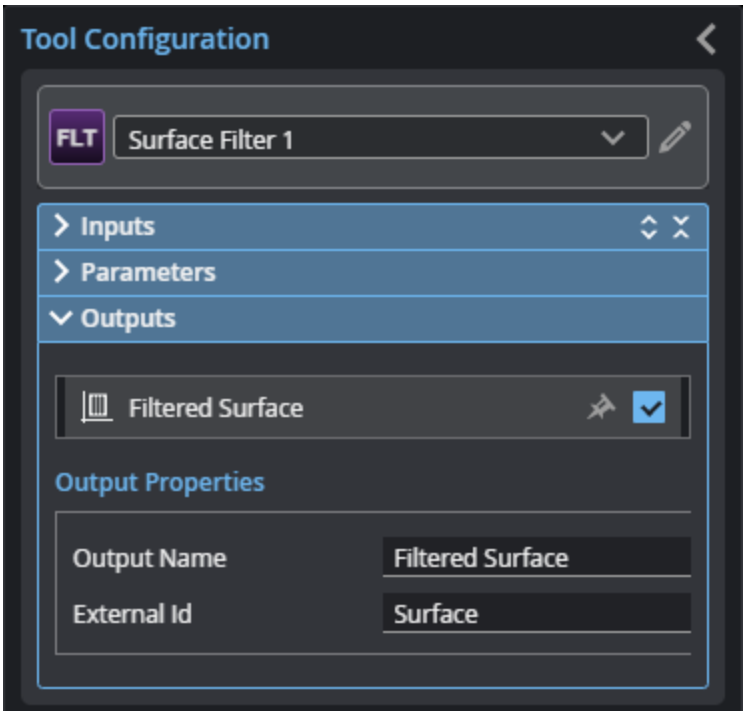
Name	Description
Gap Filling	Fills in missing data using information from the nearest neighbor data points, for example, when data is missing due to occlusions. Gap filling also fills gaps where

Name	Description
	no data is detected, which can be due to low surface reflectivity, for example dark or specular surface areas, or to actual gaps in the surface. The values in **Filter in X** and **Filter in Y** represent the maximum gaps the sensor will fill. Wider gaps are not filled. 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 the specified window. The sensor can fill gaps along both the X axis and the Y axis. In Profile mode, gap filling is limited to the X axis.
Median	Substitutes the value of a data point with the median calculated within the window or windows set in **X Window Size** or **Y Window Size** around the data point. If the number of valid (non-null) data points in the window is even, the average of the two values in the center is used. If the number of valid points is odd, the value in the center of the sorted list of values is used. Missing data points will not be filled with the median value calculated from data points in the neighborhood. With an odd-numbered window size, the output is at the center of the window. With an even window size, the output is 0.5 pixels to the right of the center (that is, using window / 2-1 values from the left, and window / 2 from the right.
Averaging	Substitutes a data point value with the mean value of that data point and its nearest neighbors within the window or windows set in **X Window Size** or **Y Window Size**. 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 neighborhood.
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 Window Size** to 0.2, only points every 0.2 millimeters will be used. The filter generates points starting from the leftmost edge of the scan data, stepping in equal steps away from that side.
Gaussian (advanced)	A Gaussian filter. Set the **Sigma**, **Units**, and **Preserve Data Outside Region** parameters as required.
Opening (advanced)	Erosion followed by dilation.
Closing (advanced)	Dilation followed by erosion.
Erosion (advanced)	Applies an erosion filter. Lets you specify the direction of the erosion; one of the following: • Horizontal

Name	Description
	• Vertical • Symmetrical
Dilation (advanced)	Applies a dilation filter. Lets you specify the direction of the dilation; one of the following: • Horizontal • Vertical • Symmetrical
Morph Gradient (advanced)	Applies a morphological gradient. The difference between dilation and erosion.
Sobel Magnitude (advanced)	Applies a Sobel magnitude filter. Lets you specify the direction of the filter; one of the following: • Horizontal • Vertical • Symmetrical
Laplacian (advanced)	Applies a Laplacian filter. Useful for detecting areas of distinct edges. Uses the following kernel: 0 -1 0 -1 4 -1 0 -1 0
Negative (advanced)	Inverts the height or intensity values in the scan data. When **Use Nulls** is enabled, null values in the input are replaced by the highest height value.
Equalize (advanced)	Normalizes the norm or value range of an array.
Binarize (advanced)	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.
Percentile (advanced)	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 (advanced)	Crops scan data relative to a user-specified reference. When **Reference Type** is set to Range, specify minimum and a maximum heights in the threshold fields. When the type is Region, use **Reference Region** to set the heights relative to one or more reference region. When the type is Plane, set the **Plane** input in the tool's Inputs section to a plane geometric feature.
Crop only (advanced)	Crops the scan data to the user-defined region.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Data

Type	Description
Filtered Surface	The filtered data, available for use as input in the **Input** drop-down in other tools.

Surface Flatness

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.

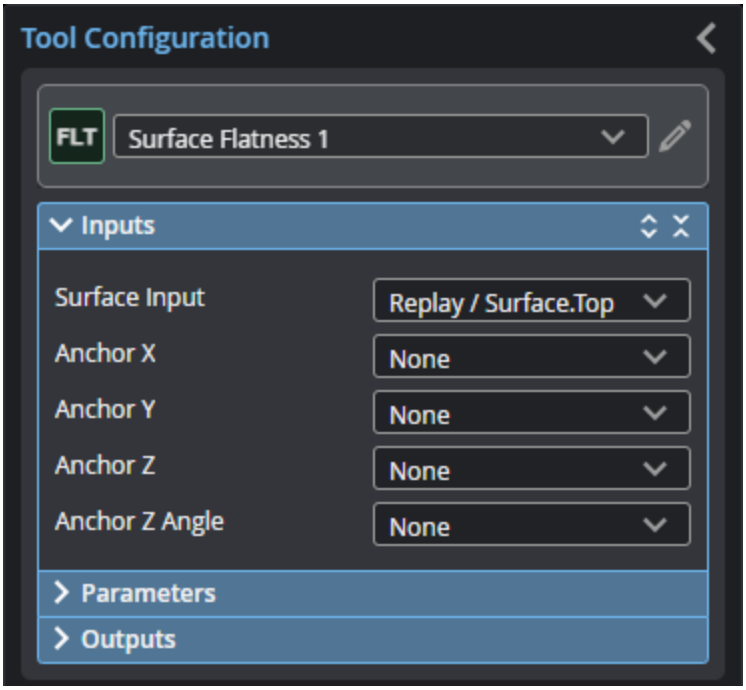
The tool lets you set a grid with user-defined cell sizes over a region, or more flexibly with multiple individual regions manually. In each case, "local" minimum and maximum heights, as well as flatness indicators (maximum - minimum), are returned. In addition, "global" minimum, maximum, and flatness measurements, which combine data from all flatness measurement areas, can also be returned. The tool measures the maximum and minimum distances from a best-fit plane for each cell or flexible region, 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.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



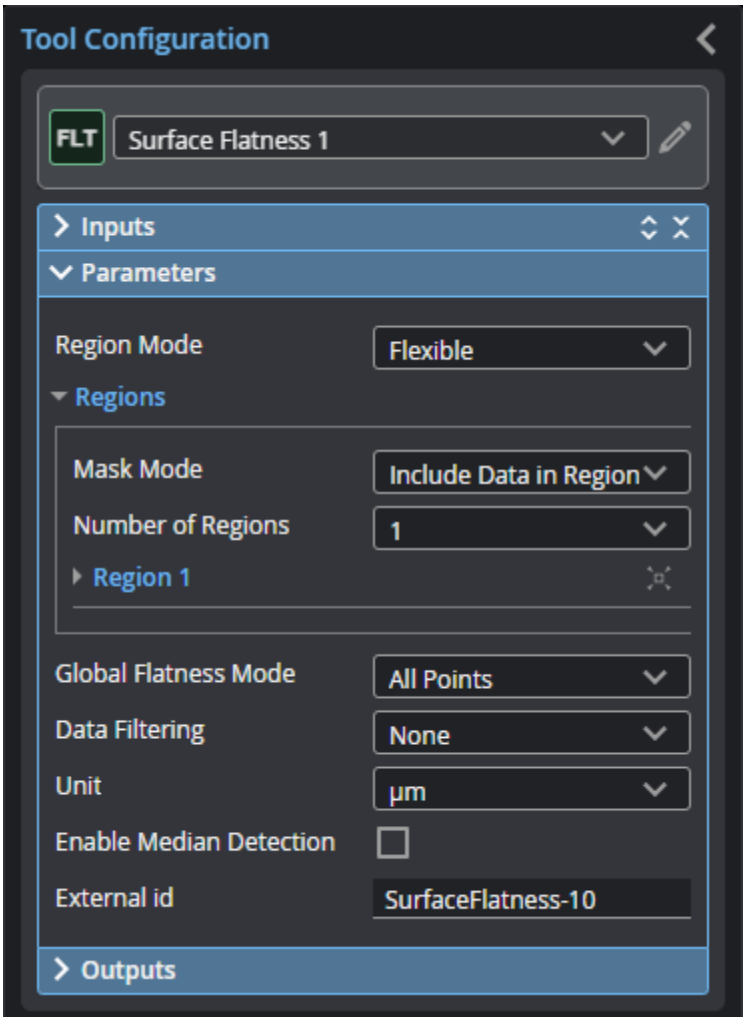
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Surface Input	The data the tool applies measurements to or processes.
Anchor X	Positional anchors are optional.
Anchor Y	
Anchor Z	
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



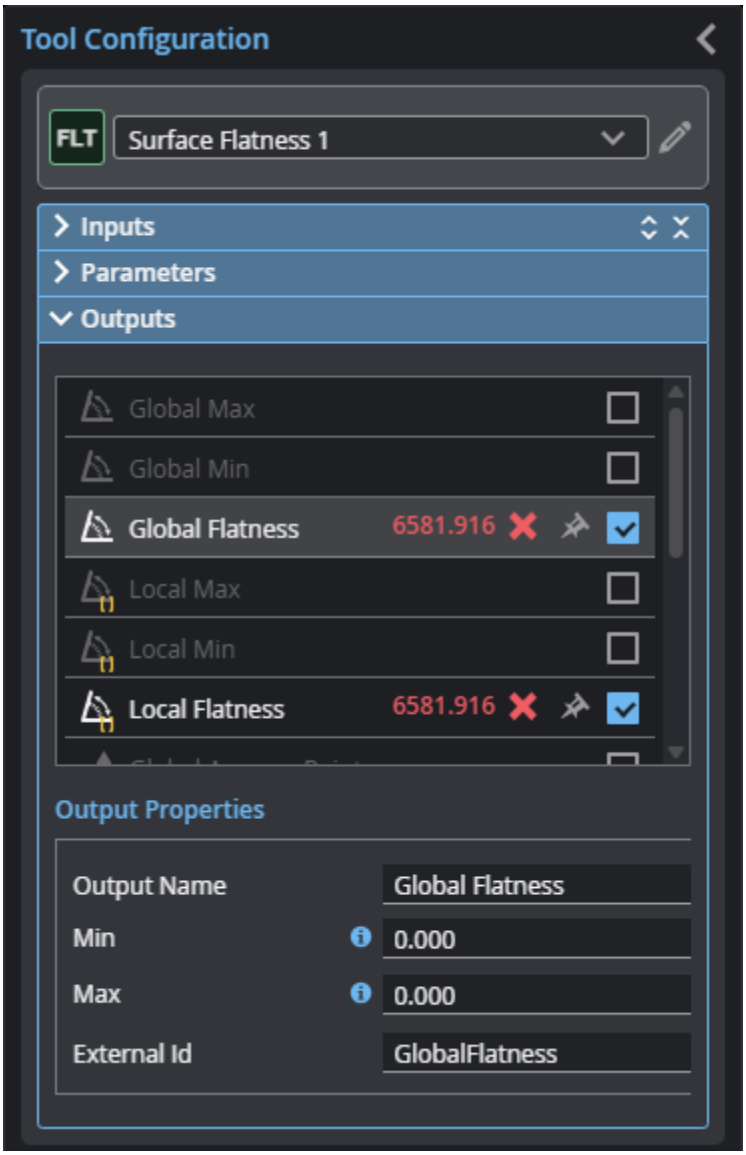
Parameters

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 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.)
Regions	When expanded, displays the region- and mask-related settings.
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
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. When you use Local Percentile or Global Percentile, you must set **High Percentile** and **Low Percentile**. **None** - The tool performs no filtering. **Local Percentile** - Applies the percentile filter to local masks first, and then merges to a global mask.

Parameter	Description
	Global Percentile - Merges to a global mask first, and then applies the percentile filter to the global mask.
Unit	Lets you choose which units the tool uses for measurement results. One of the following: • um (micrometers) • mm (millimeters)
Enable Median Detection	When enabled, Point geometric features for the global median and the local median are output.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

The unit of measurement values is set with the **Unit** drop-down.

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 **Local Min** **Local Flatness** Arrays containing the maximum distance, minimum distance, and flatness (maximum - minimum) calculated using the valid data points from the grid cells (when **Region Mode** is set to **Grid Pattern**), or the individual regions (when **Region Mode** is set to **Flexible**). For more information on arrays, see *Arrays, Batching, and Aggregation* on page 199.

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	An array of the planes fitted to the valid data points from the grid cells (when **Region Mode** is set to **Grid Pattern**), or those from the individual regions (when **Region Mode** is set to **Flexible**).
Global Average Point Global Max Point Global Min Point Global Median Point	Point geometric features representing the global average, the global maximum / minimum, and, if **Enable Median Detection** is enabled, the global median.
Local Average Point Local Max Point Local Min Point Local Median Point	Arrays of point geometric features representing the local average, local maximum / minimum, and, if **Enable Median Detection** is enabled, local median.

Surface Hole

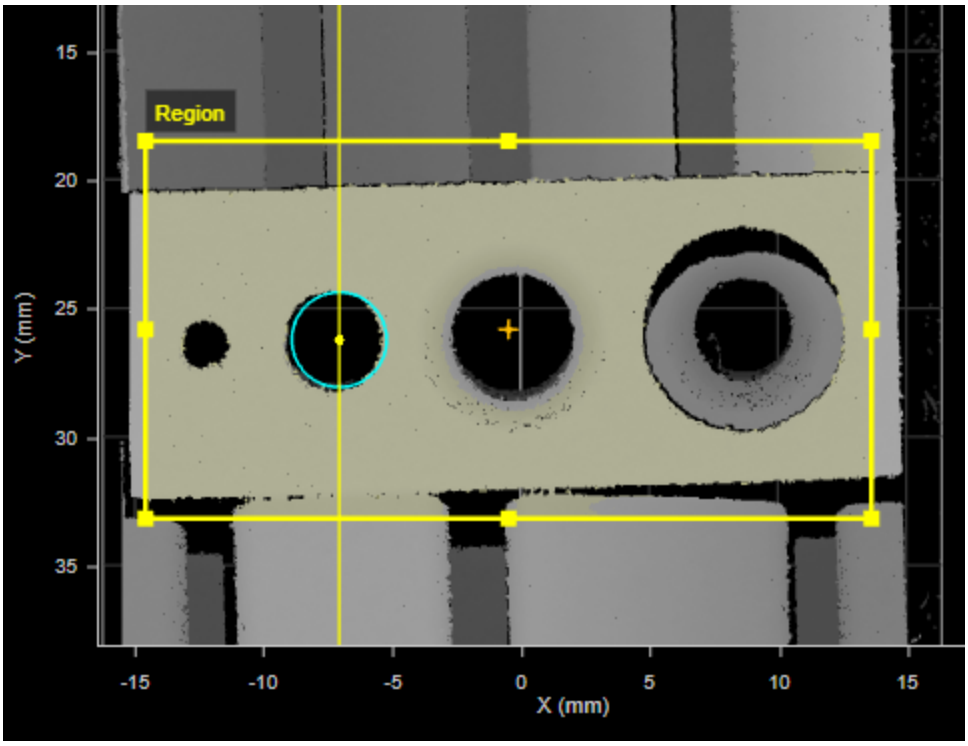
The Hole tool measures a circular opening within a region of interest on the surface and returns its position and radius.

> 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 hole and then return measurements. For a detailed explanation of the algorithm, see *Hole Algorithm* on the next page. The behavior of the algorithm can be adjusted by changing the tool's parameters.

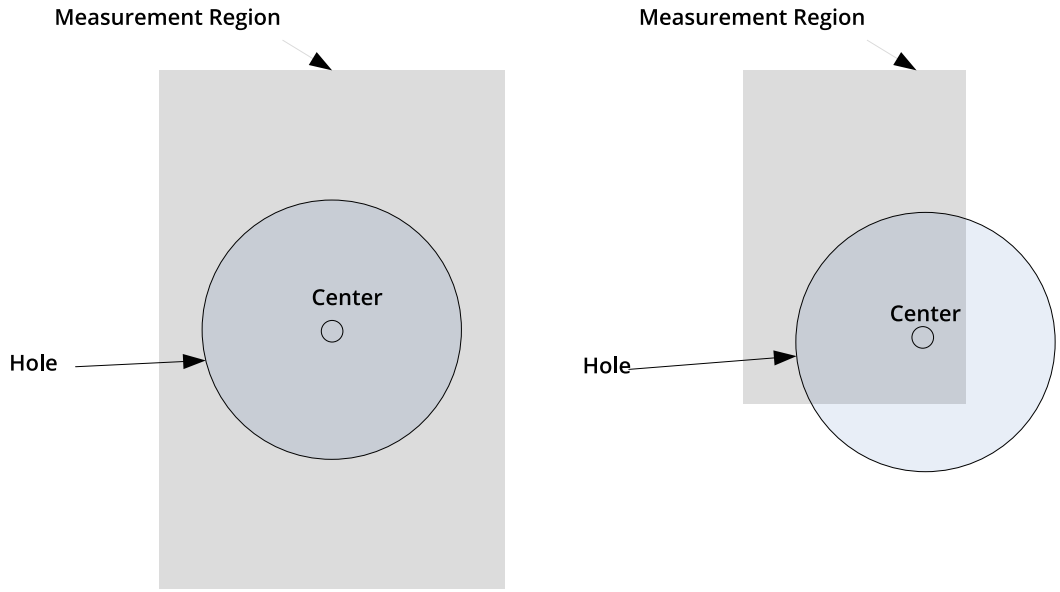
(The following screenshot is from a G2 sensor.)



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Measurement Region

The center of the hole must be inside the measurement region, even if the Partial Detection option is enabled. Either of the following is acceptable.



Hole Algorithm

The Hole tool processes the data in three phases: Search, Measure, and Filter. The algorithm can separate out background information that appears inside the hole. It can also detect holes that only partially appear in the data.

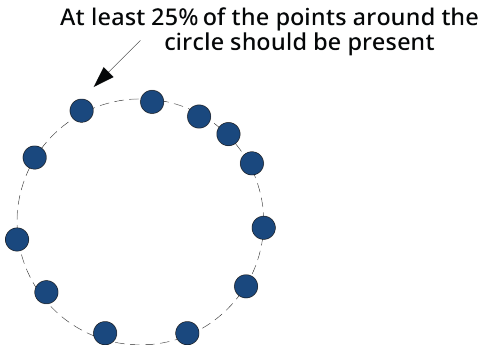
See the tool's parameters for an explanation of the options that affect the tool's algorithm.

Search phase - The tool searches for coarse data transitions (edge data) and performs a coarse fitting of the hole model (specified by the orientation angles and the nominal value) to determine the most likely candidate. If **Tilt Correction** is set to **Autoset**, the algorithm uses the data within the measurement region to estimate the orientation of the part.

Measure phase - A more rigorous edge detection algorithm is applied to precisely determine the edges around the feature. Edge detection at this stage will reject outliers and noise. The algorithm requires at least 25% of the data around the hole for the candidate to remain valid.

The accuracy of the algorithm improves when the points are spread more evenly along the hole's circumference.

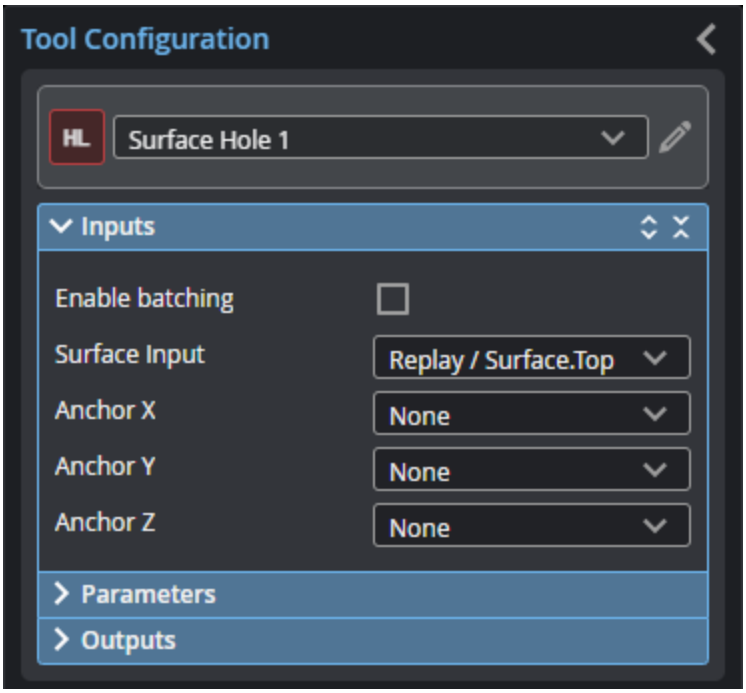
The set of refined edges is then used to locate and inspect the feature. If the Reference Regions option is enabled and set to AutoSet, the edges are also used to calculate the location of the reference regions.



Filter phase - The detected location and dimensions are then compared to the nominal and tolerance settings. If the refined feature falls within the measurement region and its measurements fit within the specified tolerance, the results are reported. If not, an invalid result is returned.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

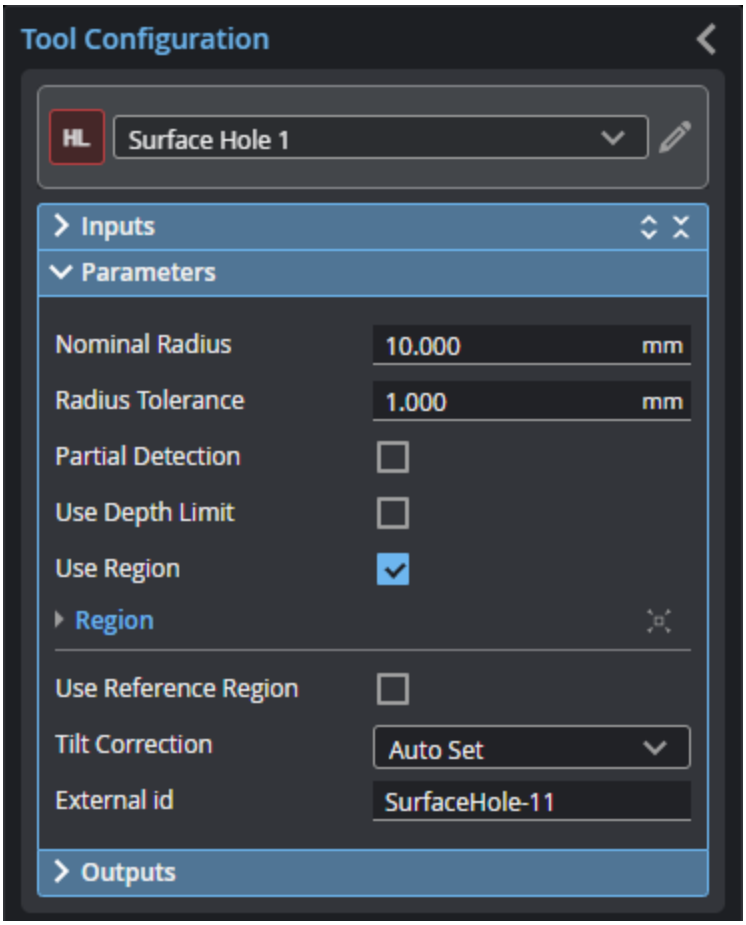
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array,

Name	Description
	other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.

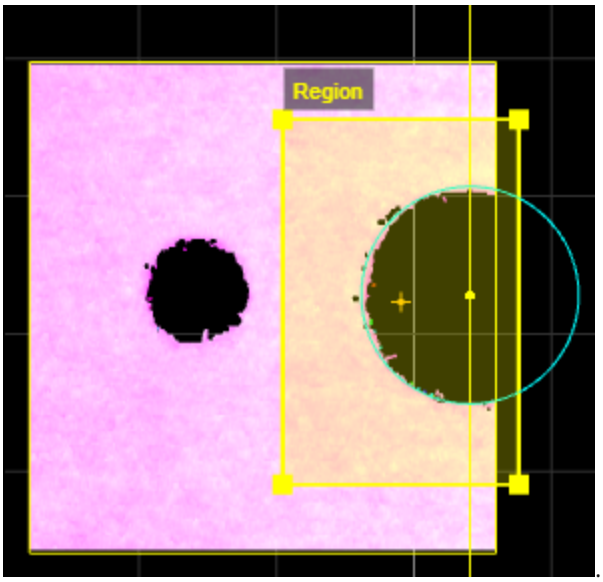
Parameters

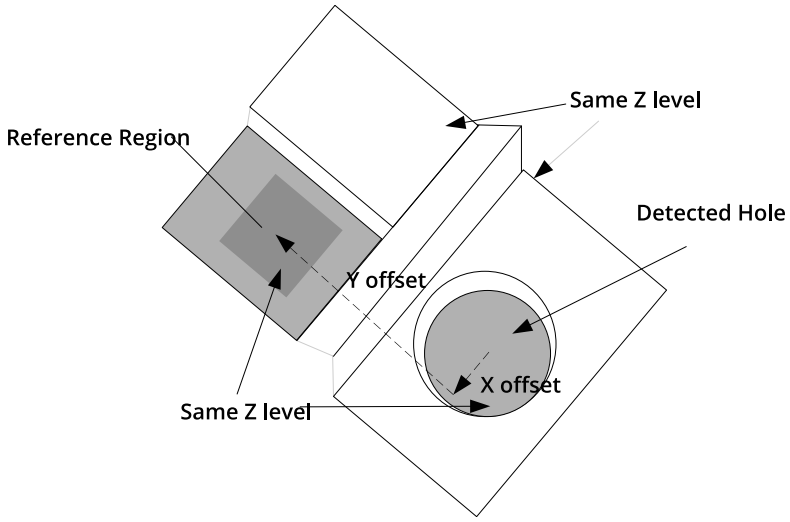
You configure the tool's parameters in the expandable **Parameters** section.



Parameters

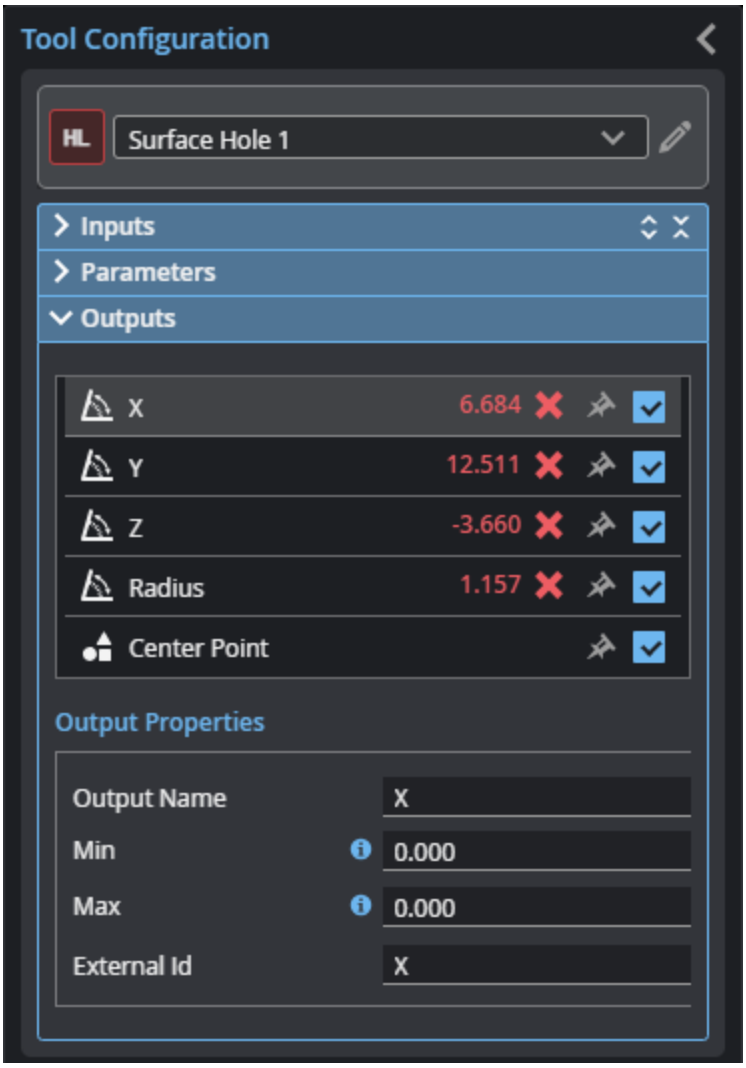
Parameter	Description
Nominal Radius	Expected radius of the hole.
Radius Tolerance	The maximum variation from the nominal radius (+/- from the nominal radius).

Parameter	Description
Partial Detection	Enable if only part of the hole is within the measurement region; the centre of the hole must be in the measurement region to be located. If disabled, the hole must be completely in the region of interest for results to be valid. 
Use Depth Limit	When **Use Depth Limit** is enabled, data below the value set in **Depth Limit** (relative to the surface) is excluded from the hole calculations.
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
Use Reference Region	When enabled, displays additional settings to let you set a reference region (see below).
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.

Parameter	Description
	 When this option is set to **Autoset**, the algorithm automatically determines the reference region. When the option is set to **1 Region** or **2 Regions**, you must manually define the 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. **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.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

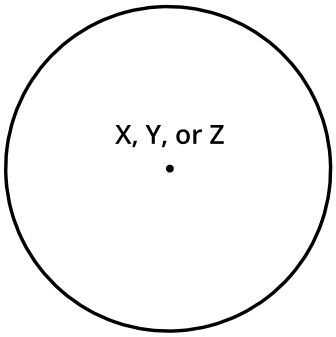
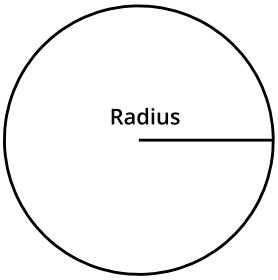


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.

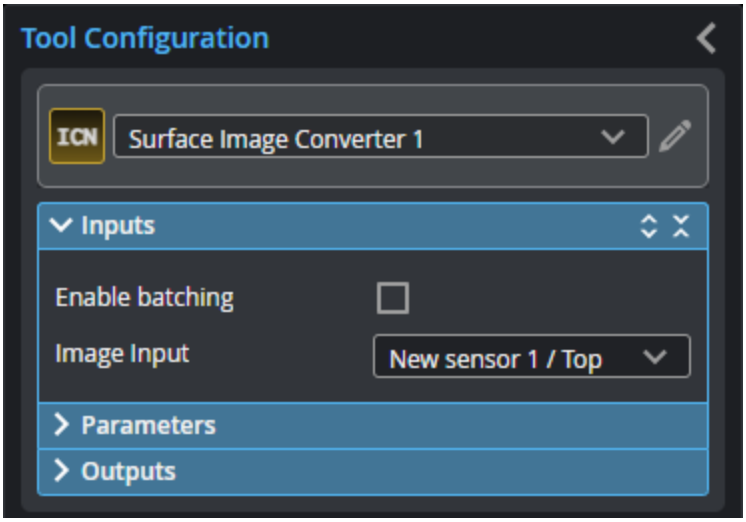
For more information on geometric features, see *Geometric Features* on page 227.

Surface Image Converter

The Surface Image Converter tool takes uniform Surface data (with or without intensity data) and converts the input into 8-bit or 16-bit grayscale Image data. You can then use Image tools available as part of the Pro Tools package. For more information, see *Pro Tools* on page 113.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

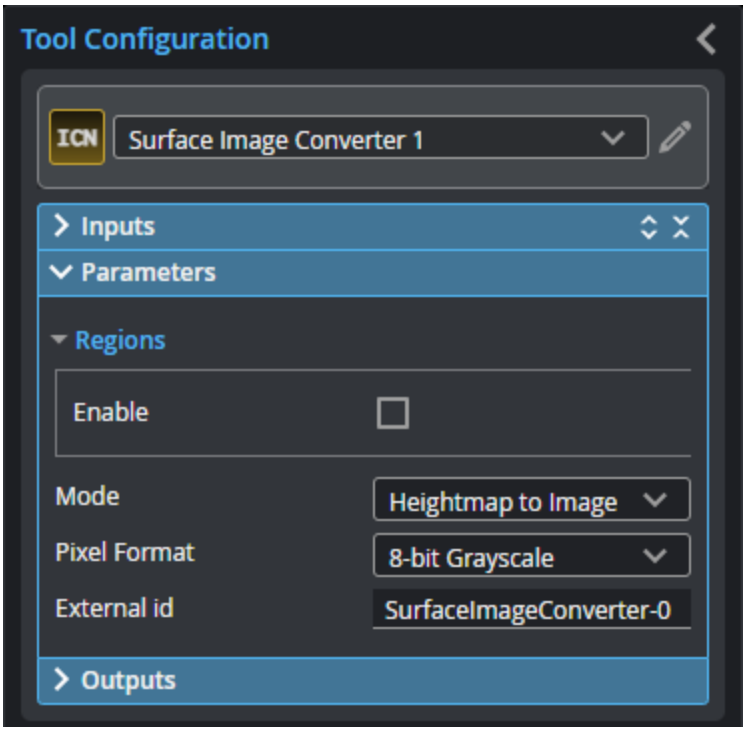


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.

Parameters

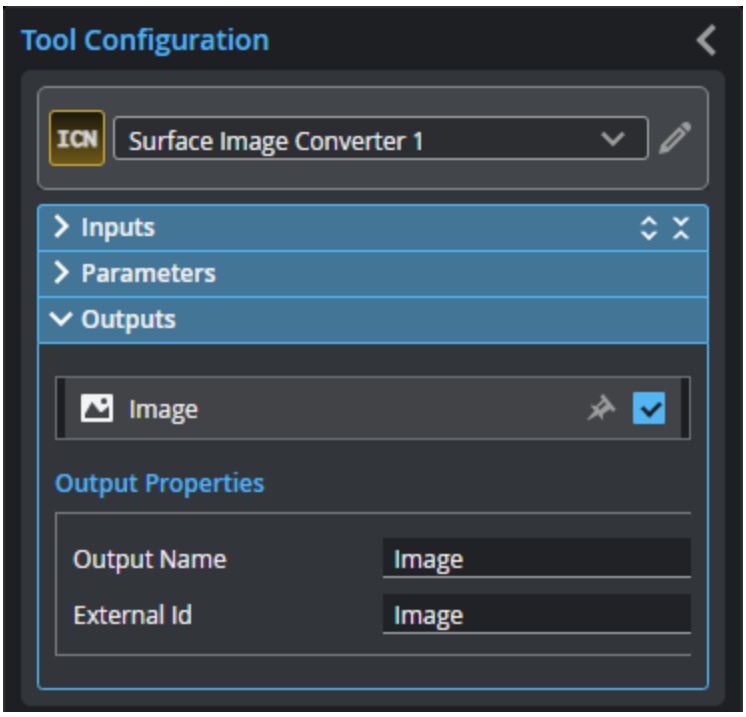
The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Mode	The tool's conversion mode. One of the following: **Heightmap to Image**: When **Mode** is set to this, set **Pixel Format** as needed. **Intensity to Image**: **Acquire intensity** on the **Acquire** page must be enabled for this mode. Otherwise, the tool displays an error.
Pixel Format	The pixel format of the output Image data. One of the following: 8-bit Grayscale, 16-bit Grayscale.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



Data

Type	Description
Image	The Image output.

Surface Inference

The Inference tool lets you deploy a model you have trained outside of GoPxL using your preferred vision software and your own training workflow ("Bring Your Own Model").

Once you've uploaded your model to the tool and selected it in the **Model** drop-down, the tool runs your model on each frame it receives from a sensor or smart camera, and returns detections or classifications, measurement values that you can configure as anchors for other, and images that can be used by other tools.

The tool does not require a Pro Tools license.

> LMI has a commercial agreement with Ultralytics that lets you deploy models trained using Ultralytics open-source code in GoPxL for commercial use without needing to purchase an Ultralytics license.

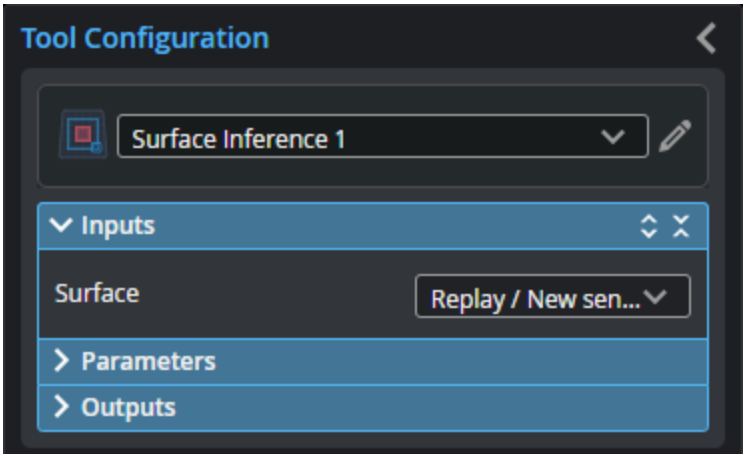
The tool supports the following model formats:

- PyTorch (.pt, .pth)
- ONNX (.onnx)
- Compiled TensorRT engines (.plan, .engine)

The tool only supports models trained for object detection and instance segmentation tasks. The model's task is automatically detected when the tool loads it.

For examples of scripts using the Inference tool outputs, see *Script Tool* on page 670.

Inputs



Inputs

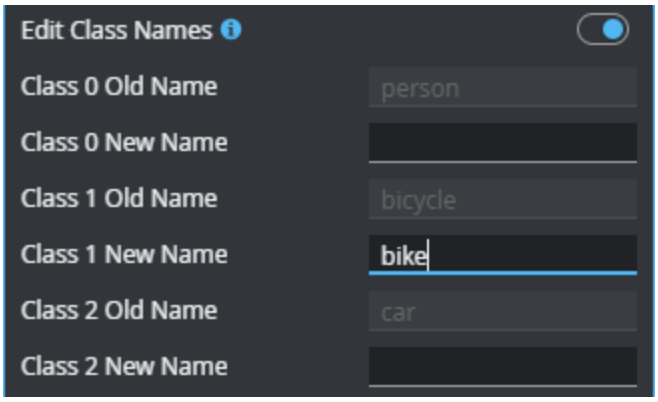
Name	Description
Surface Input	The data the tool applies measurements to or processes.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

Parameters

Parameter	Description
Model	Selects an uploaded model for inference. After you select a model, the tool loads it and displays the recognized model type in the **Detected Task** field. Click Import ( ) to upload a model from your PC. Click Download ( ) to download a model to your PC. Click Delete ( ) to delete the selected model.
Format	Drop-down list of available model formats in the dedicated storage folder for the selected model. If other model formats (.pt, .pth, .onnx, .plan, or .engine) exist in the dedicated storage folder for your model, this drop-down lets you switch between the imported format. Note that the tool performance is affected by which format you choose. If there are no optimized models available in the dedicated storage folder, an **Optimize** button will be displayed.
Detected Task	A read-only field displaying the task the loaded model was trained to perform. One of the following: Detection or Segmentation. If the type can't be determined, this field displays "Auto / Unknown".
Confidence Threshold	The minimum confidence score for a detection or segmentation to be reported.
Overlap Threshold	Helps reduce duplicate detections, due to overlapping bounding boxes. (Only displayed if the model supports duplicate suppression.) The value is between 0 and 1. With a lower value, the tool will remove duplicates more aggressively. A higher value is more permissive of overlap. Two bounding boxes that overlap perfectly would have an overlap value of 1.0. Two bounding boxes with almost no overlap would have an overlap value close to zero.
Optimize button	Click this button to optimize the model selected in **Model**. For more information, see *Optimizing Models* on page 890.
Preprocessing	How the tool fits the input frame to the model's input size. Match this to the preprocessing strategy the model was trained with. **Stretch (default)** The frame is resized directly to the model's input size, ignoring aspect ratio. Nothing is cropped or padded, so all content survives, but objects change shape (squashed or stretched) whenever the frame's proportions differ from the model's. Use this with models trained on stretched images. **Fill (center crop)** The frame is scaled up until it completely covers the model's input, then the overflow is cropped away symmetrically around the center. The frame is not distorted, no padding is applied, and data in the center is at full resolution. However, anything near the frame's edges is discarded before inference, so

Parameter	Description
	objects there can be missed entirely. **Fit** The frame is scaled down, preserving aspect ratio until it fits inside the model's expected input size. The padding needed to reach the model's exact expected input size depends on the model family (for example, Ultralytics YOLO typically pads with gray). The frame is therefore visible to the model at correct proportions and the padding color is whatever the backend chooses, not yours. **Fit and Pad** Scales the frame down and preserves the aspect ratio as with the Fit method, but the tool pads the result to the model's exact input size using the color from the **Padding** parameter (see below). Use this when you know what padding color the training pipeline used and want it reproduced exactly. **Tiling** The frame is never resized as a whole. Instead it's cut into tiles that the model runs on one by one (or in batches when **Batched Tiles** is enabled), and the per-tile detections are merged back into full-frame coordinates. For information on the related parameters, see *Tiling Parameters* on the next page.
Padding (with "Fit and Pad" preprocessing)	The color of the padding that the tool adds around the scaled frame. One of the following: Black, White, or Gray. The default is Gray, which matches the letterbox padding used during typical model training.
Edit Class Names	When this parameter is enabled, you can customize the names of the classes in the loaded model.  Class names are changed when you press Enter or move the cursor out of the field. The new class names are used in outputs, but the model is not modified.
Input Mode	How the Surface data is converted to the 2D image passed to the model. Choose the mode that most closely matches the images the model was trained on.

Parameter	Description
	Blended (Height + Intensity): The heightmap is color-coded by height and blended with the intensity image. Best for general-purpose models. **Intensity Only**: The surface's intensity image is used directly, like a conventional 2D camera image. **Height Map Only**: The heightmap is color-coded by height. Intensity is not used.
Polygon Order	Controls how many points the tool generates along each side (left/right) of a detected polygon's centerline when producing PolylineLeft/PolylineRight outputs. Higher values give a denser, smoother polyline that follows the polygon edge more accurately while lower values produce a coarser, faster polyline. Shown only when a segmentation model is loaded and polygon outputs are enabled from the output list.
Sample Points Per Polygon	The number of many points the tool generates along each side (left/right) of a detected polygon's centerline when producing Polyline Left or Polyline Right outputs. Higher values give a denser, smoother polyline that follows the polygon edge more accurately. Lower values produce a coarser, faster polyline. Only shown with segmentation models and if **Max Detections** is set to a value greater than 0.
Max Detections	Maximum number of detected regions (0-20, default 0) for which 3D polygon and corner outputs are produced. One set of outputs (Polyline Left/Right, Top/Bottom Left/Right) is created per region, numbered 1 to the value you provide. Set to 0 to disable the 3D polygon outputs entirely. Shown only when a segmentation model is loaded.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Tiling Parameters

Parameter	Description
Tiling Mode	When **Preprocessing** is set to Tiling, set **Tiling Mode** to "Sliding Window" or "Grid" (see below), and optionally set an overlap (**Tile Overlap**). **Sliding Window** The tool moves a tile you define in pixels in **Tile Width** and **Tile Height** across both axes. **Grid** Divides the frame into a grid you define using the **Rows** and **Columns** parameters. Use this mode for small defects in large, high-resolution frames, where downscaling the whole frame to model size would remove the features you want to detect.

Parameter	Description
Tile Width Tile Height or Rows / Columns	When **Tiling Mode** is set to Sliding Window, the width and height of each tile in pixels (default 640). If possible, you should use the width and height of the images the model was trained on. When **Tiling Mode** is set to Grid, the number of tiles the frame is divided into along each axis, using the **Rows** and **Columns** parameters.
Tile Overlap	How much each tile overlaps its neighbors, as a percentage of the tile size (0-95%, default 0%). Increase the value for more seamless detection and segmentation around tile boundaries, at the cost of more tiles per frame.
Tile Geometry	Read-only field that describes the current tile layout the tool uses on the input. In Grid tiling mode, it shows the resulting tile size in pixels. In Sliding Window tiling mode, it shows the resulting number of rows and columns.
Batched Tiles	When this parameter is enabled, the tool passes tiles to the model in groups instead of one at a time, which can greatly reduce per-frame inference time. The parameter is only available if the loaded model supports batched input. Set **Max Batch Size** to the maximum number of tiles the tool passes to the model. The upper limit is set by the loaded model.
Max Batch Size	Maximum number of tiles processed together when **Batched Tiles** is enabled. The upper limit is set by the loaded model. This option is available for YOLO-family models only. To find the best batch size for your setup, first pick the tile size based on what the model was trained on, then raise this parameter gradually until you hit memory limits or throughput stops improving.

Optimizing Models

When a .pt, .pth, or .onnx model is selected in **Model**, an **Optimize** button appears. You should always optimize a model after you have uploaded it.

Some models (such as PyTorch or ONNX models) are uncompiled. Optimizing these models compiles them for the hardware of your GoPxL device (a sensor, GoMax device, or PC), and ensures they run faster.

Other models (.plan and .engine) are compiled but for a different device and may not load or run properly on a GoPxL device. Optimizing these models recompiles them for your hardware.

Optimization runs in the background and may take 5 to 15 minutes. While the tool optimizes the model, the **Optimize** button is greyed out and displays "Optimizing...". When optimization finishes, the tool automatically switches to the compiled model, and the button disappears. You should wait until the optimization is complete before continuing to configure your tool chain.

Outputs

All outputs provide an external ID (available by expanding the output in the Outputs panel) for optional use in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Note that some measurements return IDs, whereas others return class names. The IDs correspond to the class names added in the project. So for example, given the classes "lays classic", "cheetos", "doritos", and "ruffles", the corresponding IDs are 0, 1, 2, and 3. The ID values are ordered based on the order in which the classes were added in the project.

Measurements

Measurement
Count The number of detected or segmented objects.
Class IDs An array of the class indexes the model predicts for each object.
Class Names An array of the class names the model predicts for the objects.
Confidence Scores An array of the confidence scores for each detection or segmentation.
Bounding Boxes X (center) **Bounding Boxes Y (center)** Arrays of the X and Y positions of the center of the bounding boxes fitted to each object.
Bounding Boxes Width **Bounding Boxes Height** Arrays of widths and heights of the bounding boxes fitted to each object.
Major Axis **Minor Axis** An array containing the lengths of each detection along their major and minor axis. For segmentation models, the measurements are fitted to the segmented shape. For detection models, they are the bounding box's longer and shorter sides.

Features

Type	Description
Bounding Box Center Points	An array of Point geometric features representing the center the fitted bounding boxes.
Major Axis Line	An array of Line geometric features representing the angle of the major axis of the segmented objects. Only available with segmentation models.

Data

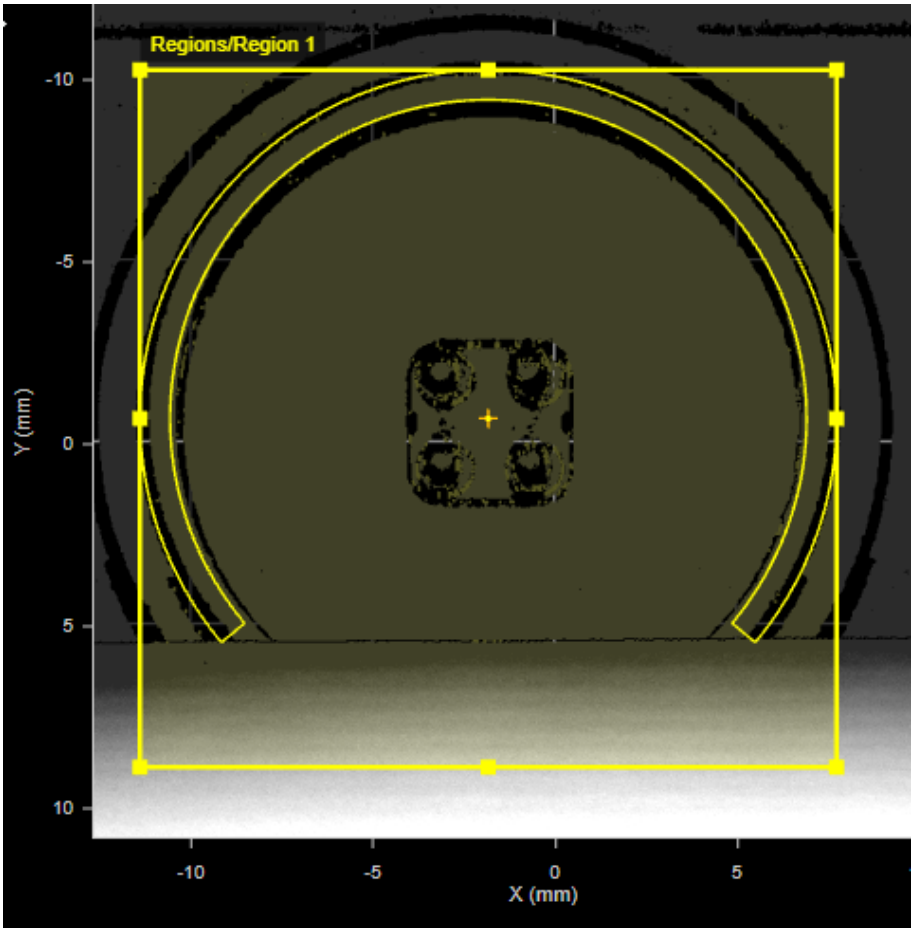
Type	Description
Annotated Image	The frame's image with an overlay displaying the class names, bounding boxes, and confidence scores of the detected or segmented objects.
Polyline Left {n} Polyline Right {n}	Arrays of 3D points sampled along the left and right boundaries of detected region {n}, in the surface's units. The number of points per boundary is set by **Sample Points Per Polygon**. Only shown with segmentation models and if **Max Detections** is set to a value greater than 0.
Top Left {n} Top Right {n} Bottom Left {n} Bottom Right {n}	3D points marking the corners of the smallest box that encloses detected region {n}, oriented along the region's own centerline. When there are fewer detections than the value provided in **Max Detections**, the remaining indexes are empty or invalid. Only shown with segmentation models and if **Max Detections** is set to a value greater than 0.
Mask Image	The detected regions are drawn as colored areas on a black background. (Segmentation models only.)
Detection Region Images	An array of cropped images of the detections, taken around the detection's bounding box. With a segmentation model, the pixels outside the segmented target in each region image are zeroed according to the segmentation mask. Useful for feeding images of individual detections to other tools in the tool chain.
Inference Input	The actual image (or array of images when Tiling is used in **Preprocessing**) on which the tool runs the model. The image includes any preprocessing in the tool such as padding. Use this to verify what the model receives as input.
Preview Image	The 2D view of the surface passed to the model. Useful for verifying the effects of **Input Mode**.

Surface Mask

The Surface Mask tool lets define up to 16 regions to extract data from a surface. Each region's size, position, and shape (circular, elliptical, 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. Extracted data is output in a single surface.

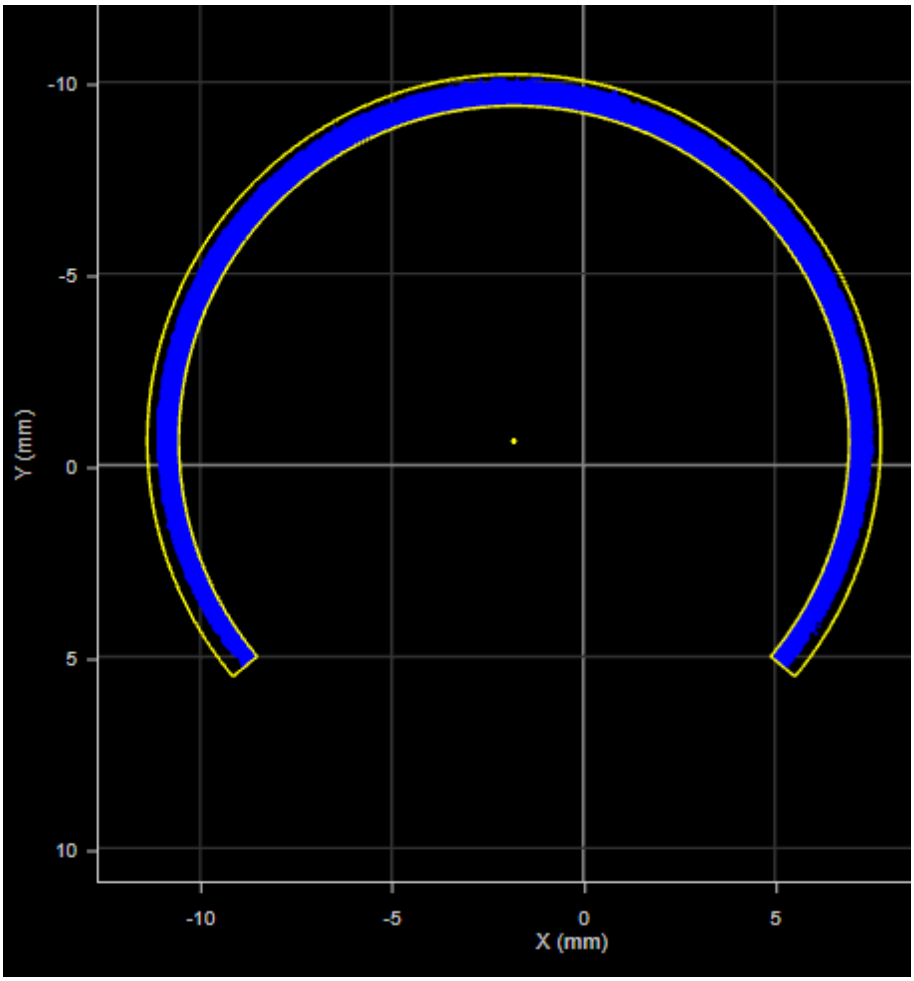
The resulting surface can then be further processed or measured by other tools.

For example, given the following scan data:



A circle region box containing a partial ring (cyan)

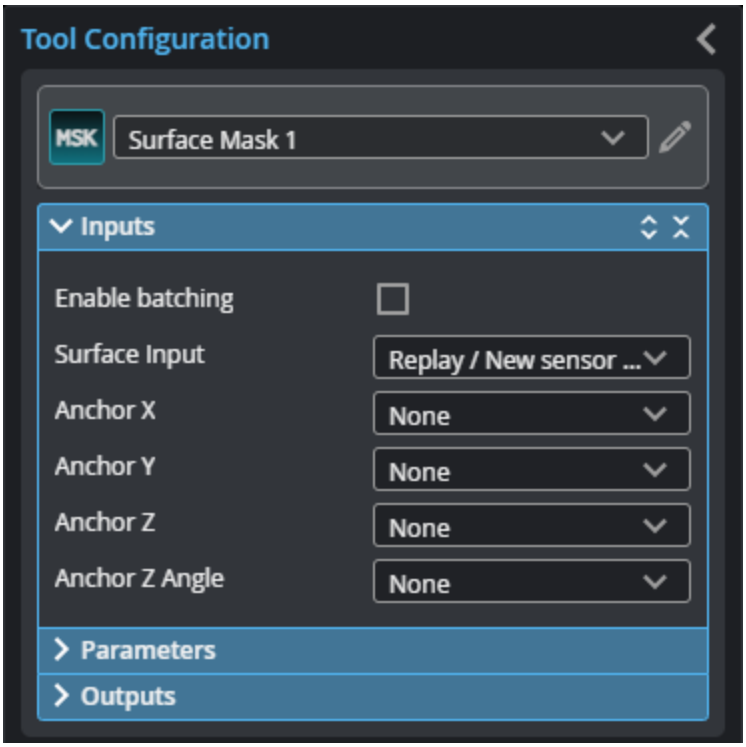
The image below shows the extracted data. The extracted surface data can then be further processed by other tools, or measurements can be applied to the surface data.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



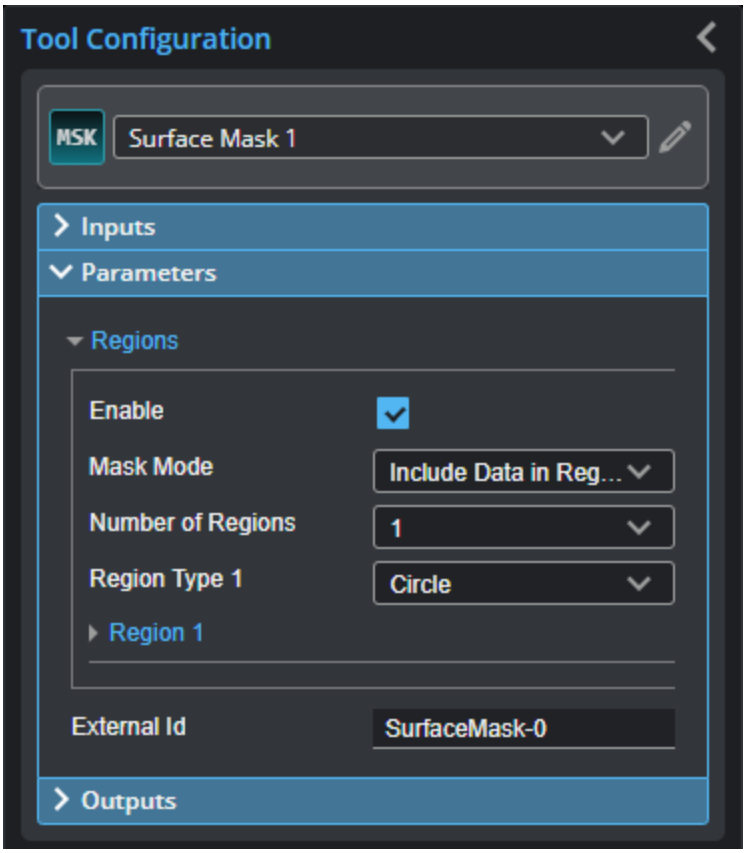
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

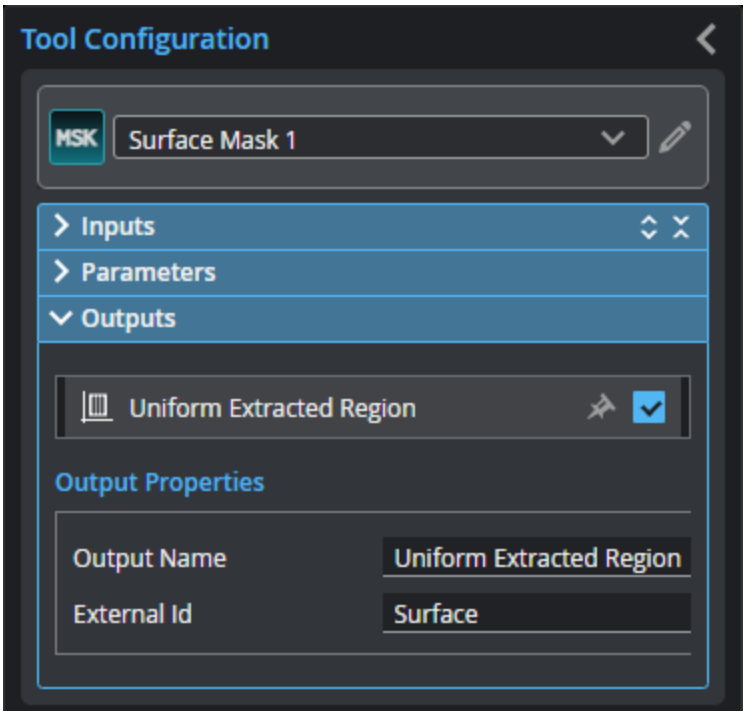


Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Data

Type	Description
Uniform Extracted Region Point Cloud Extracted Region	The surface containing the extracted region or regions. (Uniform and point cloud data, respectively.)

Surface Merge Wide

This tool is only intended for use with a multi-sensor G2 (line profile) system.

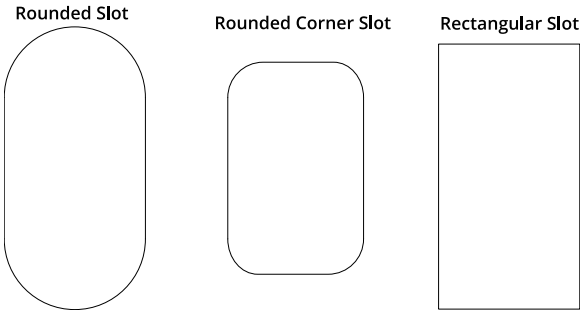
Surface Mesh

This tool is only intended for use with a multi-sensor G2 (line profile) system.

Surface 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.

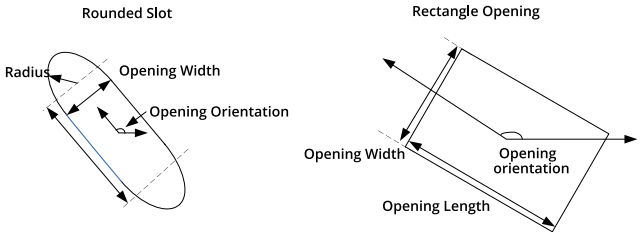


The tool uses a complex feature-locating algorithm to find a hold and then return measurements. For a detailed explanation of the algorithm the tool uses, see *Opening Algorithm* on the next page. The behavior of the algorithm can be adjusted by changing the tool's parameters.

The tool 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.

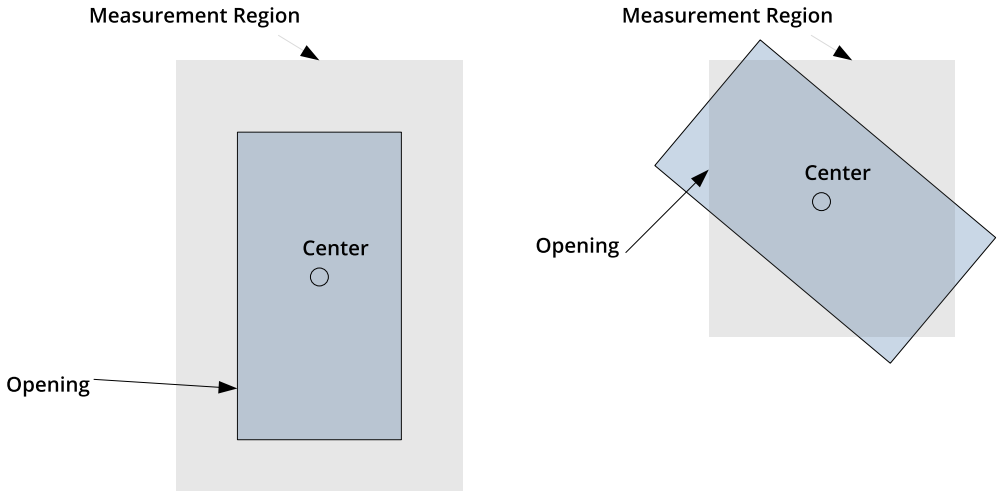


Measurement Panel

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

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.



Opening Algorithm

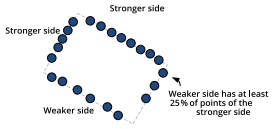
The Opening tool processes the data in three phases: Search, Measure, and Filter.

See the tool's parameters for an explanation of the options that affect the tool's algorithm.

Search phase - The tool searches for coarse data transitions (edge data) and performs a coarse fitting of the opening shape (specified by the orientation angles and the nominal dimensions) to determine the most likely candidate. If **Tilt Correction** is enabled, the algorithm uses the flat surface in the measurement region to estimate the orientation of the part.

Measure phase - A more rigorous edge detection algorithm is applied to precisely determine the edges around the feature. Edge detection at this stage will reject outliers and noise. The algorithm requires opposite sides and ends to be associated with a comparable number of edge pixels, with the weaker side or end having at least 25% of the stronger.

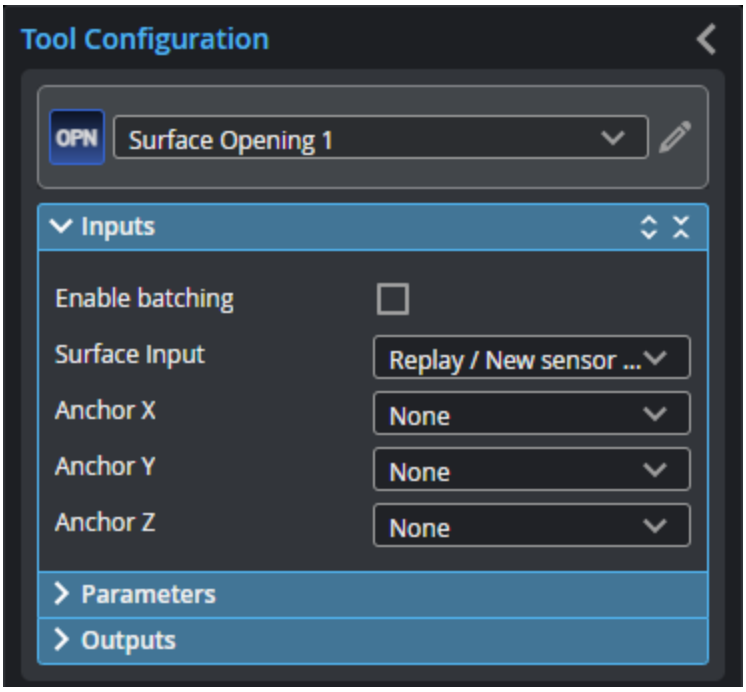
The set of refined edges is then used to locate and inspect the feature. If the **Reference Regions** setting is enabled, the edges are also used to calculate the location of the reference regions.



Filter phase - The detected location and dimensions are compared to the nominal and tolerance settings. If the refined feature falls within the measurement region and its measurements fit within the specified tolerance, the results are reported. If not, an invalid result is returned.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



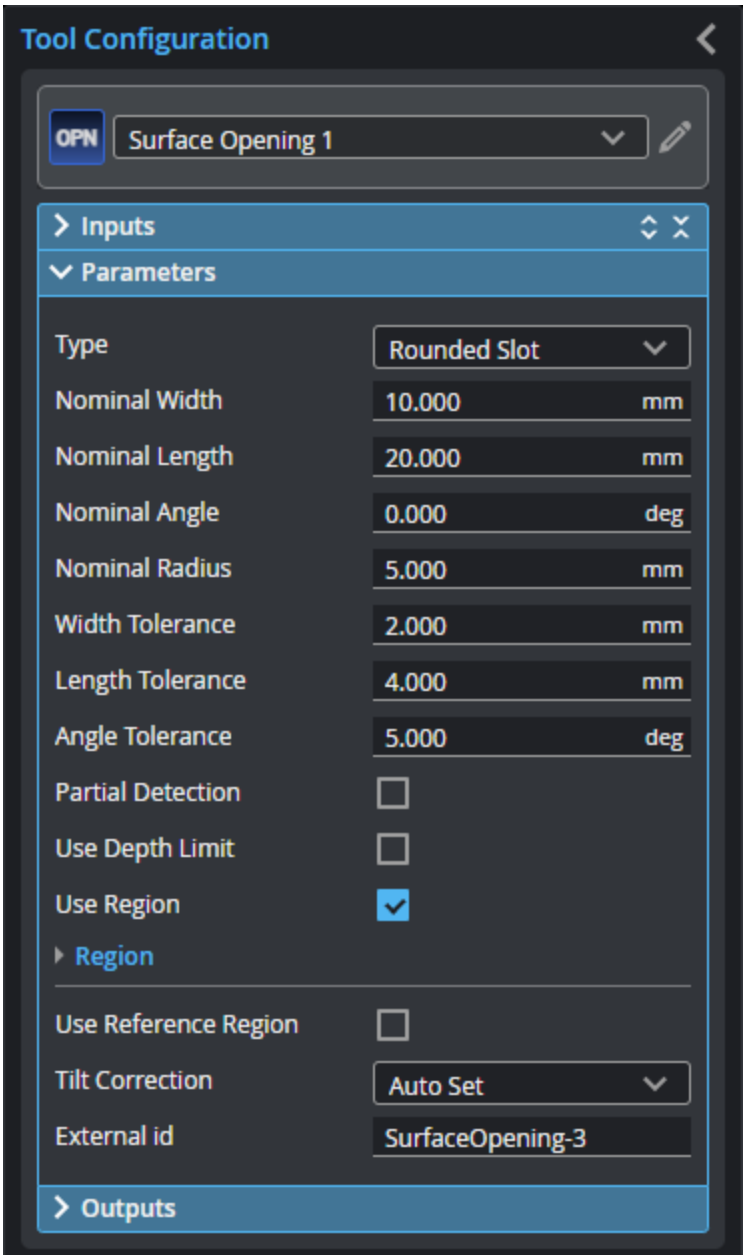
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.

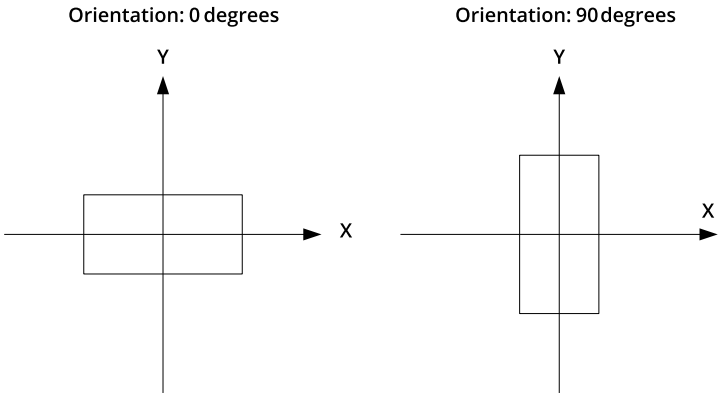
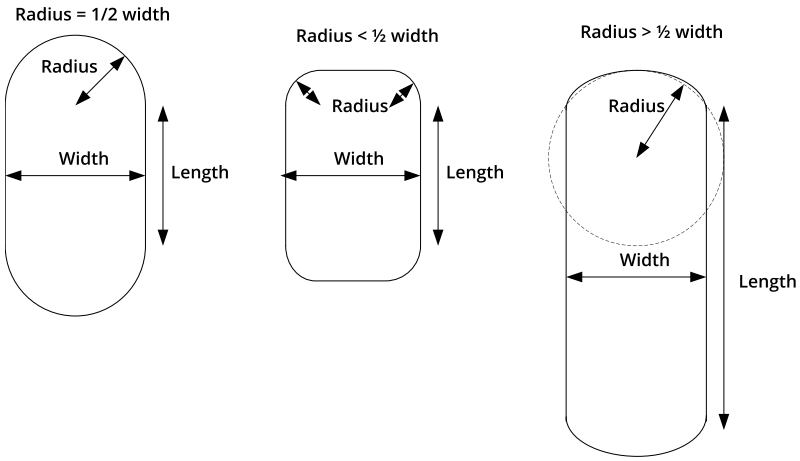
Parameters

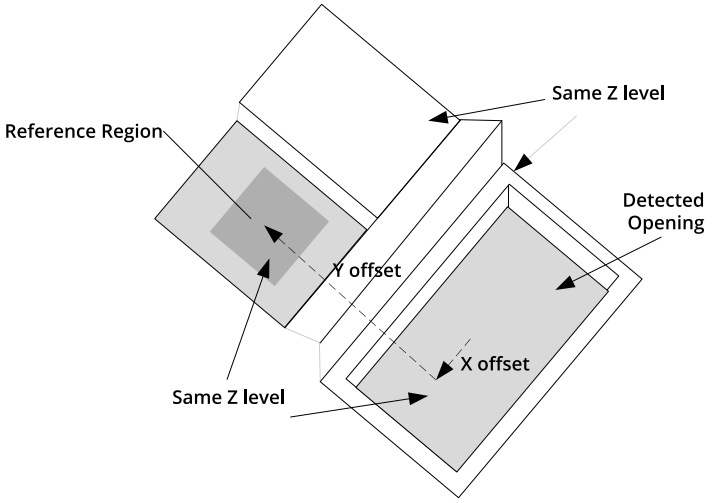
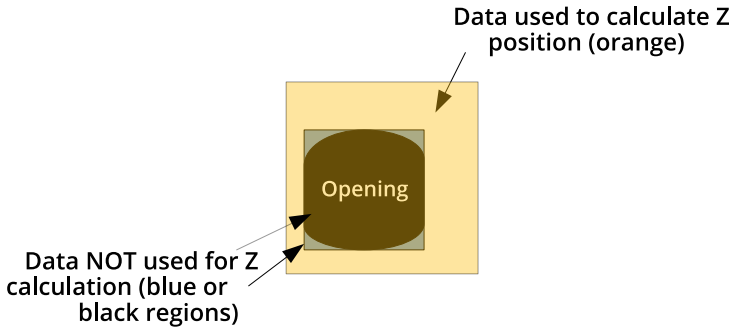
You configure the tool's parameters in the expandable **Parameters** section.



Parameters

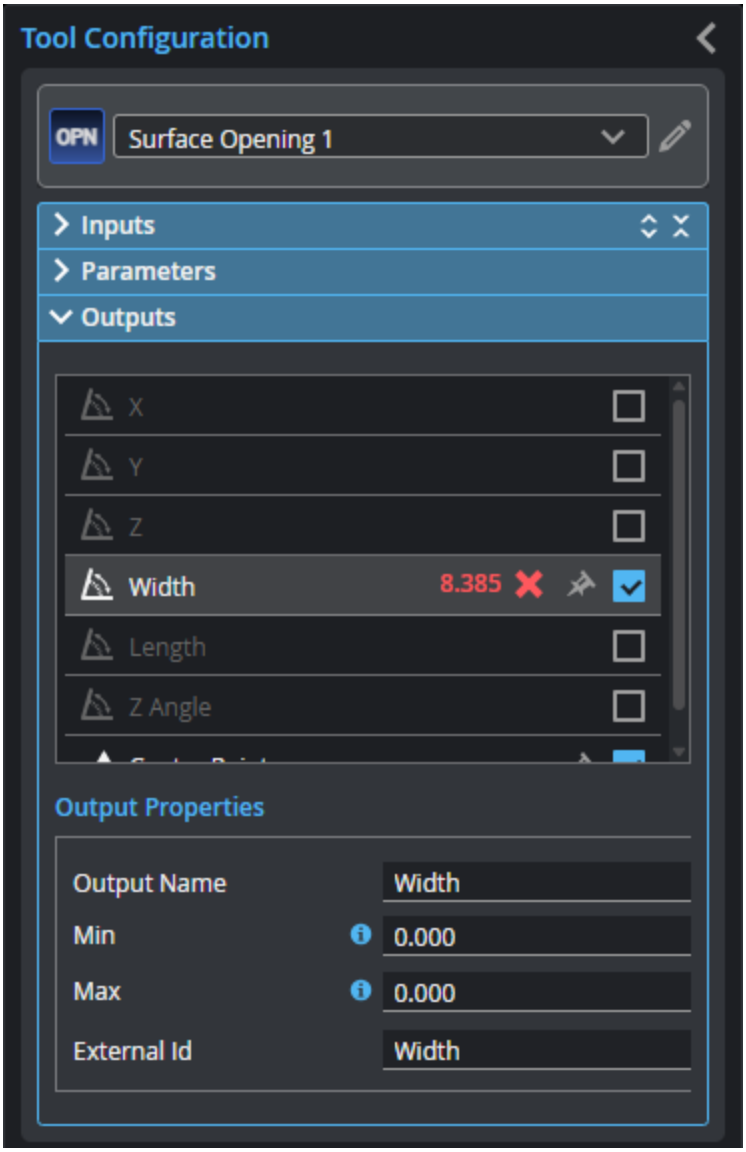
Parameter	Description
Type	One of the following: Rounded Slot, Rectangle.
Nominal Width	Nominal width of the opening.
Nominal Length	Nominal length of the opening.
Nominal Angle	Nominal angle of the opening. The default orientation is the length of the opening along the X axis.

Parameter	Description
	
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 Length Tolerance Angle Tolerance	The maximum variation from the nominal width, length, and angle (+/- 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.
Use Depth Limit Depth Limit	When **Use Depth Limit** is enabled, data below the value set in **Depth Limit** (relative to the surface) is excluded from the opening calculations.
Use Region	If enabled, displays an expandable region section where you define the region.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
Use Reference Region Reference Type	When **Use Reference Region** is enabled, the tool uses 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

Parameter	Description
Reference Region {n}	surface around the opening is not flat.  When the Reference Regions setting is disabled, the tool measures the opening's Z position using 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).
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

Parameter	Description
	the **X Angle** and **Y Angle** parameters of this tool. For more information, see Plane.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

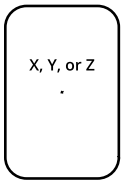
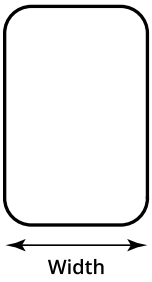
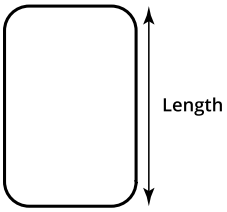
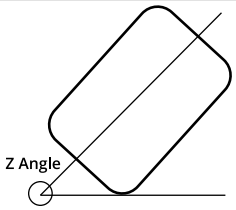


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.	
Length Determines the length of the opening.	
Z 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 227.

Surface Part Detection

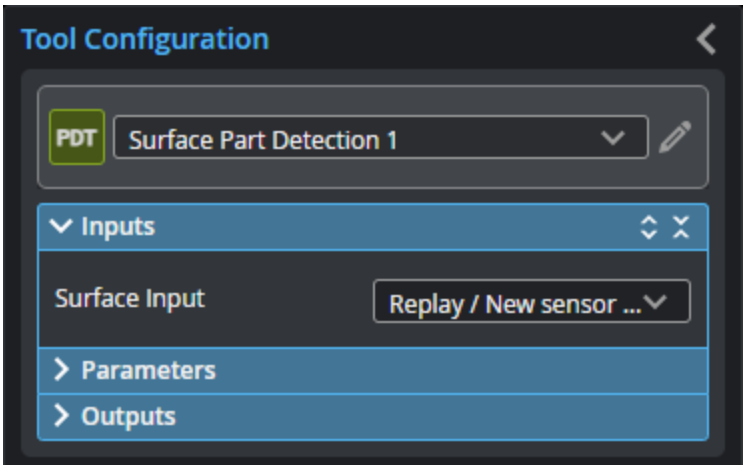
The Surface Part Detection tool identifies discrete objects in Surface data and then outputs them as Surface data. Surface measurements can then be performed on each object.

> You can isolate and then measure parts using Surface Blob and Surface Segmentation tools (for more information on these tools, see *Surface Blob* on page 760 and *Surface Segmentation* on page 951). For a comparison of part detection and these tools, see *Isolating Parts from Surface Data* on page 712.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

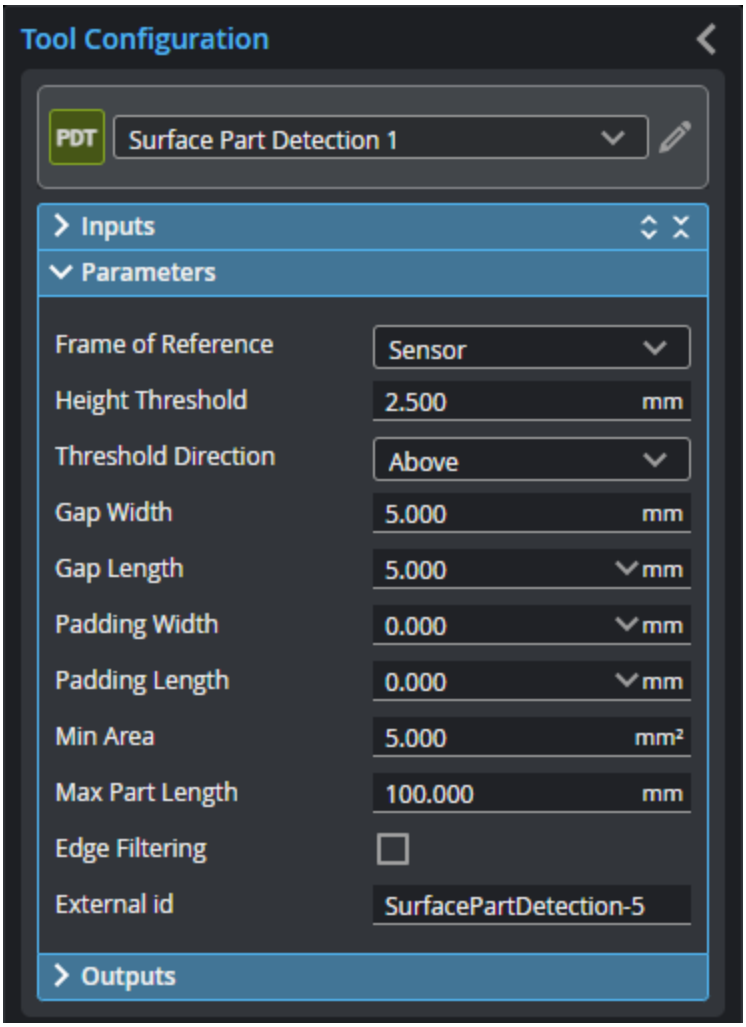


Inputs

Name	Description
Surface Input	The source of data for the tool.

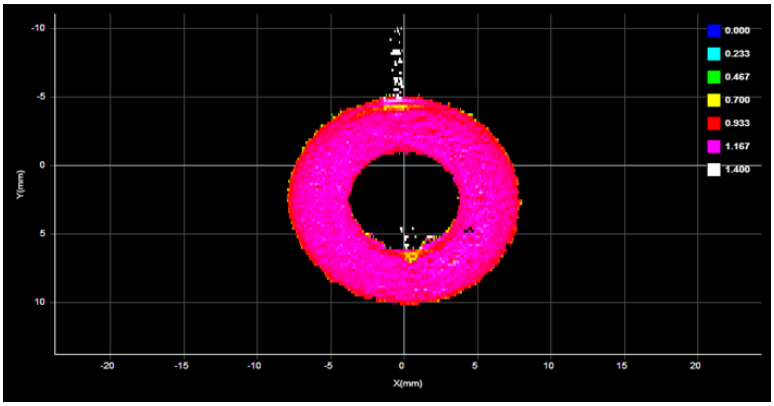
Parameters

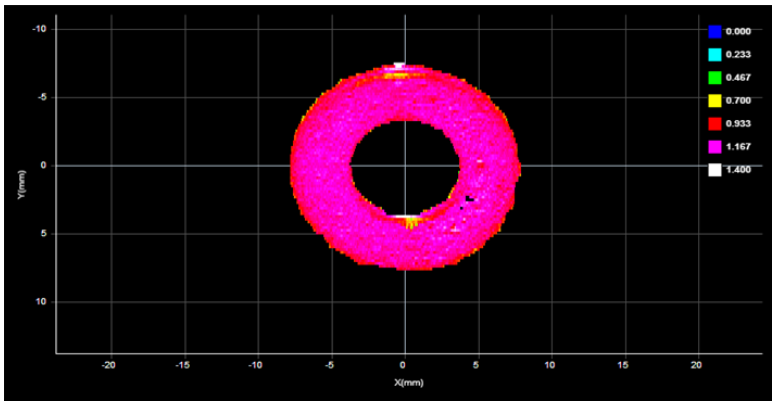
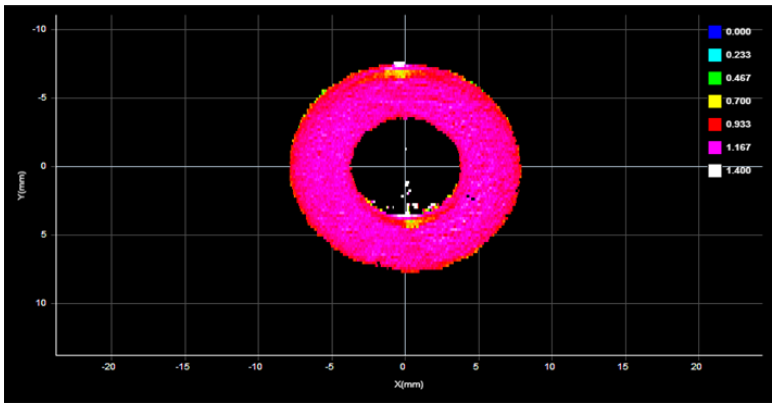
You configure the tool's parameters in the expandable **Parameters** section.



Parameters

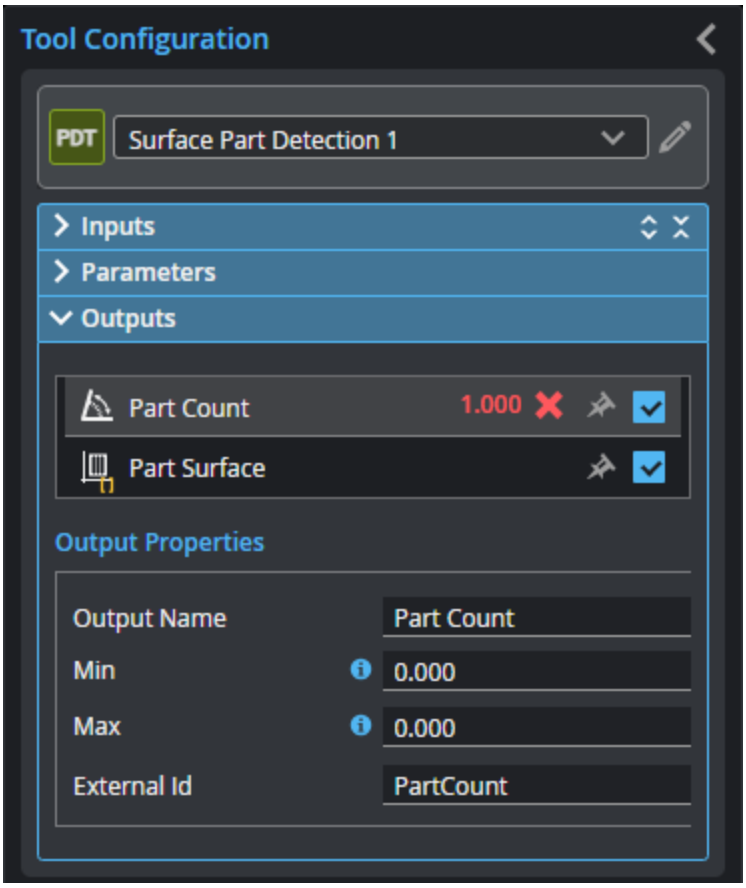
Parameter	Description
Frame of Reference	Determines the coordinate reference for surface measurements. **Sensor** When **Frame of Reference** is set to **Sensor**, the sensor's frame of reference is used. **Part** When **Frame of Reference** is set to **Part**, all measurements are relative to the center 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 *Surface Bounding Box* on page 774).
Height Threshold	Determines the height threshold for part detection. The setting for **Threshold Direction** (see below) determines if parts should be detected above or below the

Parameter	Description
	value in **Height 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 Gap Length	**Gap Width** and **Gap Length** determine the minimum separation between objects on the X and the Y axis, respectively. If parts are closer than the gap interval, they will be merged into a single Surface output.
Pad Width Pad Length	These parameters are useful when processing part data with third-party software such as HexSight, Halcon, etc. **Pad Width** and **Pad Length** control the amount of additional scan data output 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.
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 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.
Edge Filter	When **Edge Filter** is enabled, an **Edge Filter** expanding section is displayed that contains the settings described below. 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. (The following screenshots are from a G2 sensor.)  *Edge Filter disabled (scan shows reflection noise)*

Parameter	Description
	 *Edge Filter enabled (reflection noise eliminated or reduced)*  *Edge Filter enabled, Keep Interior enabled*
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Measurements

Measurement
Part Count The number of parts the engine detects.

Data

Type	Description
Part Count	The number of parts detected.
Part Surface	The Surface data of the detected part.

Surface Pattern Matching

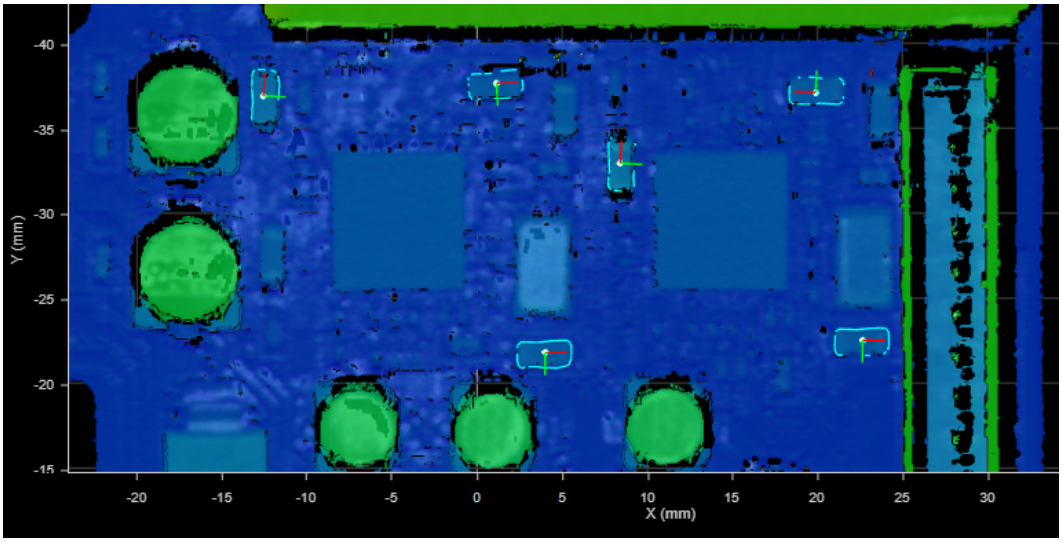
The Surface Pattern Matching tool locates parts and features by comparing 2D contours (on the XY plane) found in scan data to pattern models you have defined; note that the tool does not use height data in its algorithms. Models represent contour-based "golden parts" or "golden features." (Models can be modified in the provided standalone model editor; for more information, see *Pattern Editor* on page 1307.)

The tool can process multiple occurrences of a part or feature in a frame of scan data. For each matching part or feature (called an instance), the tool returns X and Y position and rotation (angle) measurements, which you can use to anchor other measurements (for more information on anchoring, see *Measurement Anchoring* on page 209).

The tool also returns a point and a line geometric feature for each instance, which you can use in conjunction with Surface Transform tools to shift and rotate scan data to reliably position the target.

Finally, the tool returns a match quality that you can use as a general conformity measure for matching instances (for example, checking for dents in a target), as well as a count of located instances.

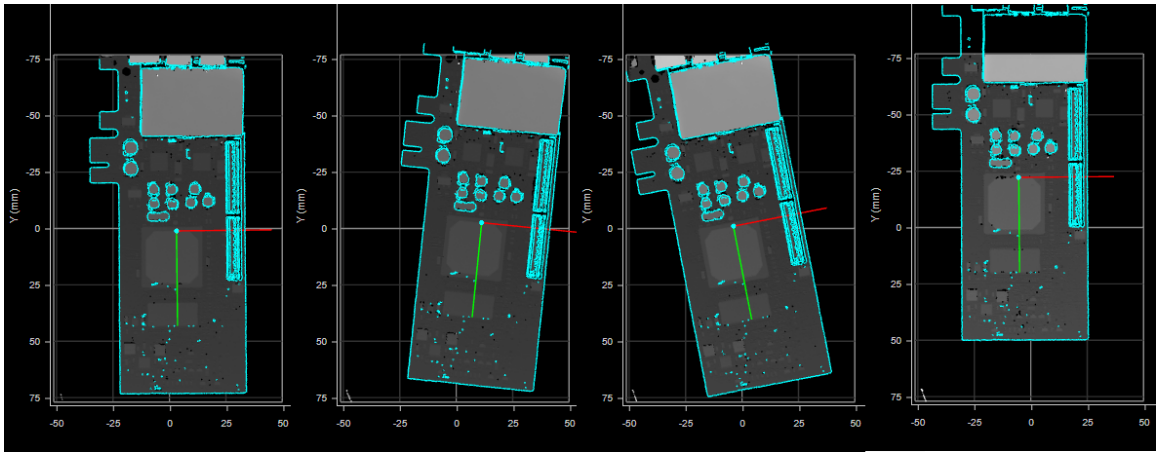
> In order to create a template for a *feature* on a target, you typically need to enable the **Use Region** checkbox to limit the tool to the contours related to that feature. After that, when running the tool to find instances of the feature, you should either modify the region to limit it to areas of the target that might contain the feature you are looking for or *disable* **Use Region** so that the tool can locate instances of the feature in all of the scan data. You can also use the **Use Region** parameter when creating a template to limit it to a unique portion of an outer edge of a target.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

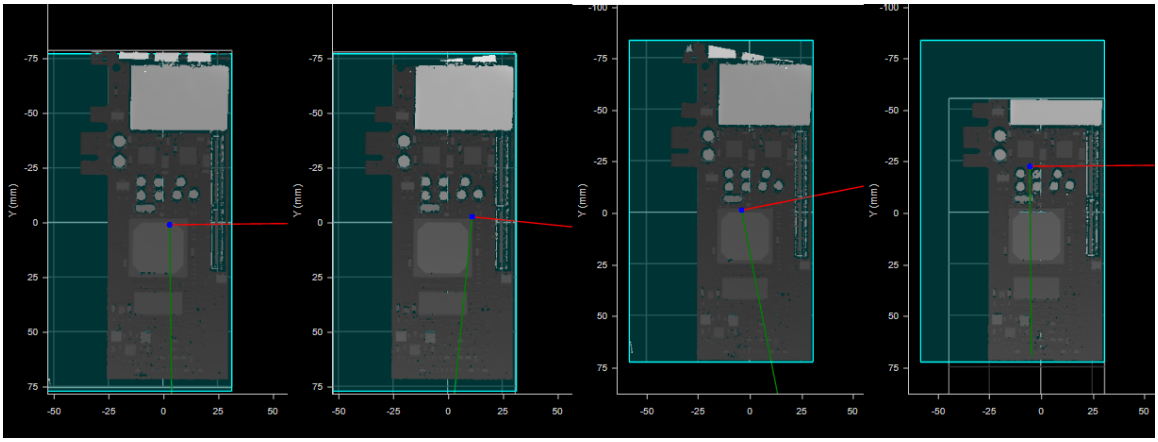
Note that when you use the geometric features with a Surface Transform tool to transform the scan data from frame to frame, you can often avoid the need to anchor other measurements, because

the transforms ensure that any features you are interested are always in the same location. This can save considerable setup time and reduce the complexity of an application. For example, in the following frames of scan data, in which a PCB shifts from frame to frame, a Surface Pattern Matching tool successfully locates the entire PCB using its outer contours and the contours of various components on the PCB, as indicated by a dark blue outline. Note the "missing" data in the second and fourth frames, on the lower right and left edges, respectively: the tool still locates the PCB, despite the occlusions.



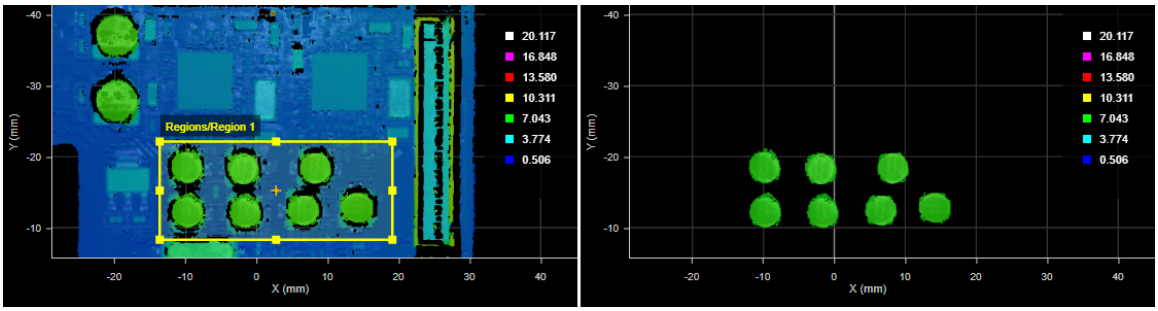
Four frames of scan data. Light blue outline represents the matching template. The first frame was used to create the template.

When the tool's Point and Line geometric features are passed to a Surface Transform tool, the transformed scan data ensures that features are always in the same location and orientation.



Transformed scan data of the four frames.

Other measurement tools can then be placed over the capacitors, without needing to anchor them. In the following image (the third frame, which was significantly shifted and rotated), a Surface Filter tool isolates the capacitors based on height. Subsequent tools can perform measurements on the isolated data to verify that all capacitors are present, are seated properly, and so on.

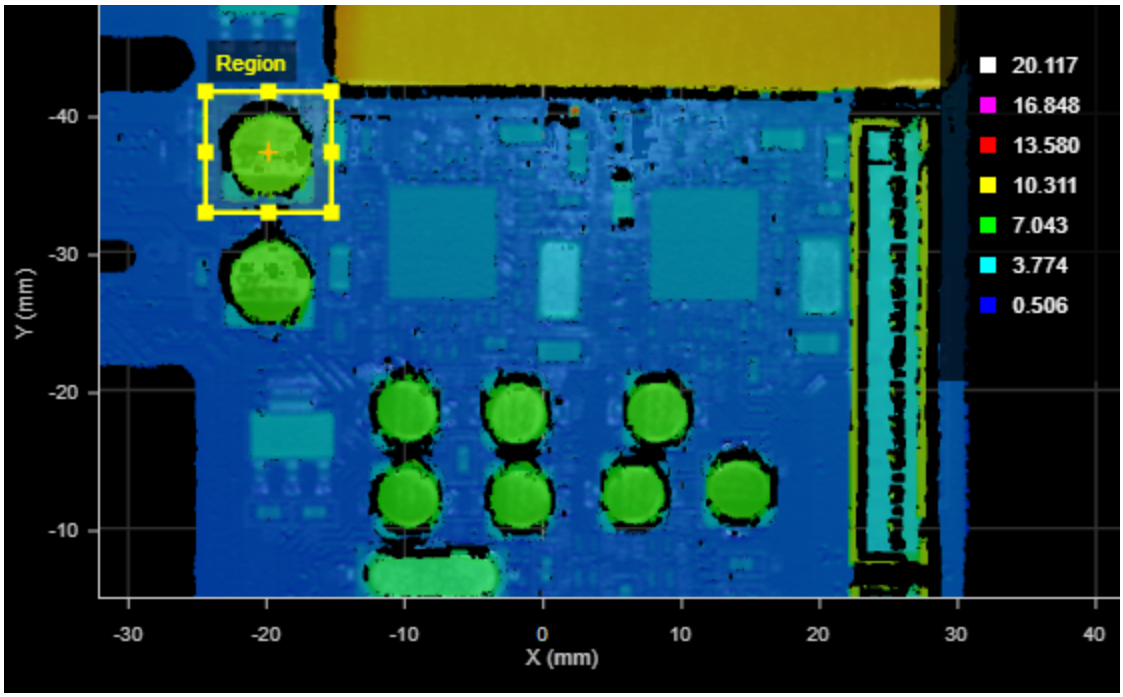


By adding multiple Surface Pattern Matching tools to a job and defining different templates for each, you can match multiple types of features or parts, for example, matching different parts moving on a conveyor, or different types of features on a single target.

Creating a Template

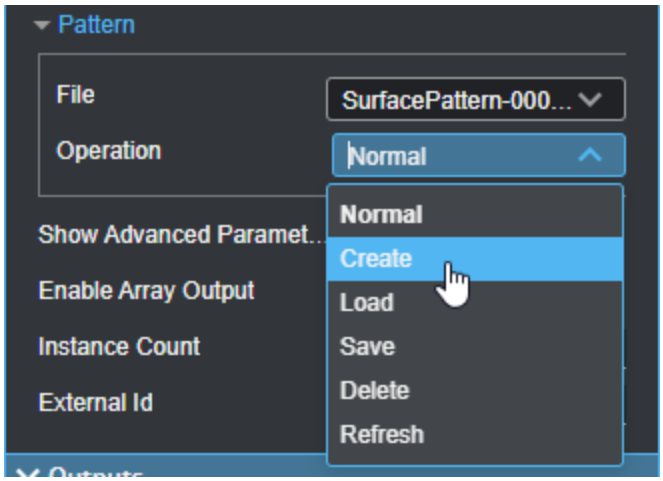
To create a template:

1. Scan a part that is representative of a "good" part (no damage, all features are present, and so on).
2. Add an instance of Surface Pattern Matching.
3. If you need to perform pattern matching on a feature on the part, enable **Use Region** and position the region over the feature.

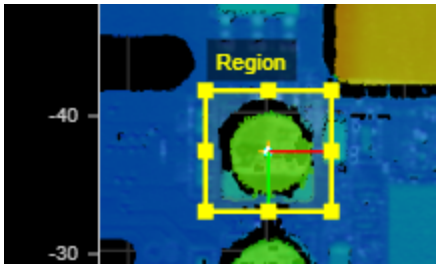


Otherwise, if you need to match the entire part, for example to orient parts from frame to frame, leave **Use Region** disabled to perform pattern matching on all of the scan data. This only works if no unwanted data is in the frame, such as the conveyor.

4. In the **Operation** drop-down, choose **Create**.



The tool creates a model and saves it either to the PC or to the sensor. The tool also "matches" the pattern.

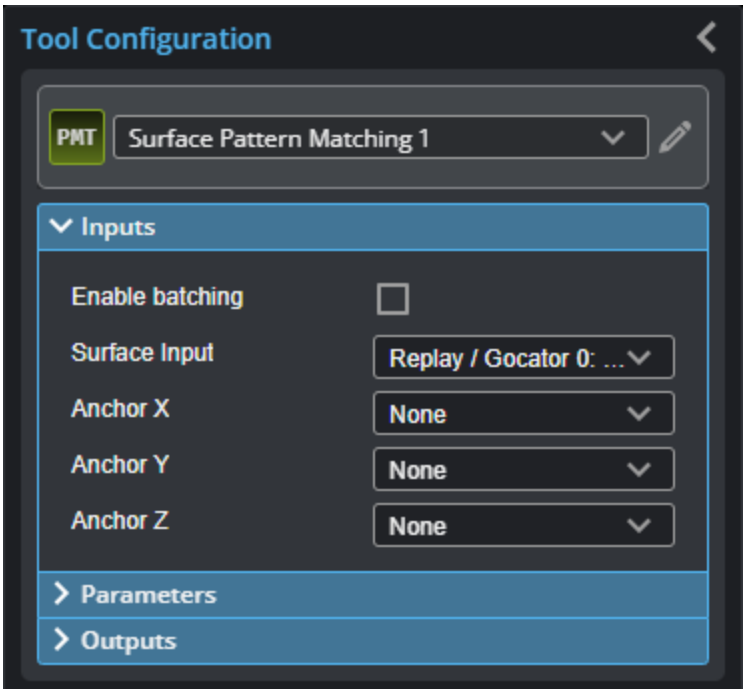


5. Disable, resize, or reposition the region as necessary to match the pattern in parts you will be scanning in production.

After creating a template, configure the tool's parameters (see below) for use during production runs.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



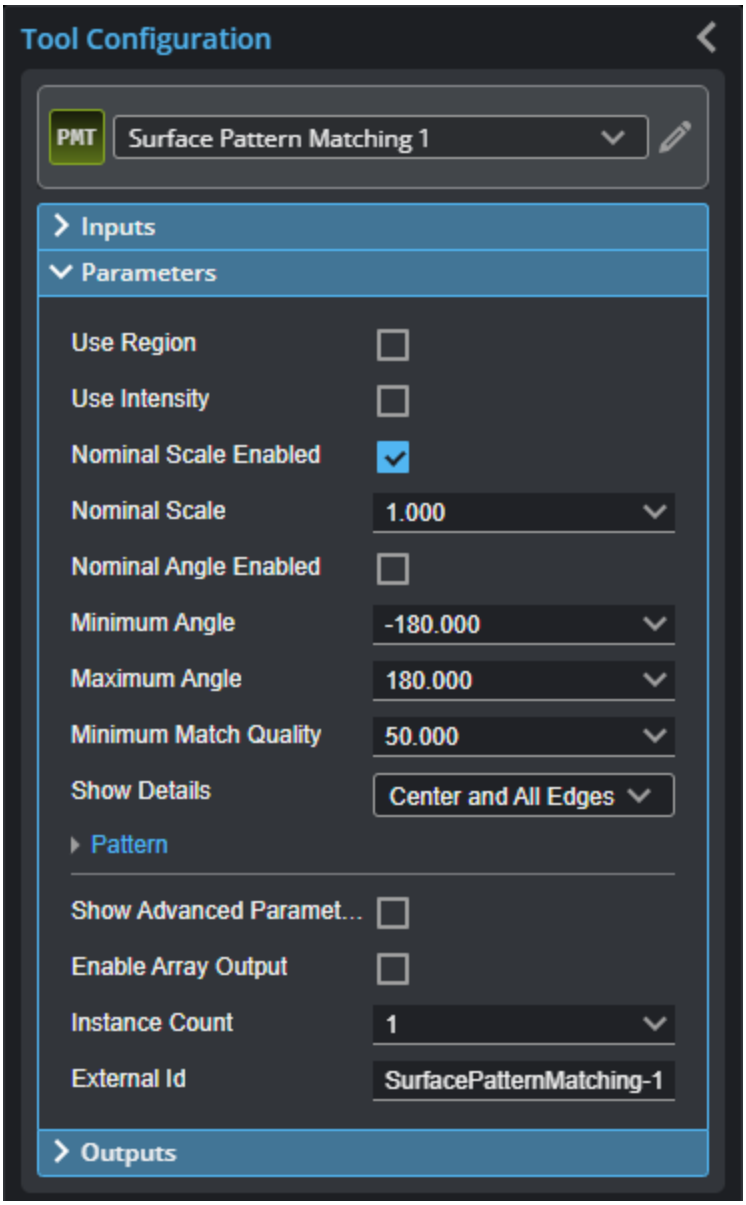
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Surface Input	The data the tool applies measurements to or processes.
Anchor X	Positional anchors are optional.
Anchor Y	
Anchor Z	

Parameters

You configure the tool's parameters in the expandable **Parameters** section.



Parameters

Parameter	Description
Use Region	Determines whether the tool uses a user-defined region to search for matching instances, or, when first creating a template, whether the tool limits establishing template contours to the data in the ROI.
Region	Settings related to the region. For more information, see *Regions* on page 214.
Use Intensity	Determines whether the tool uses intensity instead of heightmap data to locate instances or create templates.
Nominal Scale Enabled	If **Nominal Scale Enabled** is checked, the tool displays the **Nominal Scale** setting and the tool uses the user-defined nominal scale. Otherwise, the tool displays **Minimum Scale** and **Maximum Scale** settings and uses the user-defined range.

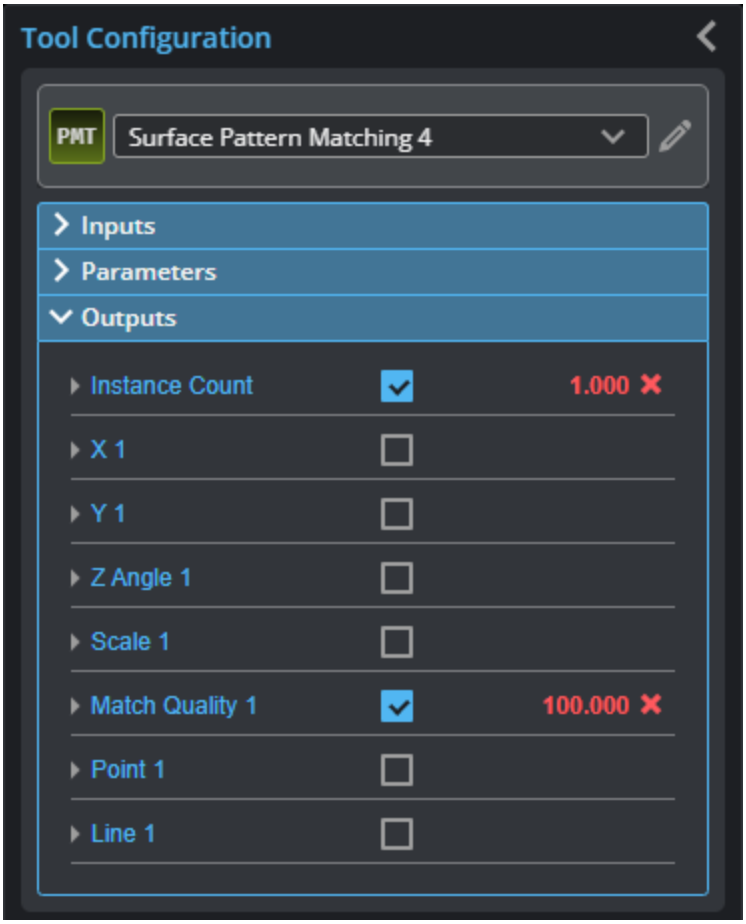
Parameter	Description
Nominal Scale	The scale factor the tool requires to recognize an instance. Displayed when **Nominal Scale Enabled** is enabled.
Minimum Scale Maximum Scale	The minimum and maximum scale factors set the acceptable scaling range of an instance. Displayed when **Nominal Scale Enabled** is disabled.
Nominal Angle Enabled	If enabled, displays the **Nominal Angle** setting and the tool uses the user-defined nominal angle. Otherwise, the tool displays **Minimum Angle** and **Maximum Angle** settings and uses the user-defined range.
Nominal Angle	The angle the tool requires to recognize an instance. Displayed when **Nominal Angle Enabled** is enabled.
Minimum Angle Maximum Angle	The minimum and maximum angles allowed for the tool to recognize an instance, respectively. Displayed when **Nominal Angle Enabled** is disabled.
Minimum Match Quality	Minimum percentage of template contours that must match in the scan data for the tool to consider the object instance as valid.
Show Detail	Toggles whether to overlay tool details. **Center only**: The center of the found pattern and its XYZ axes. **Center and All Edges**: The center of the found pattern and its XYZ axes, as well as the outline of the full pattern template. **Center and Matched Edges**: The center of the found pattern and its XYZ axes, as well as the matched edges only.
Pattern	When expanded, displays parameters related to template files. **File** A drop-down containing the currently available templates. **Operation** The operation to perform on the currently selected template in the File drop-down. One of the following: **Normal**: The default value after having performed another operation. **Create**: Creates a new template based on the current frame of scan data. Limited to the region if **Use Region** is enabled. **Load**: Loads the currently selected template. **Save**: Saves contour data to the currently selected template, overwriting its contour data. **Delete**: Deletes the currently selected template. **Refresh**: Refreshes the list of templates. This is useful if you have renamed a template (using the Pattern Editor or manually in the client PC's file system).

Parameter	Description
Show Advanced Parameters	When enabled, the tool displays the following additional advanced parameters. **Sorting** The sort order of instances the tool uses. One of the following: • Evidence: The instances are ordered according to their hypothesis strength. • Quality: The instances are ordered according to their match quality. This setting can significantly increase the tool's search time because it can't output instance results until it has found and compared all instances to determine their order. The time required to output the first instance corresponds to the total time needed to search the image and analyze all the potential instances. The time for additional instances is zero since the search process is already complete. • Position - X increasing • Position - X decreasing • Position - Y increasing • Position - Y decreasing **Recognition Level** The "effort" the tool will expend on recognizing an instance in scan data. Ranges from Fast to Accurate (that is, there is a trade-off between accuracy and speed). Only used during pattern matching (and not during pattern template creation). **Positioning Level** How accurately the tool determines the position of the instance. Ranges from Fast to Accurate. Only used during pattern matching (and not during pattern template creation). **Match Tolerance Mode** The tolerance the tool uses when attempting to match contours in the scan data to the loaded template; this is the maximum distance between the contours in the template and in the scan data. Either Pattern-based (a looser tolerance, the default), Resolution-based (a tighter tolerance), or Manual. Typically, the Pattern-based option should produce a better match quality. However, if you are getting too many false matches, try setting the mode to Resolution-based or set the value manually. The tolerances calculated from the pattern-based and resolution-based modes are displayed in the mode drop-down. Note that the resolution-based mode results in a much tighter tolerance with high resolution scan data.

Parameter	Description
	Add Border Considers a drop to NULL, outside the region, as an edge. Use this when performing part detection or when there is no data around the part. If there is nothing in the region, then there will be nothing in the template either.
Enable Array Output	When **Enable Array Output** is disabled, the tool produces individual outputs for each instance. (For example, X 1 and Y 1 for the first instance, X 2 and Y 2 for the second instance, and so on.) When **Enable Array Output** is enabled, the tool produces an array for each output. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Instance Count	The maximum number of instances the tool will locate.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement
Instance Count Returns the number of parts or features matching the loaded template up to the value set in the Instance Count parameter.
X {n} **Y {n}** The X and Y position of the center of matched instance {n}.
Z Angle {n} The angle of matched instance {n} relative to the sensor's coordinate system.
Scale {n} The scale of matched object {n} relative to the loaded template.
Match Quality {n} Percentage of matched model contours for the selected object instance. Match quality ranges from 0 to 100, with 100 being the best quality. A value of 100 means 100% of the model contours were successfully matched to the actual contours detected in the scan data. Use the **Minimum Match Quality** parameter to set the minimum acceptable value.

Features

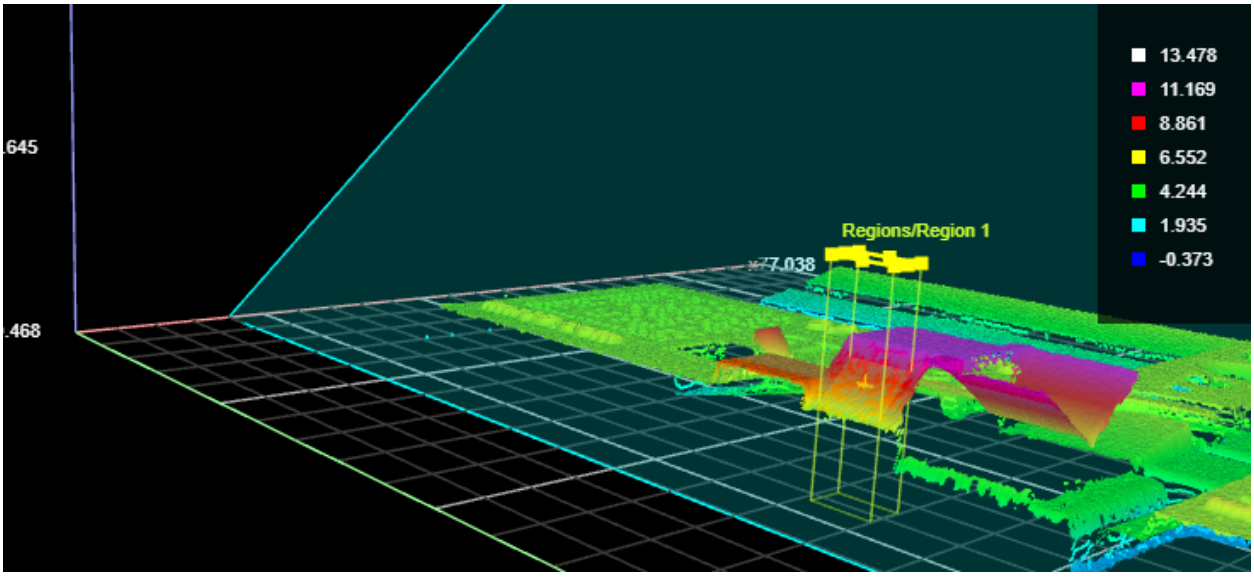
Type	Description
Point	A point representing the center of the region used when creating a template and the template's default reference point. (Note that the reference point of a template can be changed in the model editor.)
Line	A line parallel to the X axis passing through the Point feature.



For more information on geometric features, see *Geometric Features* on page 227.

Surface 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.



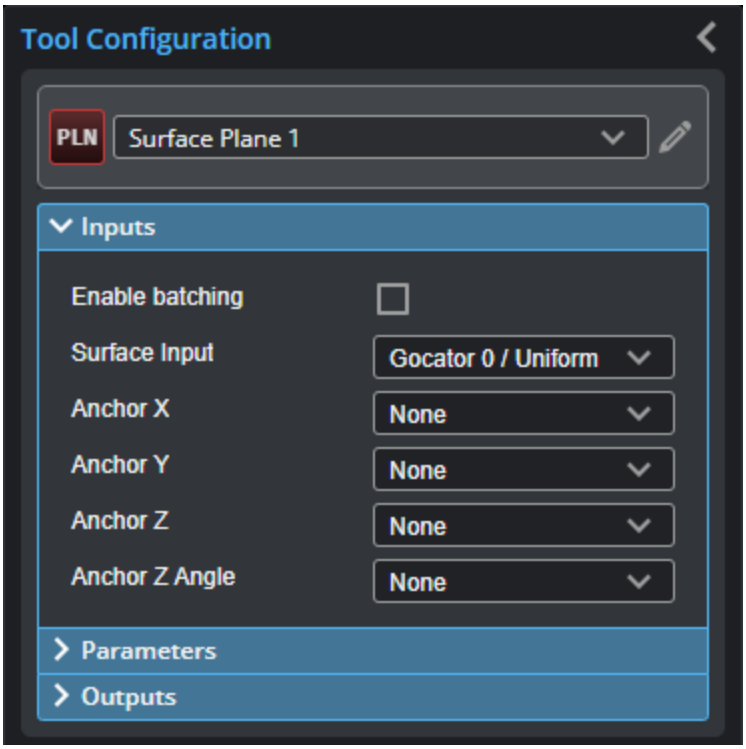
The Z offset reported is the Z position at zero position on the X axis and the Y axis.

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.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



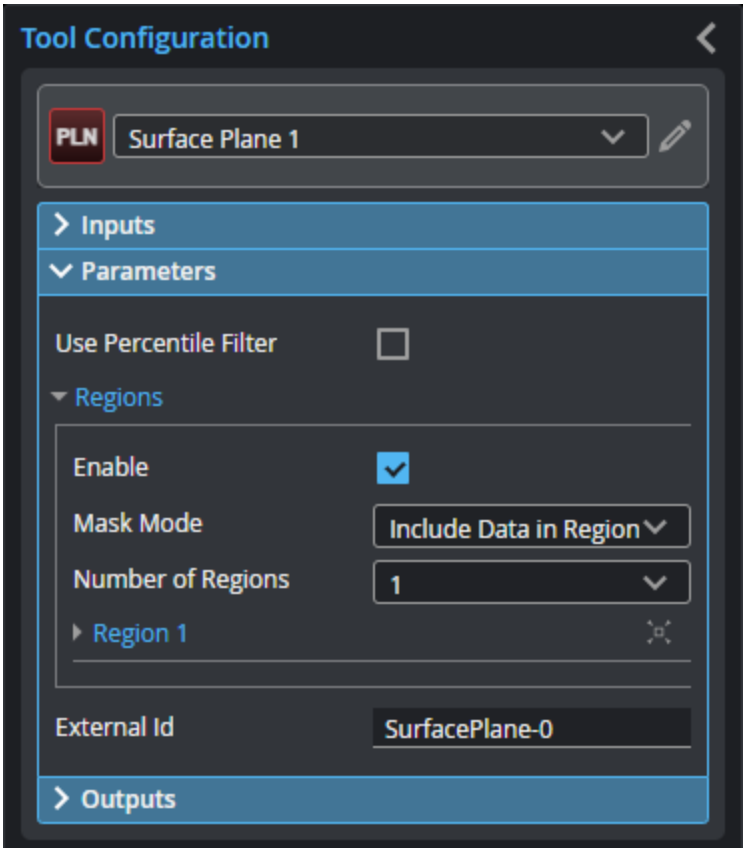
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

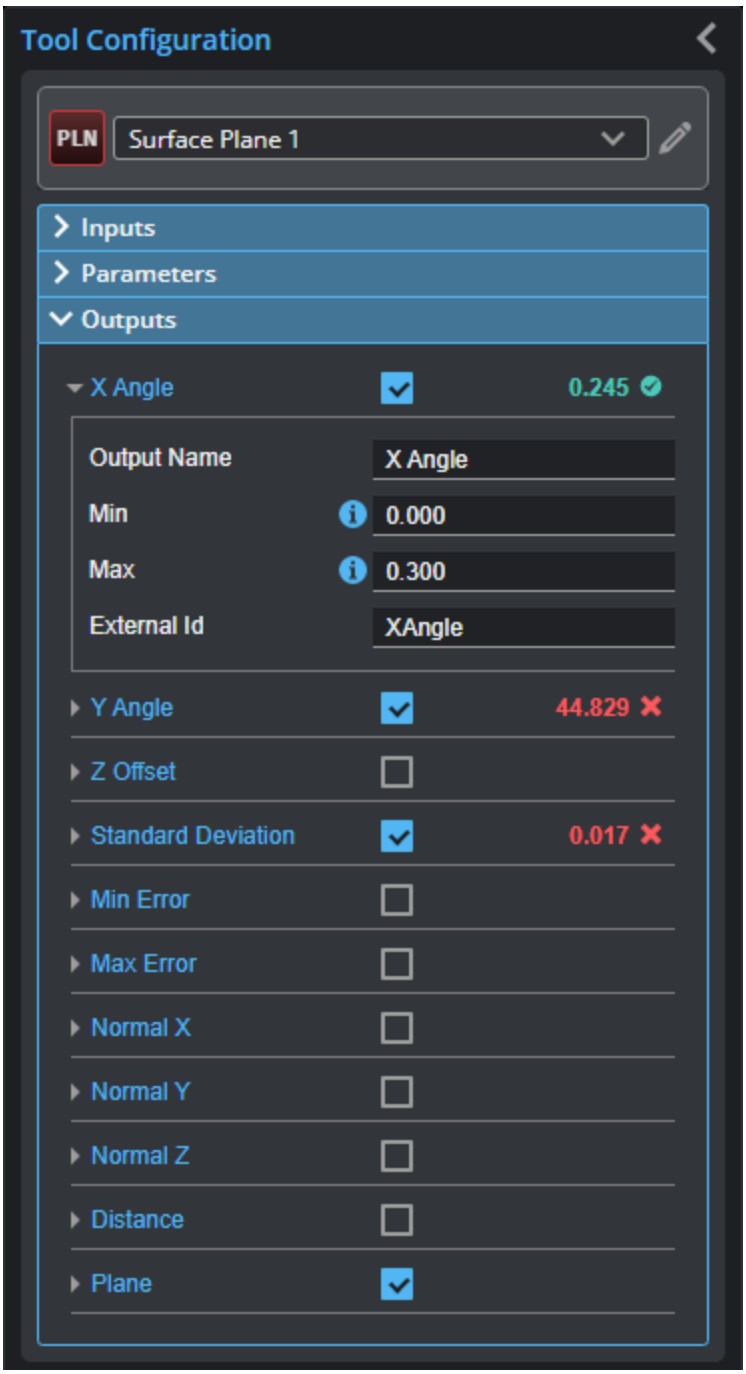
The following parameters are in the expandable **Parameters** section in the tool's configuration.



Parameters

Parameter	Description
Use Percentile Filter High Percentile Low Percentile	When **Use Percentile Filter** is checked, displays parameters to set the low and high percentile. Use this to remove outliers.
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

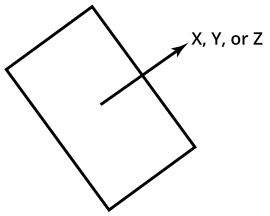
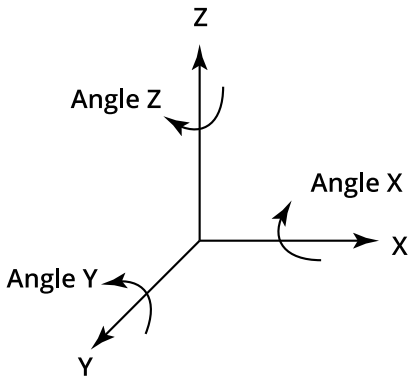


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	Illustration
X Angle Determines the X angle of the surface with respect to the alignment target.	
Y Angle Determines the Y angle of the surface with respect to the alignment target.	
Z Offset 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.	
Normal X Returns the X component of the surface normal vector.	
Normal Y Returns the Y component of the surface normal vector.	

Measurement	Illustration
Normal Z Returns the Z component of the surface normal vector.	
Distance Distance from the origin to the plane.	

Features

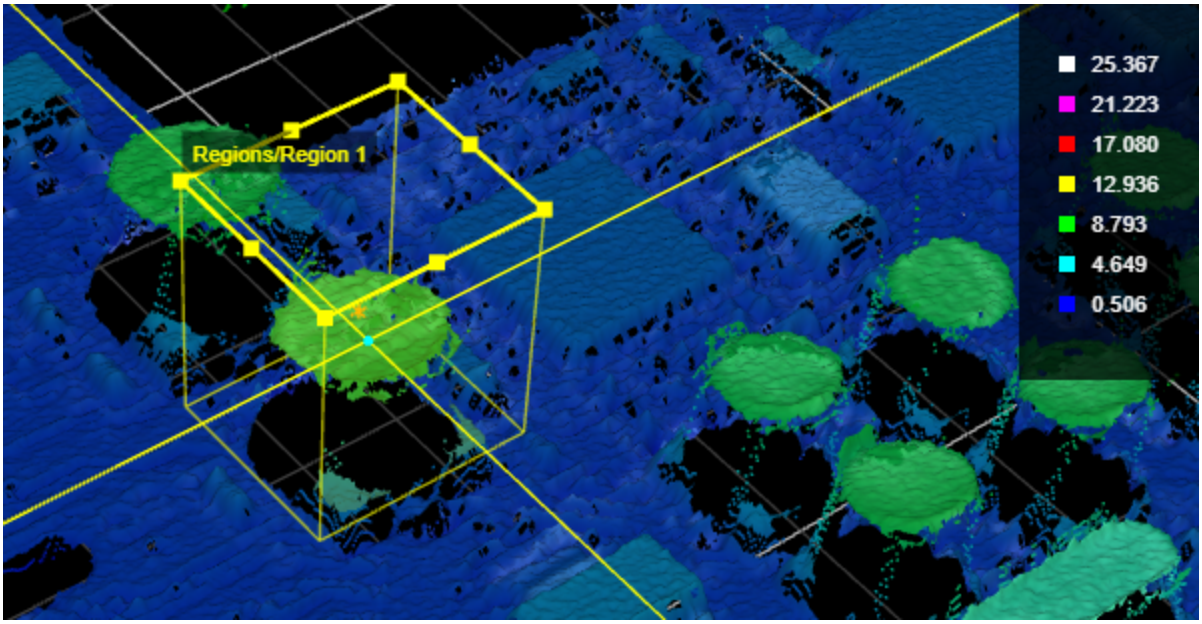
Type	Description
Min Error Point Max Error Point	Points representing the minimum and maximum errors.
Plane	The fitted plane.



For more information on geometric features, see *Geometric Features* on page 227.

Surface Position

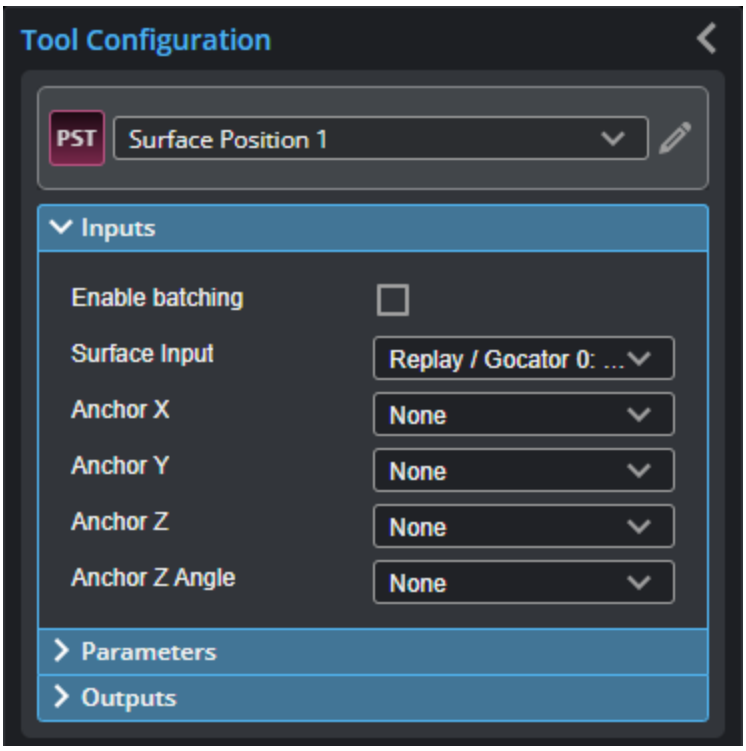
The Position tool reports the X, Y, or Z position of a part. You must specify the feature type specified and is one of the following: Average (the mean X, Y, and Z of the data points), Median (median X, Y, and Z of the data points), Centroid (the centroid of the data considered as a volume with respect to the z = 0 plane), Min X, Max X, Min Y, Max Y, Min Z, or Max Z.



Position X and Y on a PCB component

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

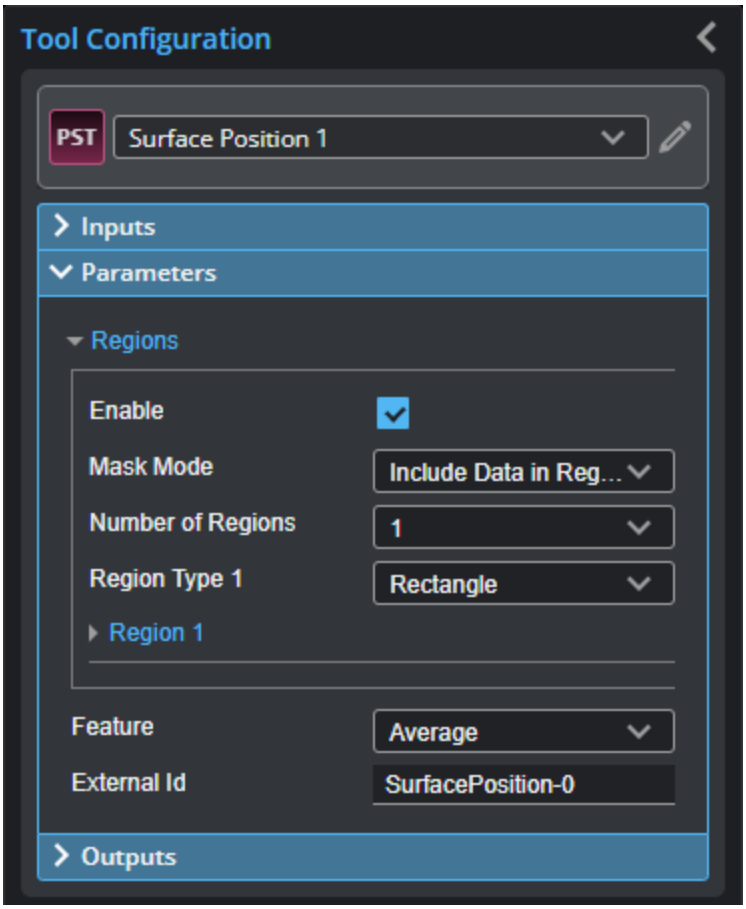
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool.

Name	Description
	Rotational anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

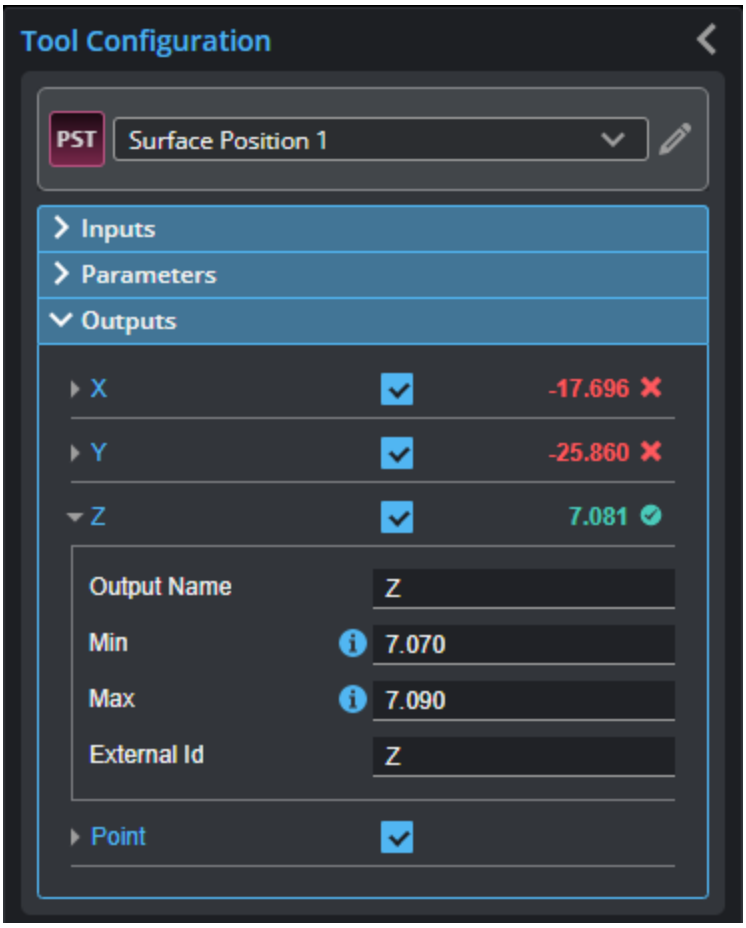


Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Feature	The feature point types the tool uses. For each, one of the following (for more information, see *Feature Points* on page 224):

Parameter	Description
	• Average • Median • Centroid • Max X • Min X • Max Y • Min Y • Max Z • Min Z When more than one point is at minimum Y or maximum Y (and the feature is set to Min Y or Max Y, respectively), the rightmost point (the one at greater positive X) is selected by the tool.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

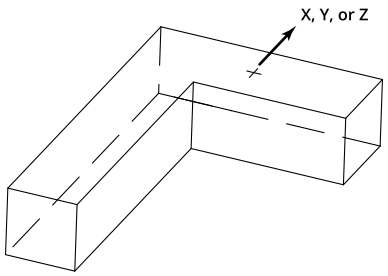


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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
Point	A Point geometric feature representing the returned position.

For more information on geometric features, see *Geometric Features* on page 227.

Surface Roughness

The Surface Roughness tool generates measurements of surface roughness.

> Because the accuracy of a roughness measurement is limited by the X resolution of a sensor, this tool is typically only used on confocal sensors, whose high resolution is better suited for roughness applications.

The following measurements are available:

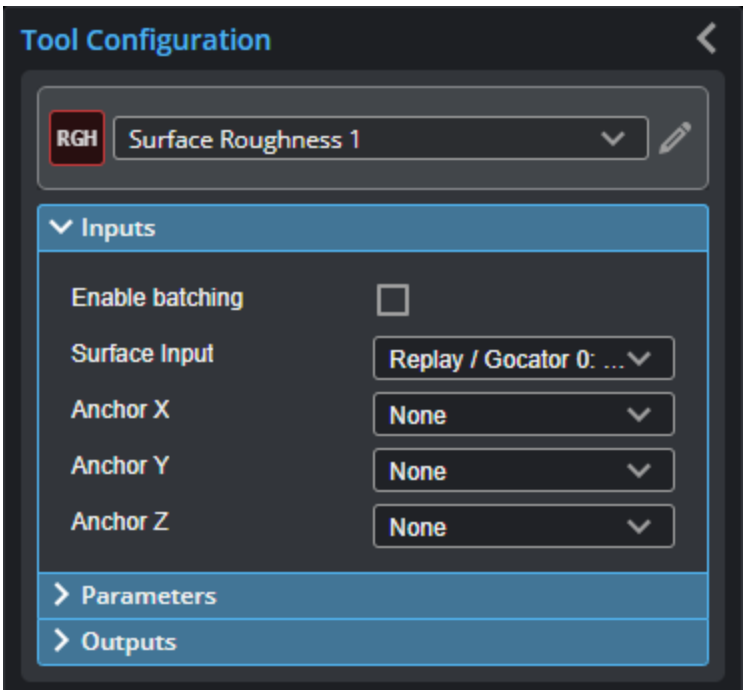
Sa	Sk
Sq	Spk
Sp	Svk
Sv	Smr1
Sz	Smr2
Sku	Sal
Ssk	Str

The tool offers five filter operations that can be enabled or disabled individually as required. The following filters are available:

- Filling Gaps: Designed to fill any holes in the surface and on the edge while trying to keep the surface as natural as possible.
- Remove Form: Offers both homogeneous and inhomogeneous polynomial filters from 1 to 8 orders.
- Remove Outliers: Used to remove outliers which are defined as a percentage of points in relation to the total number of points at the furthest distance from the height center.
- Gaussian and Cubic Spline filters: The tool offers Gaussian and Cubic Spline filters as standard filters for removing waviness. The Gaussian filter is used by default. The tool offers 4 conventional "end-effect" treatment options that are recommended by ISO 16610-28.
- The tool offers Gaussian filters with small Cut-Off wavelength (3 µm) as a low pass filter.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs



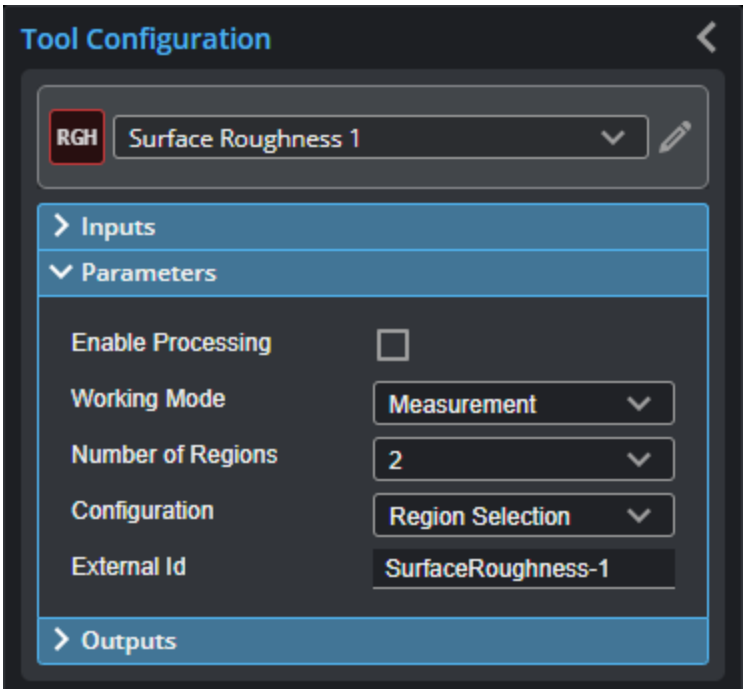
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

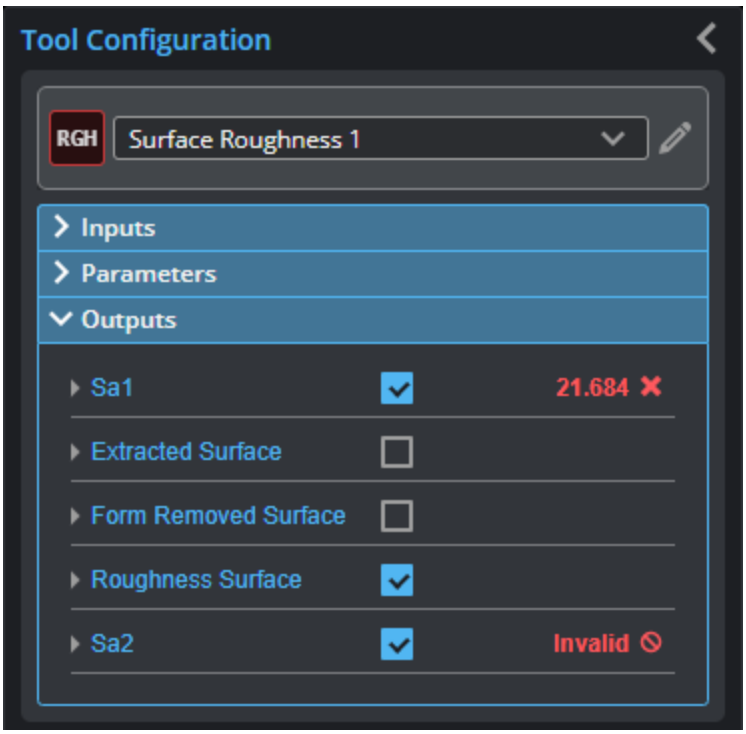


Parameters

Parameter	Description
Enable Processing	When enabled, the tool starts performing measurements.
Working Mode	The tool's working mode. One of the following: • Measurement • Check Parameters: Use this to check the calibration parameters. When you set the tool to this mode, an expandable **Calibration** section displays, containing additional parameters. For more information, see *Calibration* below. • Reset Parameters: If the calibration parameters are no longer used or a new calibration is carried out, select **Reset parameters** once. • Roughness Calibration: See *Roughness Calculation Procedure* on page 941. See *Calibration Procedure* on page 941.
Number of Regions	Sets the number of regions to be calculated, currently up to 16. (Two additional regions are reserved for calibration.)
Calibration	An expandable section displayed when **Working Mode** is set to **Check Parameters**. The tool gives the summed mean values and the recorded sample number after each measurement. For mirror calibration, the mean values of the background noise of Sa and Sq and the number of measurements are returned. For roughness calibration, the nominal values of Sa and Sq must be specified

Parameter	Description
	before the measurement starts. The mean scaling value and the number of measurements are returned.
Configuration	Lets you choose which region to configure in the expandable Region {n} section. Regions are independent. The two regions for mirror and roughness calibration targets are suspended after the specified number of evaluation regions, that is, after changing the number of evaluation regions, the settings for “Mirror” and “Roughness” must be checked again.
Region	An expandable section containing the parameters for the region selected in **Configuration**.
Use Region	If enabled, uses the region you define.
Number of Regions	Select the number of regions to be calculated, currently up to 18. Two regions are available for selection (the last 2 are reserved for calibration).
Fill Gaps Mode	Set to Linear Interpolation to fill any holes in the surface and on the edge while trying to keep the surface as natural as possible.
Remove Form	Check to remove form from the surface with homogeneous or inhomogeneous polynomial filter.
Polynomial Order	Order of polynomial function.
Remove Outliers	The outlier is defined here as a percentage of points in relation to the total number of points at the furthest distance from the height center. When using the integrated histogram curve, the outliers outside the threshold range are replaced by the corresponding limit values, which are calculated by the points within the threshold range.
Remove Waviness	Check to remove waviness using Gaussian or Cubic Spline filter.
Filter Type	Choose Gaussian filter or Spline Filter.
End Effect	Methods for dealing with end-effects. One of the following: • Zero Padding criterion (ZPA) • Linear Extrapolation (LEX) • Line Symmetric Reflection (LSR) • Point Symmetrical Reflection (PSR)
Use Low Pass Filter Low Pass Cut-off	Check **Use Low Pass Filter** to use low pass filter with cut-off wavelength defined in **Low Pass Cut-off**.
Margins	Custom margin distance.
Output Parameters	Some combination of measurement outputs can be selected. Dynamic measurement management is used.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

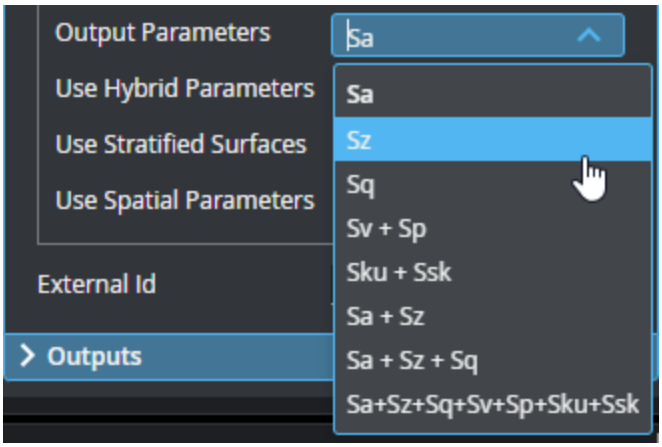


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

You select the following parameters (or a combination of the parameters) in the **Output Parameters** drop-down to list them in the tool's **Outputs** section.



Measurement	Formula
Sa {n} Arithmetical mean height in µm, which is the difference in height of each point compared to the arithmetical mean of the surface.	$Sa = \frac{1}{A}\iint_A |z(x,y)|dxdy.$
Sz {n} Maximum height in µm, which is defined as the sum of the largest peak height value and the largest pit depth value within the defined area.	$Sz = Sp + |Sv|$
Sq {n} Root mean square height in µm, which is equivalent to the standard deviation of heights.	$Sq = \sqrt{\frac{1}{A}\iint_A z^2(x,y)dxdy}$
Sv {n} Maximum pit Height in µm, which is the absolute value of the height of the largest pit within the defined area.	$Sv = \lvert \min_A z(x,y) \rvert$
Sp {n} Height of the highest peak within the defined area in µm.	$Sp = \max_A z(x,y)$
Sku {n} Kurtosis, which is a measure of the sharpness of the roughness profile. (No unit.)	$Sku = \frac{1}{Sq^4}\frac{1}{A}\iint_A z^4(x,y)dxdy.$
Ssk {n} Skewness, which represents the degree of bias of the roughness shape (asperity). (No unit.)	$Ssk = \frac{1}{Sq^3}\frac{1}{A}\iint_A z^3(x,y)dxdy.$
Sdq {n}, Sdr {n} Areal roughness hybrid parameters. To enable these parameters, check **Use Hybrid Parameters**. Sdq is the root mean square gradient. (No unit.) Sdr is the developed interfacial area ratio, as a percentage.	
Sk {n}, Spk {n}, Svk {n}, Smr1 {n}, Smr2 {n} Areal parameter for stratified surfaces.	

Measurement	Formula
Sk is the core roughness depth, in µm.	
Spk is the reduced summit height, in µm.	
Svk is the reduced valley depth, in µm.	
Smr1 is the upper bearing area, as a percentage.	
Smr2 is the lower bearing area, as a percentage.	

To add the following parameters in the tool's **Outputs** section, check **Use Hybrid Parameters**. The parameter outputs are enabled by default.

Measurements (Hybrid Parameters)

Measurement	Formula
Sdq {n}, Sdr {n}	
Areal roughness hybrid parameters. To enable these parameters, check **Use Hybrid Parameters**.	
Sdq is the root mean square gradient. (No unit.)	
Sdr is the developed interfacial area ratio, as a percentage.	

To add the following parameters in the tool's **Outputs** section, check **Use Stratified Surfaces**. The parameter outputs are enabled by default.

Measurements (Areal Parameters)

Measurement	Formula
Sk {n}, Spk {n}, Svk {n}, Smr1 {n}, Smr2 {n}	
Areal parameters for stratified surfaces.	
Sk is the core roughness depth, in µm.	
Spk is the reduced summit height, in µm.	
Svk is the reduced valley depth, in µm.	
Smr1 is the upper bearing area, as a percentage.	
Smr2 is the lower bearing area, as a percentage.	

To add the following parameters in the tool's **Outputs** section, check **Use Spatial Parameters**. The parameter outputs are enabled by default.

Measurements (Areal Parameters)

Measurement	Formula
Sal {n}, Str {n}	
Spatial parameters.	
Sal is the auto-correlation length, in µm.	
Str is the texture aspect ratio. (No unit.)	

Data

Type	Description
Extracted Surface	Surface data after filling gaps.
Form Removed Surface	Surface data after subtracting the polynomial pattern.
Roughness Surface	Surface data after a high pass filtering.

Calibration Procedure

1. Deactivate "Enabled Processing" to get into the editing mode.
2. Use "Check Parameters" in the "Working Mode" option to check the last calibration.
3. Use Reset Parameters to ensure that all parameters from the last calibration are reset. If no correction is used, this action can also be used.
4. Change the "Working Mode" to "Mirror calibration".
5. Editing position and size of the region and other settings.
6. Use the input data measured with a mirror.
7. Activate "Enabled Processing" to start the mirror calibration.
8. Repeat the measurement as necessary. The tool gives the summed up mean value of background noise and the recorded sample number after each measurement. 25 measurements is recommended, but there is no upper limit to the number of measurements.
9. Deactivate "Enabled Processing" to get into the editing mode.
10. Change the "Working Mode" to "Roughness calibration".
11. Enter the nominal values of Sa and Sq of the calibration target. Depending on requirements, Sa, Sq can be calculated separately or together, (since the two correlate very well, a mean value of the correction factor can be formed if the two are activated).
12. Editing position and size of the region and other settings.
13. Use the input data measured with a roughness calibration target.
14. Activate "Enabled Processing" to start the roughness calibration.
15. Repeat the measurement as necessary. The tool gives the summed up mean value of correction factor and the recorded sample number after each measurement. 25 measurements is recommended, but there is no upper limit to the number of measurements.
16. It is assumed that a job file is created after the calibration, so the information is preserved in the job file.

Roughness Calculation Procedure

1. Deactivate "Enabled Processing" to get into the editing mode.
2. Select the number of regions to be calculated, currently up to 20 - 2 regions are available for selection (the last 2 are reserved for calibration).

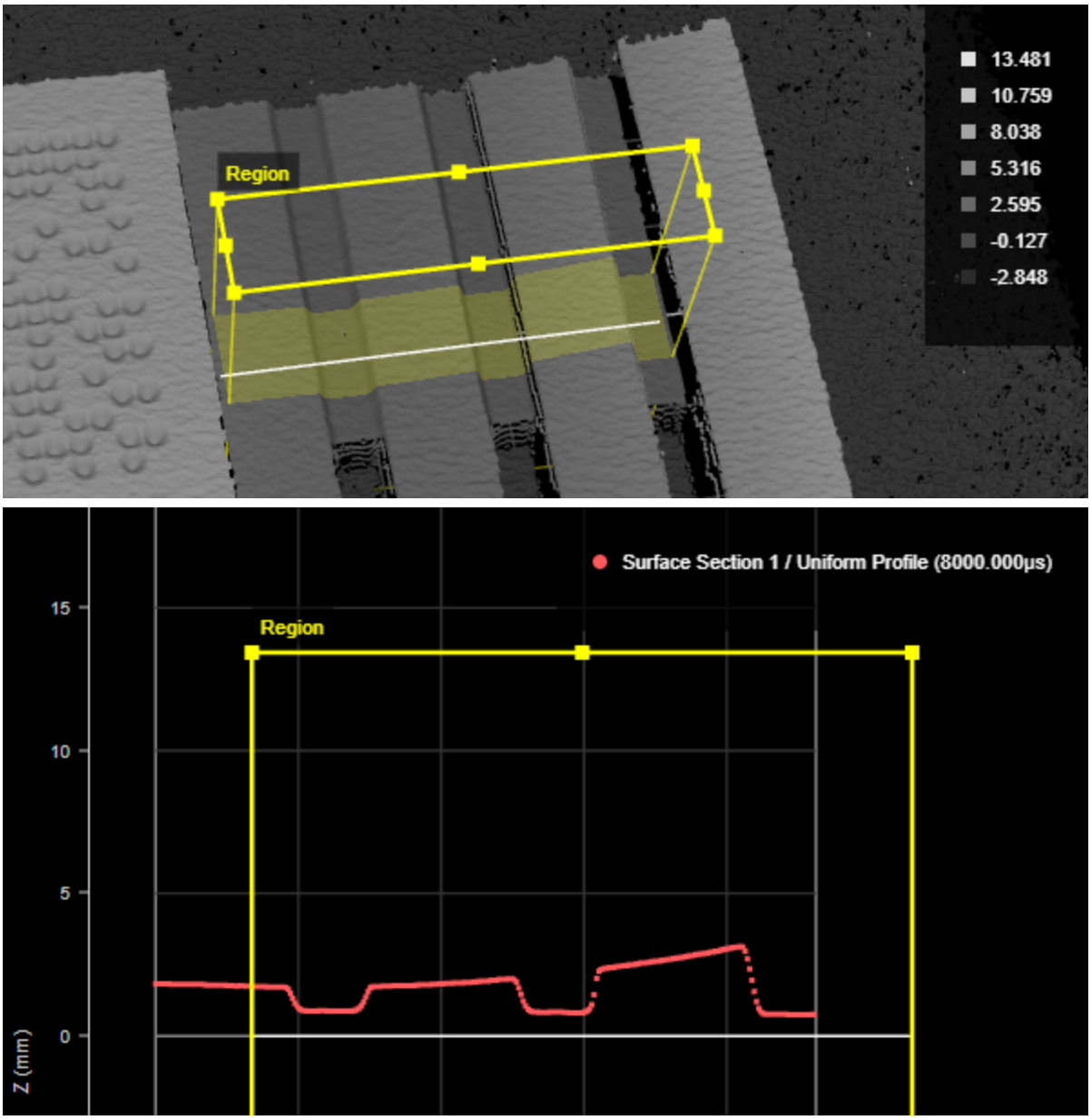
3. Configure individual regions. The default settings correspond roughly to the settings when using Mountains Map, so only the region should be set. Otherwise, all available configurations for each region should be set according to specification.
4. Activate the measurements and output surfaces.
5. Activate "Enabled Processing" to start the calculation.

Surface Section

The Surface Section tool lets you define a line on a surface (a "section") from which the tool extracts a profile. You can then apply any Profile tool to the resulting profile (see *Profile Tools* on page 515). A section can have any XY orientation on the surface. The resulting profile is on the XZ plane.

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



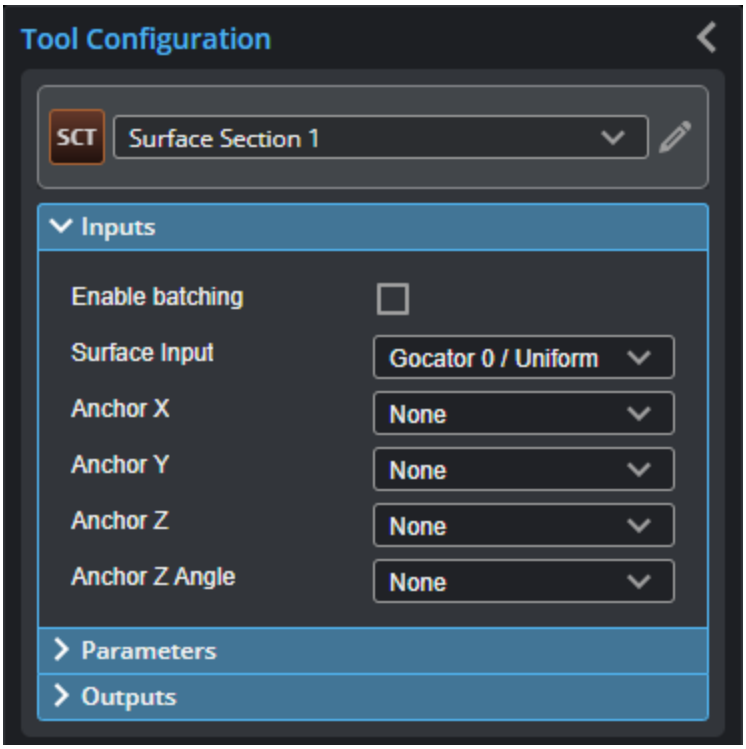
The resulting profile. The leftmost point of the profile is at the X origin. The region is positioned showing the X and Z position of the center of the region on the surface.

You can anchor the tool to some other easily identifiable feature on the scan target, which "shifts" the section in relation to that feature in case the target moves. This increases repeatability. For more information, see *Measurement Anchoring* on page 209.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

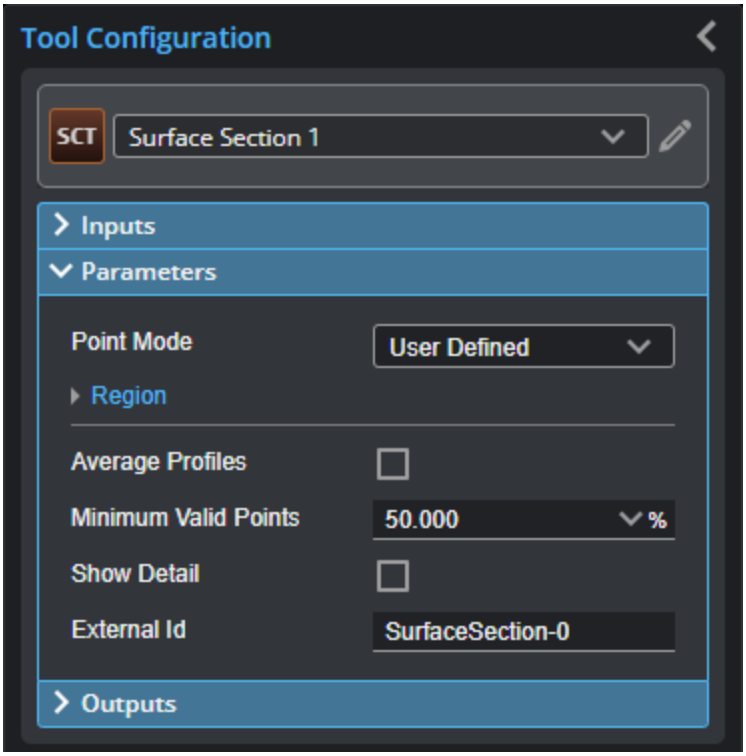
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Name	Description
Point 1 Point 2	The Point geometric features the tool uses to define the end points of the section. Only displayed if **Point Mode** is set to Feature Points.

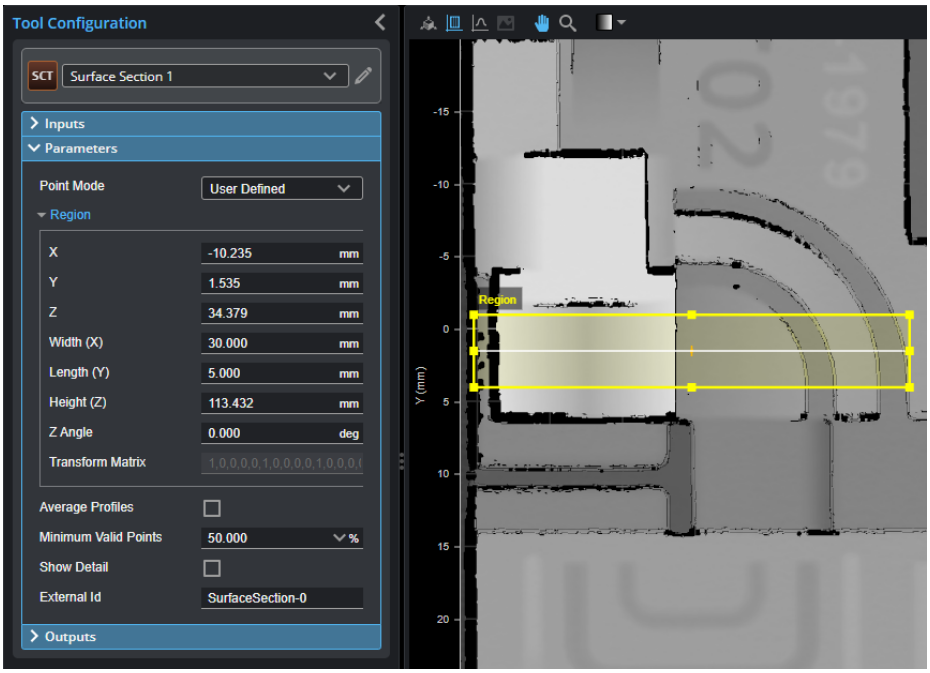
Parameters

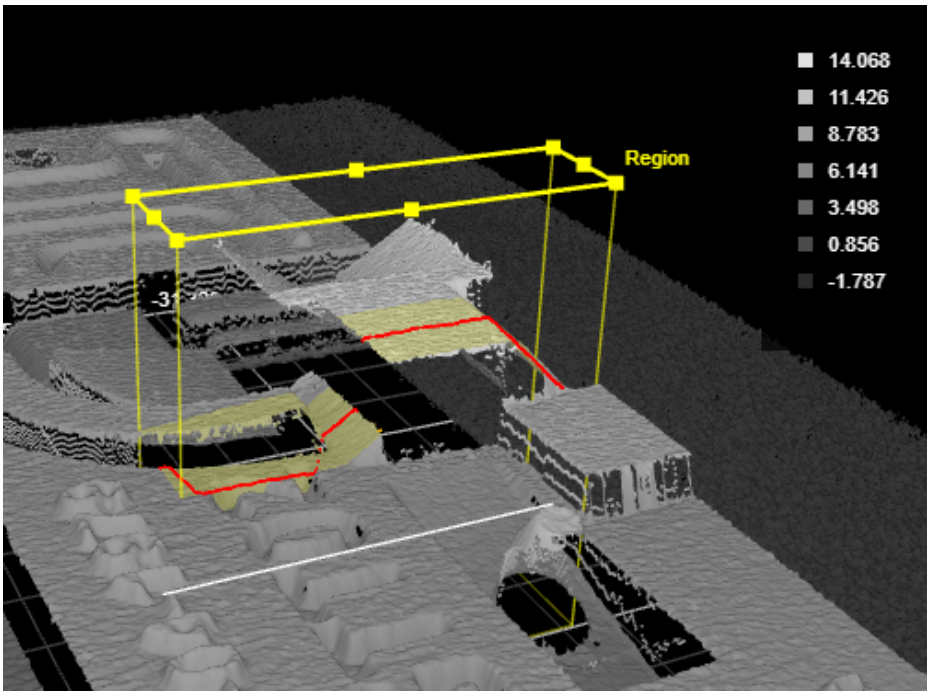
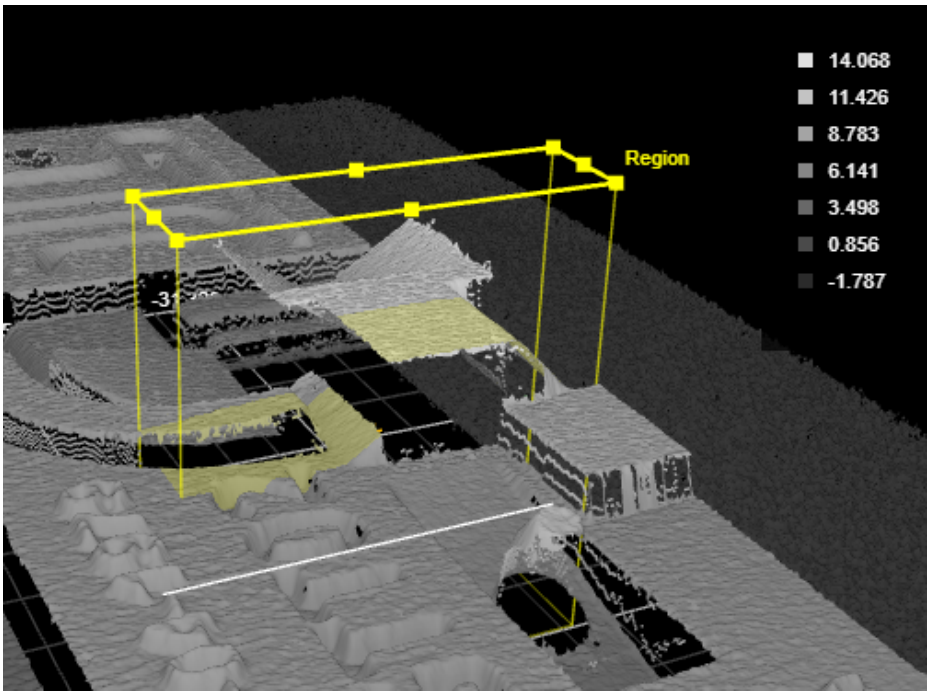
You configure the tool's parameters in the expandable **Parameters** section.



Parameters

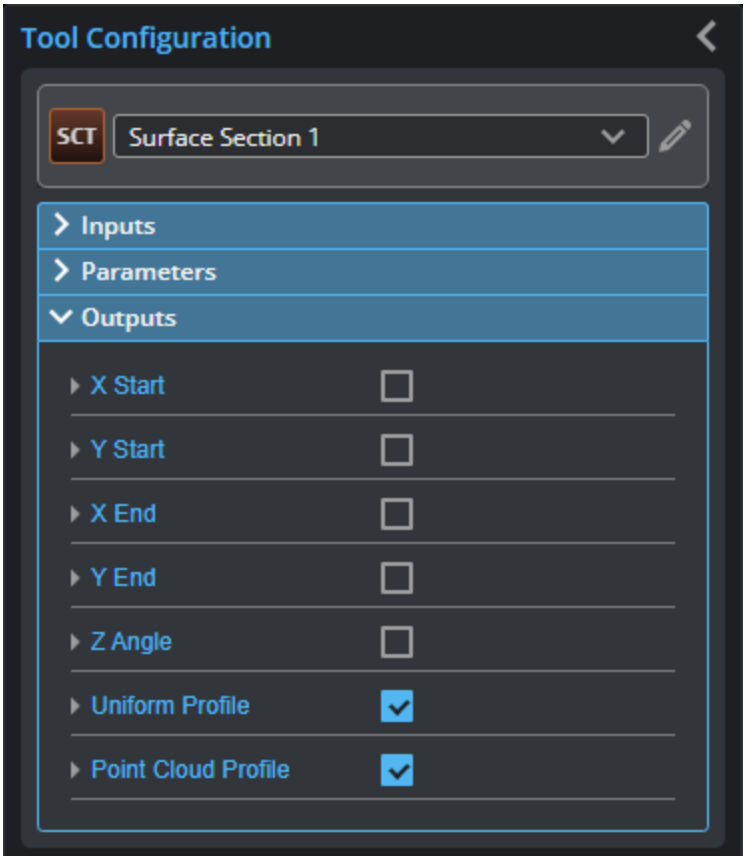
Parameter	Description
Point Mode	Selects how the tool defines the end points of the section. One of the following: **User-Defined**: Enables a region (see **Region** below) whose size, position, and rotation you define. The section is placed along the X axis in the middle of the region. **Feature Points**: Uses two Point geometric feature inputs as the end points of the section.
Region	A rectangular region to define the section to extract. By default, the section is taken along the X axis of the region. The length of the region (along the Y axis) is the averaging width for the section. In the following, the region is set to a length of 5 mm and a width of 30 mm. The tool will therefore use a 5 mm averaging window, perpendicular to the X axis (width) of the region, to calculate the data point using its neighbor. If **Average**

Parameter	Description
	Profiles is enabled, the tool averages the points along the entire length of the region.  For more information, see *Regions* on page 214.
Average Profiles	When enabled, averages the profiles in the region to produce the section profile. Otherwise, the tool only uses the data points under the line running parallel to the X axis in the middle of the region.
Minimum Valid Points	When **Average Profiles** is enabled, 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.
Show Detail	Determines whether data points (in red) are displayed under the section in the data viewer. If this setting is disabled, only the line representing the defined section is displayed. **Show Details** enabled:

Parameter	Description
	 **Show Details** disabled: 
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.

Measurement

Z Angle

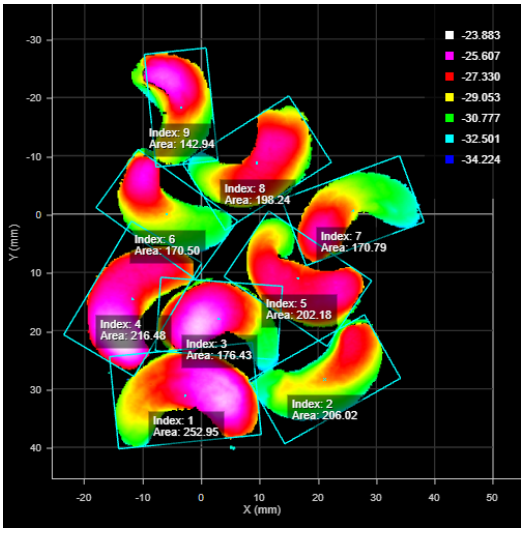
Returns the rotation of the section around the Z axis.

Data

Type	Description
Uniform Profile Point Cloud Profile	The profile (uniform or point cloud, respectively) that the tool extracts from the surface. Available to profile tools for profile measurement.

Surface Segmentation

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 the center of its bounding box, its length and width, and its area, as well as several more global measurements, such as maximum / minimum width or length, and so on. 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 *Profile Part Detection* on page 603.

	The Segmentation tool cannot handle large overlaps.

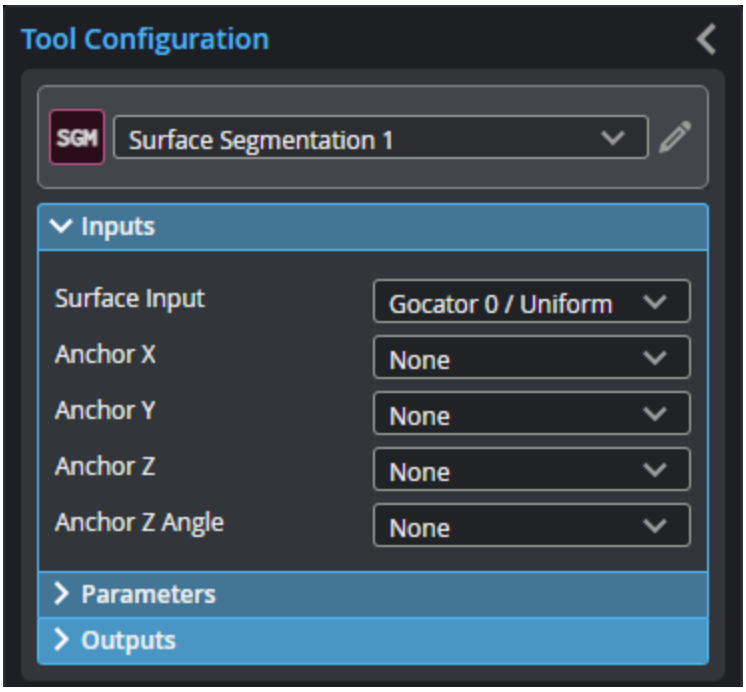
	The Segmentation tool does not perform template matching.

	To reduce processing time, consider using the decimation filter. For more information on this filter, see *Surface Filter* on page 862.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



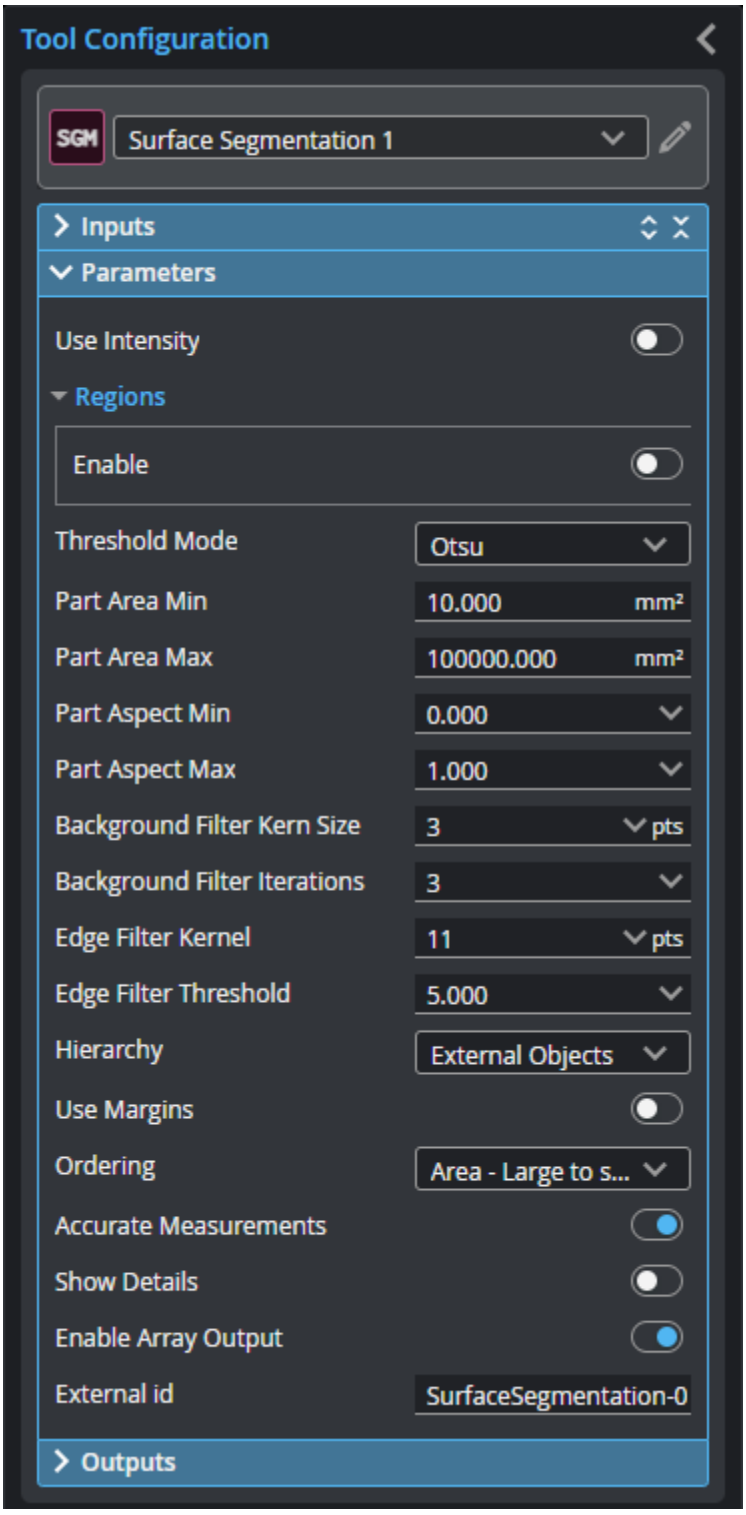
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Surface Input	The data the tool applies measurements to or processes.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

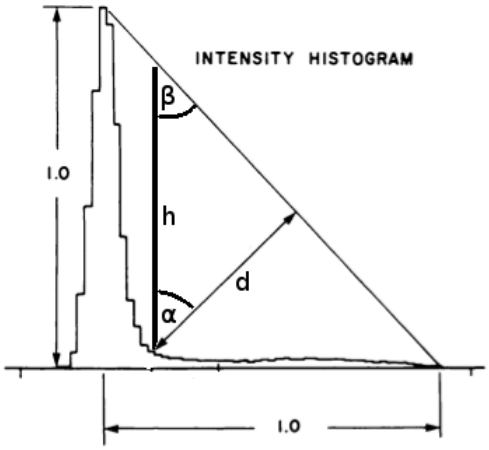
Parameters

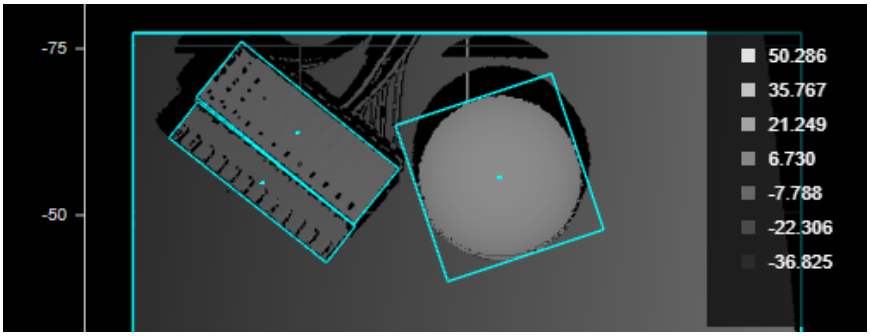
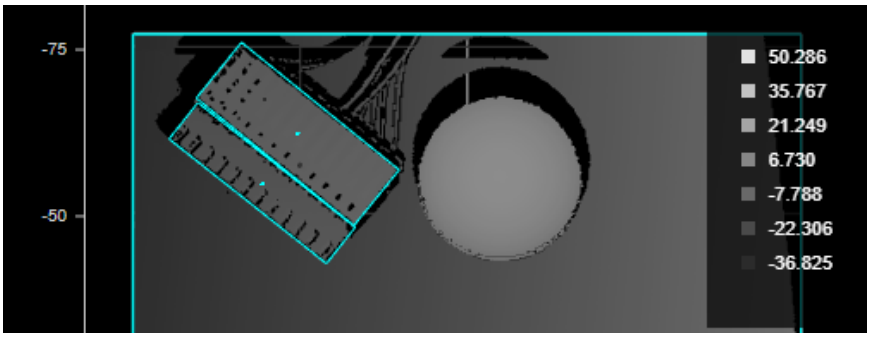
You configure the tool's parameters in the expandable **Parameters** section.

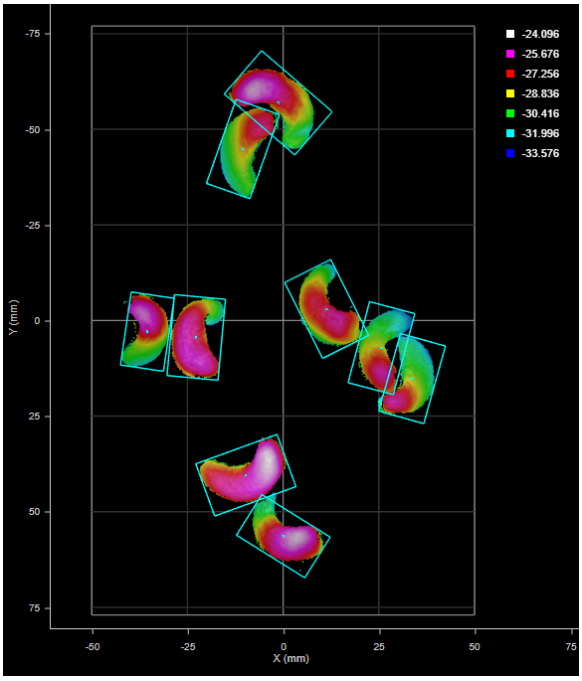


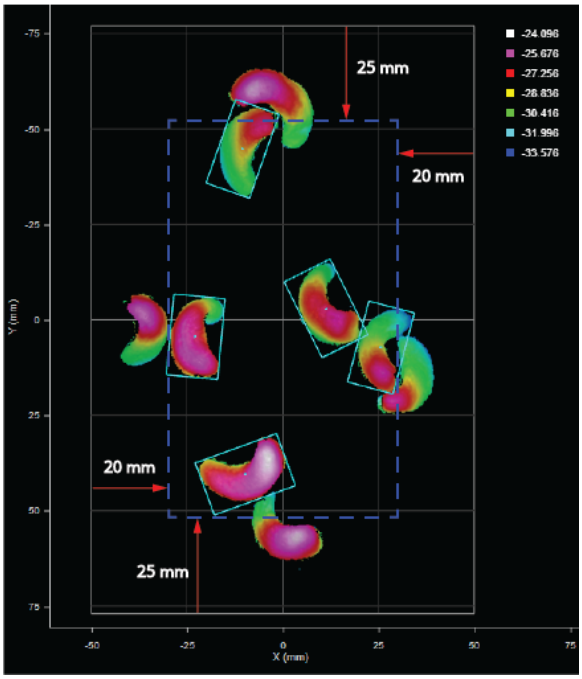
Parameters

Parameter	Description
Use Intensity	Causes the tool to use intensity. **Acquire intensity** must be enabled on the **Scan**

Parameter	Description
	page; otherwise, the tool will display an error.
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Threshold Mode	Sets the threshold mode. You should experiment with the different options to find the one that works the best with your scan data. **Otsu** Uses "Otsu's method" to determine the threshold that maximizes the between-class variance. **Triangle** Determines the threshold based on the maximum distance to the histogram from the triangle generated from the peak histogram bin to the farthest smallest bin.  **Fixed** Lets you provide a fixed threshold in the **Threshold** setting.
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.

Parameter	Description
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.
Edge Filter Kernel	Use this value to adjust the “granularity” of the part edge detection.
Edge Filter Threshold	Controls the separation of the parts, increasing the gap between the parts so that they can be detected more easily.
Hierarchy	Use this setting to detect segments when they are surrounded by background data. Choose one of the following: **All Parts** or **External Objects**. **All Objects** This option lets you segment parts with surrounding background data. 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 Objects** In the following image, 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 scan data.  **Holes**

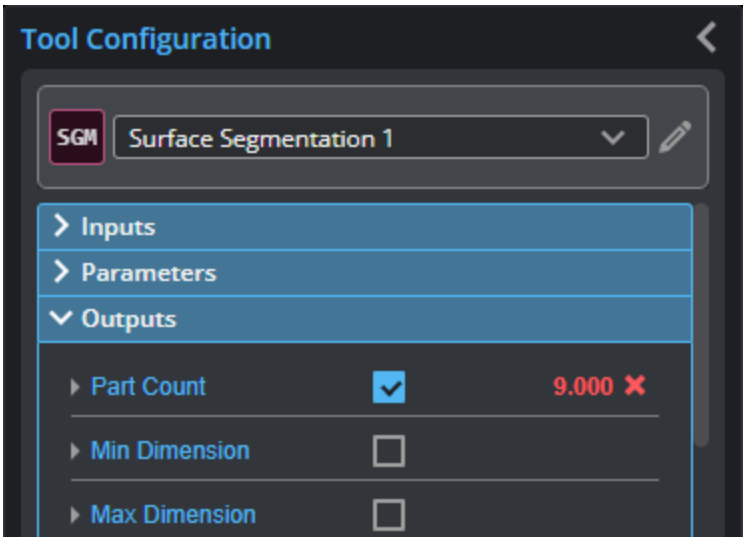
Parameter	Description
	Use this option to get holes inside objects. **All Contours** Use this option to get all detected contours.
Hierarchy	Provides options to let you find either external objects only or all objects. **External Objects** Use this option to ignore smaller objects in larger objects. Only the outermost objects are returned. **All objects** Use this option to include smaller objects in larger objects. **Holes** Use this option to get holes inside objects. **All Contours** Use this option to get all detected contours.
Use Margins Margins Left Right Top Bottom	When **Use Margins** is enabled, the tool discards parts whose center point is within the margins defined in the expandable Margins section. Margins extend inwards from the outer edges of the sensor's FOV (XY scan area) and are positive. In the following, with margins set to 0, nine parts are detected  In the following, the left and right margins are set to 20 mm, and the top and bottom margins are set to 25 mm. The sensor's FOV is 100 x 154 mm. The tool discards the four parts whose center points are within those margins. (Total part

Parameter	Description
	count is reduced to five.) 
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
Accurate Measurements	Returns more accurate width, height, and area measurements, as well as a better Surface output, but the trade-off is greater processing time. If you only need center points or the number of parts, disable this parameter for faster processing..
Show Details	Toggles whether the tool displays the index and area of each individual part. Enabling this will increase processing time.
Enable Array Output	When **Enable Array Output** is disabled, the tool outputs each part as an individual Surface output. When **Enable Array Output** is enabled, the tool outputs parts in arrays. Be sure to enable the Part Surface in the **Outputs** section after enabling this parameter. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.

Parameter	Description
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. If **Enable Array Output** is checked, this parameter is hidden and all parts are output in arrays.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

> When **Enable Array Output** is unchecked, individual measurement and feature outputs are produced for each part, for example, Max Z 1, Max Z 2, and so on, or Center 1, Center 2, and so on. When this parameter is checked, these measurements and features are output as arrays.

Measurements

Measurement
Part Count Returns the total number of segments identified, based on the tool's parameters.

Measurement
Min Dimension **Max Dimension** The minimum and maximum dimensions among all of the identified segments, respectively.
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, respectively.
Sum Area The sum of the areas of the segments.
Mean Area The mean area of the segments.
Min Height **Max Height** **Mean Height** The minimum, maximum, and mean heights among all of the identified segments, respecitvely.
X Center {n} **Y Center {n}** The X and Y positions of the center of a part segmented from the surface.
Width {n} **Length {n}** The width and length of a part segmented from the surface. These are always the major and minor axis of a part, respectively.
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 1014.

Features

Type	Description
Center {n}	The point representing the center of a segmented part.

Data

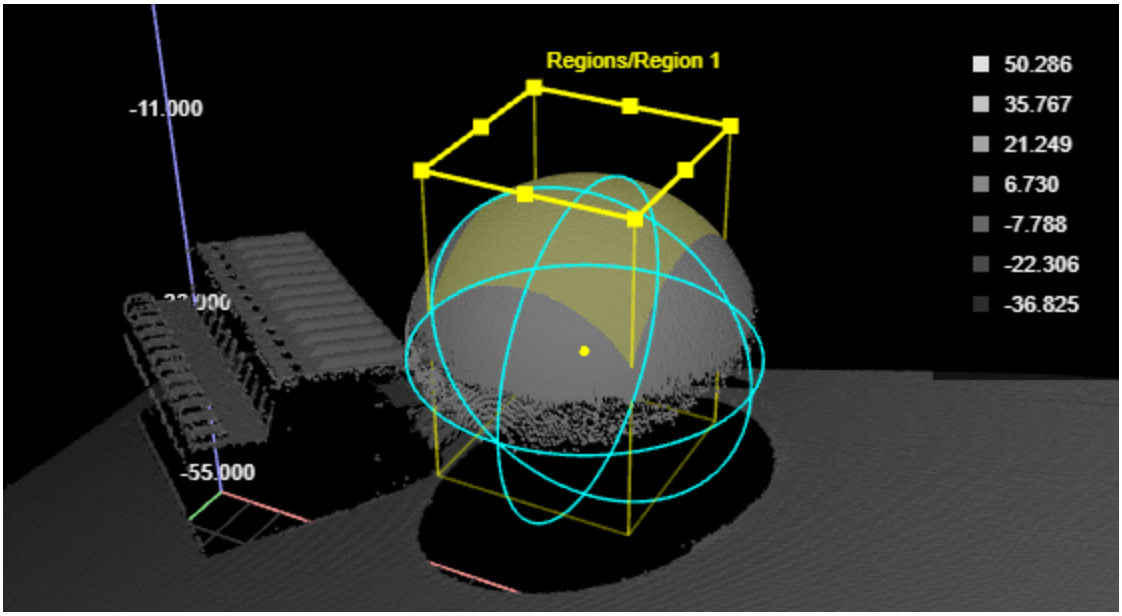
Type	Description
Part Surface {n}	Surface data corresponding to each segmented part.

Type	Description
Diagnostic	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 Sphere

The Sphere tool lets you compute characteristics of a scanned sphere by specifying a region to inspect.

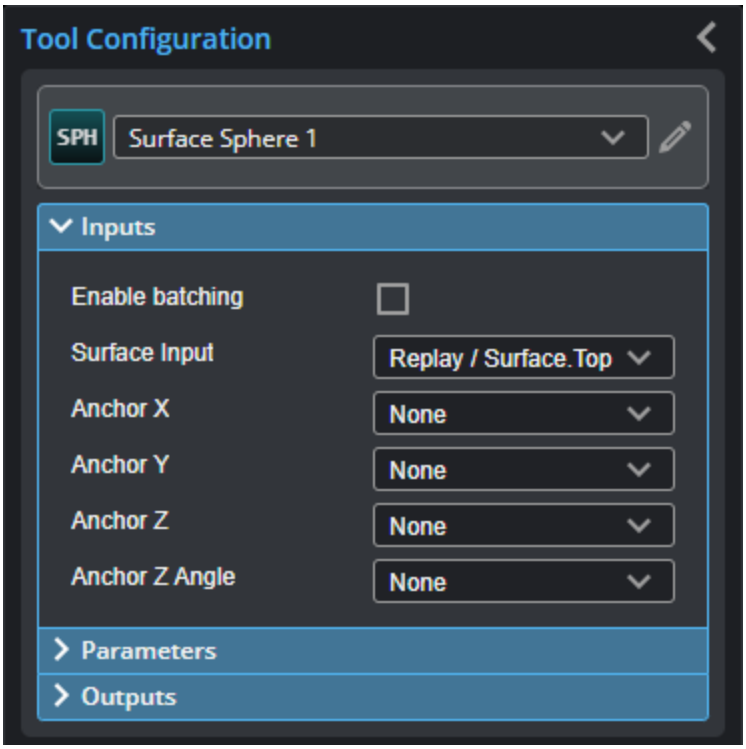
> For the tool to work properly, the tool's region must usually be enabled and properly placed to include only the sphere's scan data.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



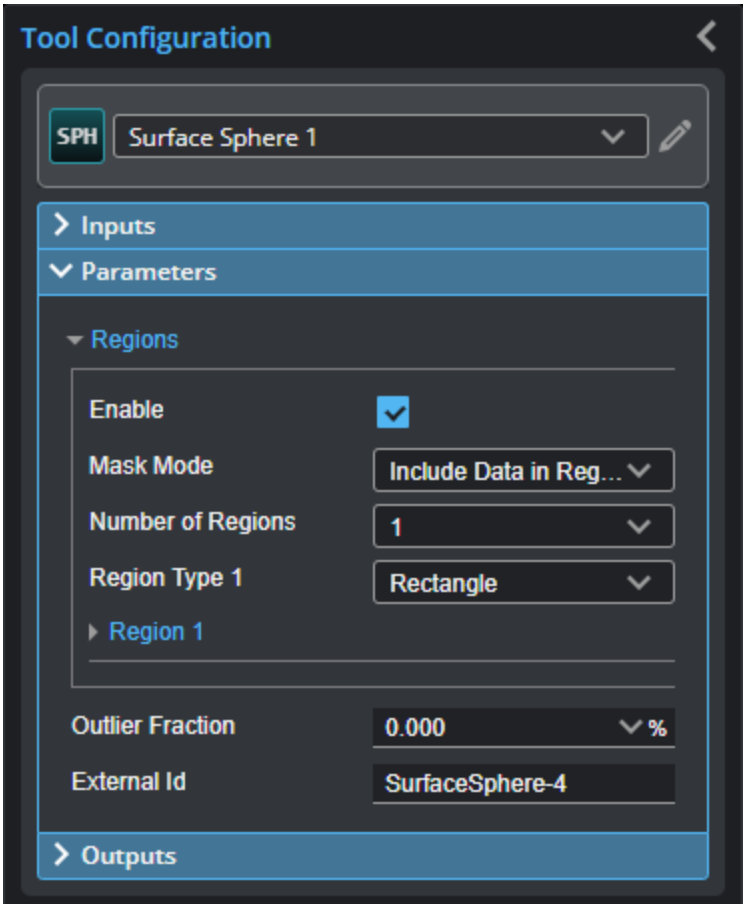
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes. This tool can optionally take an array as input. For more information, see *Arrays, Batching, and Aggregation* on page 199.
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

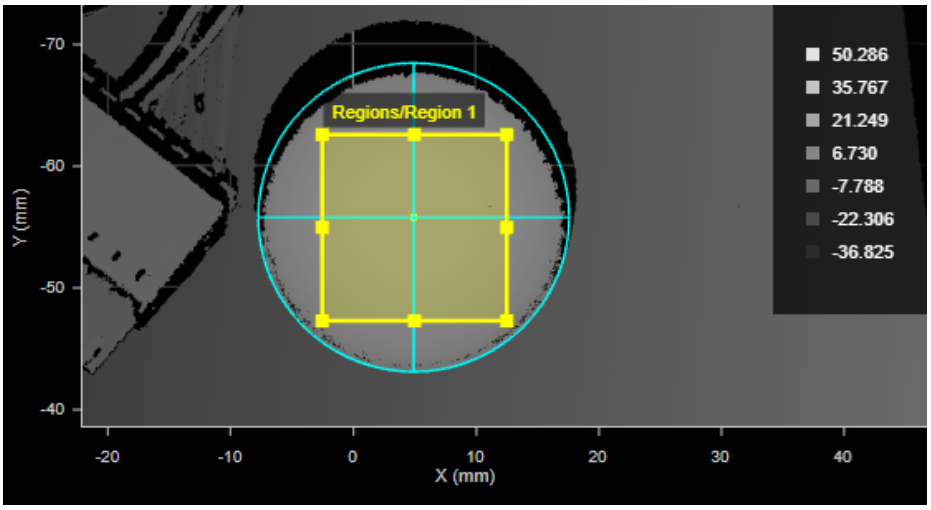
Parameters

You configure the tool's parameters in the expandable **Parameters** section.



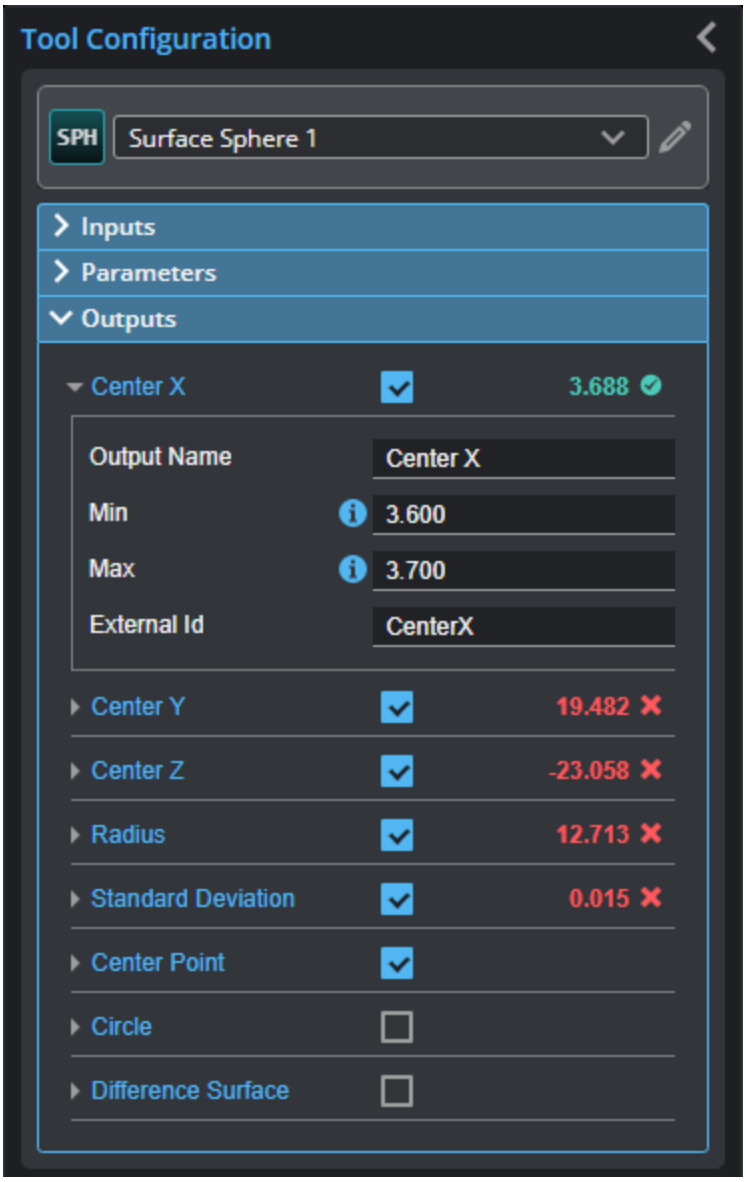
Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Number of Regions Region Type {n} Region {n}	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214. 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.

Parameter	Description
	
Outlier Fraction	The percentage of outlier points to exclude. Setting this to a small value can help the tool fit a better sphere when noise is present.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

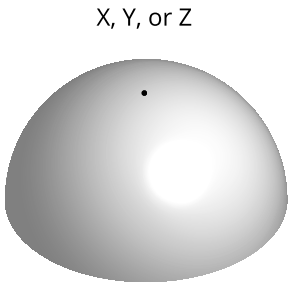
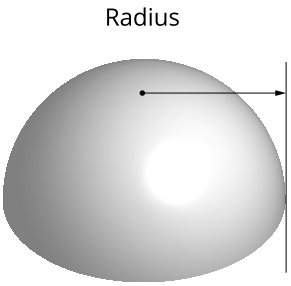


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	Illustration
Center X, **Center Y**, **Center Z** These measurements determine the X, Y, Z position of the center of the sphere, respectively.	
Radius Determines the radius of the sphere.	
Standard Deviation 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.	

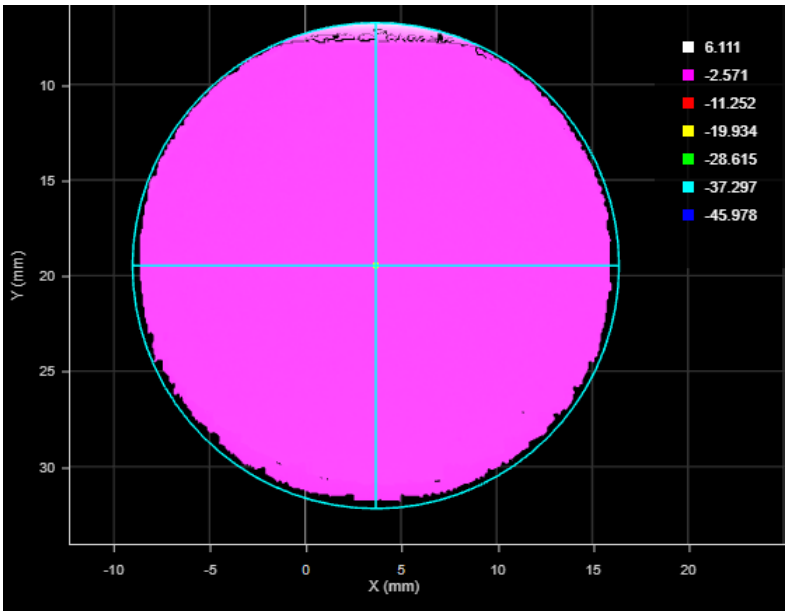
Features

Type	Description
Center Point	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 227.

Data

Type	Description
Difference Surface	Shows the fit error at each point in the height map.

Type	Description
	

Surface Stitch

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 **Surface Input** drop-down.

The tool performs rotation first, and then translation.

> The tool combines data simply by overwriting in sequence: it performs no averaging or blending. The tool also performs no fitting.

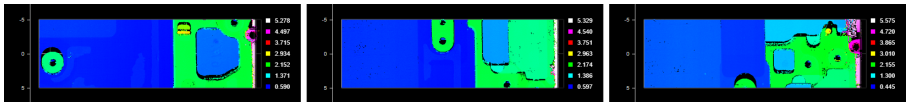
> Results are only as accurate as the motion system.

> Seams are often seen in combined data in stitching performed in anything other than along the Y axis.

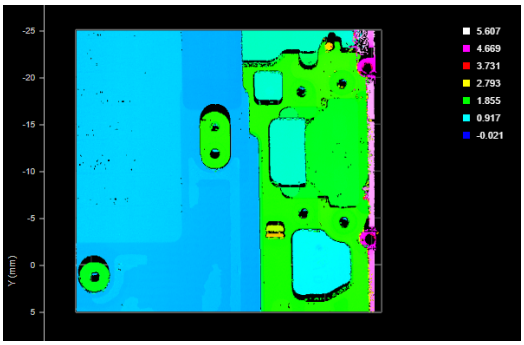
The tool returns one measurement, which simply indicates the number of scans successfully added to the combined scan data.

The following shows three individual frames.

(The following screenshots are from a G2 sensor.)

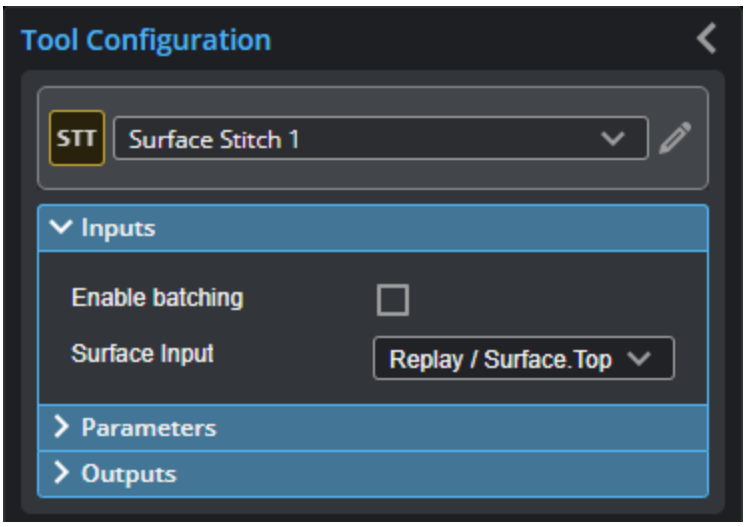


In the following, the tool has combined the frames into a single surface.



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

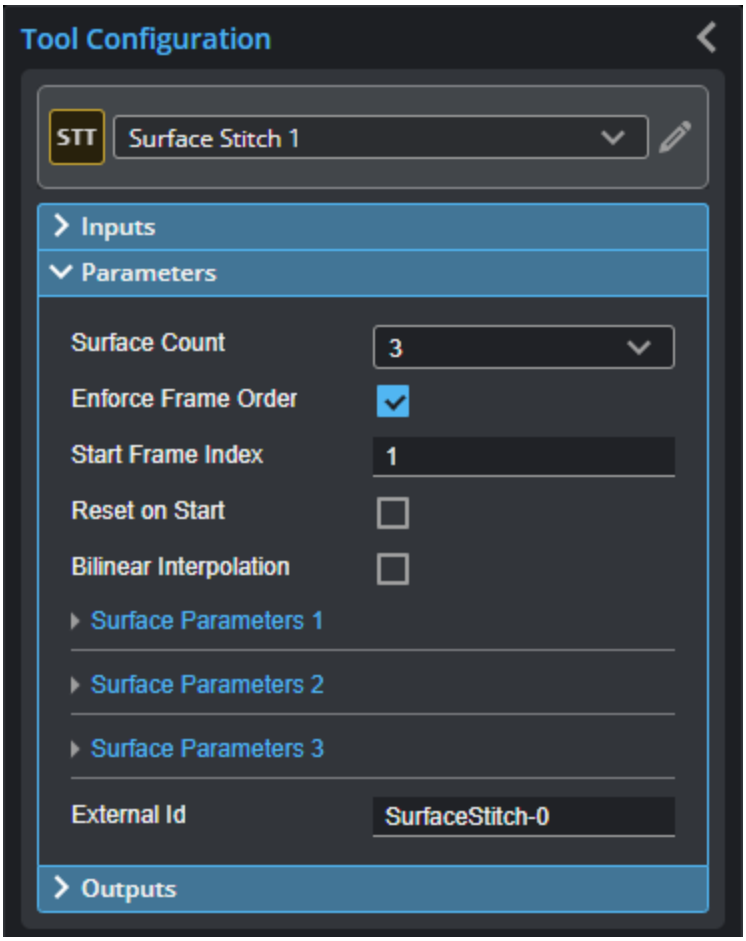


Inputs

Name	Description
Surface Input	The data the tool applies measurements to or processes.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

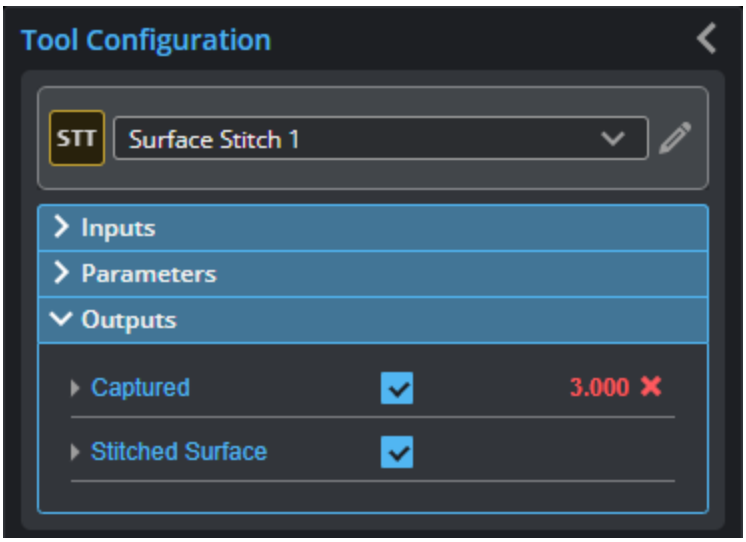


Parameters

Parameter	Description
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. 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).
Start Frame Index	The start frame index.
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

Parameter	Description
	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.
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.
Interpolation Mode	Provides three interpolation modes the tool uses to stitch the input Surface data scans together to produce the uniform output (Stitched Surface). These modes do nothing when the input is point cloud data. **Nearest Neighbor Interpolation** Data points are generated in the stitched output by interpolating with the nearest neighbor. In earlier versions of this tool, this is equivalent to **Bilinear Interpolation** being unchecked. **Bilinear** Data points are generated in the stitched output by using bilinear interpolation with the points to the left and right. This mode is best for applications with a non-zero Z rotation and minor X and Y rotation. In earlier versions of this tool, this is equivalent to **Bilinear Interpolation** being checked. **Gaussian** When stitching uniform Surface data where there is a larger X or Y angle, use this option to improve stitching.
Surface Parameters {n}	For each scan to be added to the combined surface scan, an expandable **Surface Parameters** section is added. The following settings are available: • X, Y, and Z Offset • X, Y, and Z Angle • Mirror: Mirrors input Surface {n}.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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 and Intensity* on page 139.

Data

Type	Description
Stitched Surface	The stitched surface scan.
Stitched Raw Surface	

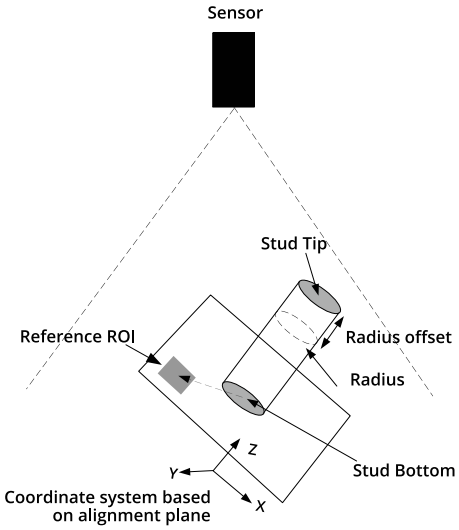
Surface Stud

The Stud tool measures the location and radius of a stud.

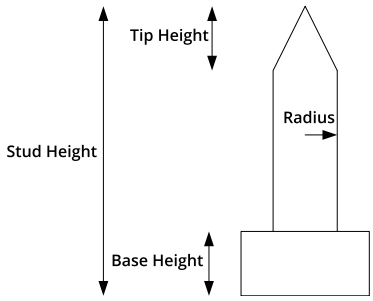
> The tool does not search for or detect the feature. The tool expects that the feature, conforming reasonably well to the defined parameters, is present and that it is on a sufficiently uniform background.

The tool uses a complex feature-locating algorithm to find a hold and then return measurements. For a detailed explanation of the algorithm, see *Stud Algorithm* on the next page.

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.



Measurement Panel

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Stud Algorithm

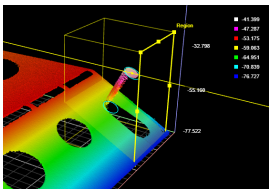
The Stud algorithm measures the stud in three steps: searching for the tip, finding the reference plane, and shaft fitting. Note that the tip and the side of the stud must be within the measurement region.

See the tool's parameters for an explanation of the options that affect the tool's algorithm.

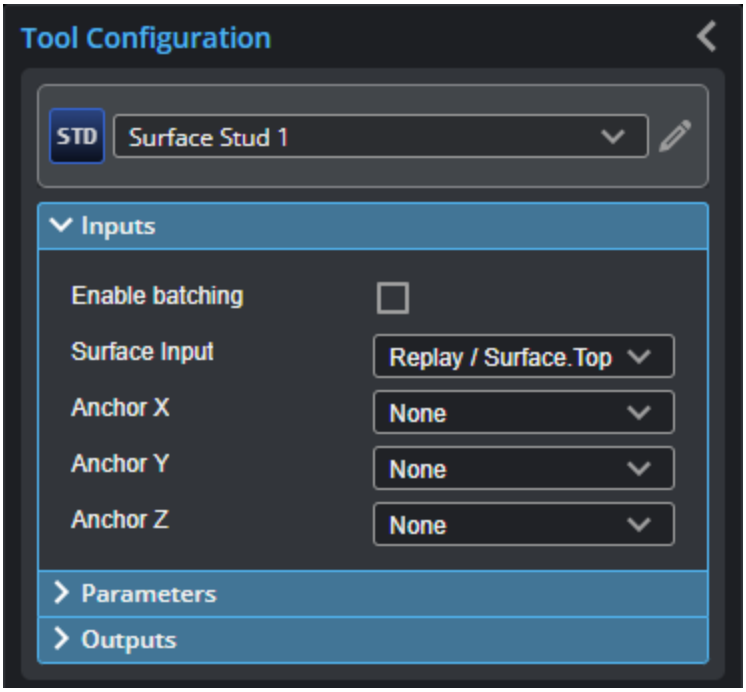
Searching for the tip - The algorithm looks for the approximate location of the tip. If Auto-Tilt is enabled, the algorithm uses the flat surface around the tip to estimate the orientations of the part. The approximate tip is the location of the highest (maximum Z) pixel after correction for the nominal tilt angle.

Finding the reference plane - The reference regions are positioned using the approximate tip, the nominal angle values, and the nominal stud length. Compared to the hole/opening, misplaced stud reference regions are more likely to cause a failure to produce any measurement.

Shaft fitting - The shaft region is determined based on the approximate tip position, the nominal angles, the reference plane position, and the stud nominal size parameters. Shaft fitting is successful if the algorithm can fit at least three circles with the stud diameter along the shaft. Fitting each circle requires sufficient data along the top portion the shaft. Because of occlusions, the bottom of the shaft is often not visible to the sensor and the algorithm is designed to handle this situation.



Inputs



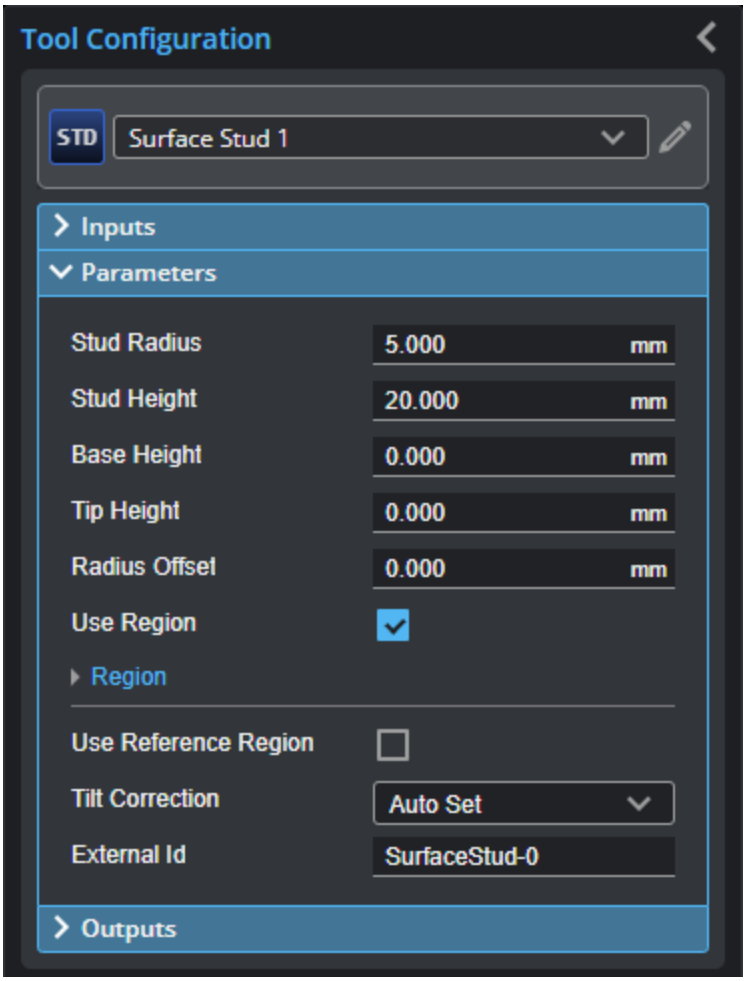
To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Anchor X	Positional anchors are optional.
Anchor Y	
Anchor Z	

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

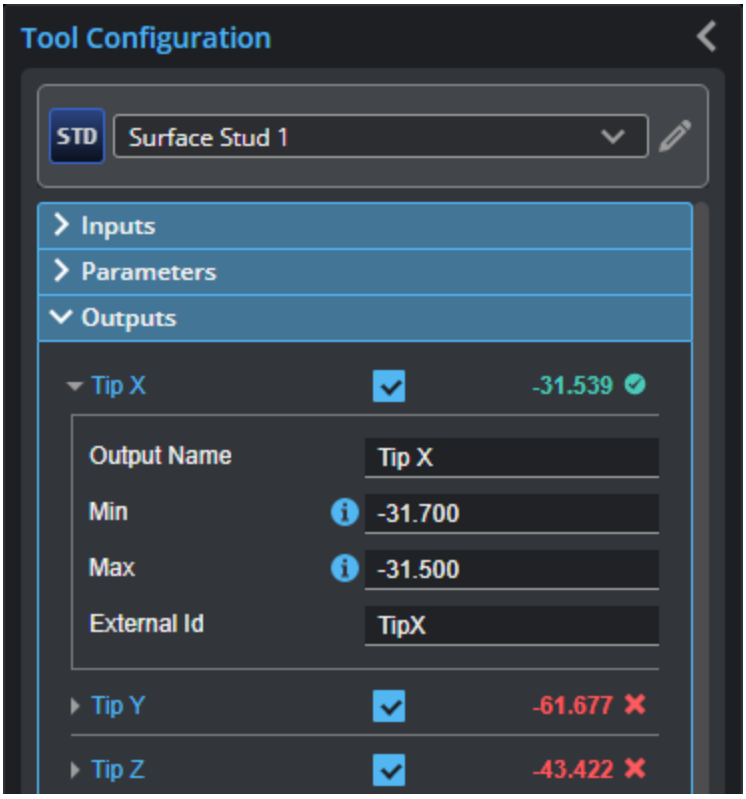


Parameters

Parameter	Description
Stud Radius	Expected radius of the stud.
Stud Height	Expected height/length of the stud.
Base Height	The height above the base surface that will be ignored when the (truncated) cone is fitted to the stud data.
Tip Height	The height from the top of the surface that will be ignored when the (truncated) cone is fitted to the stud data.
Radius Offset *(Radius measurement only)*	The distance from the tip of the stud from which the radius is measured.
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
Use Reference Region	When enabled, displays the **Reference Type** setting.

Parameter	Description
Reference Type	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.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

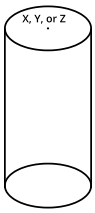
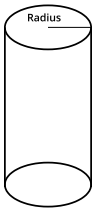


To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.	
Shaft X Determines the X position of the stud shaft. The position is defined with the parameter Radius Offset.	
Shaft Y Determines the Y position of the stud shaft. The position is defined with the parameter Radius Offset.	
Shaft Z Determines the Z position of the stud shaft. The position is defined with the parameter Radius Offset.	
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

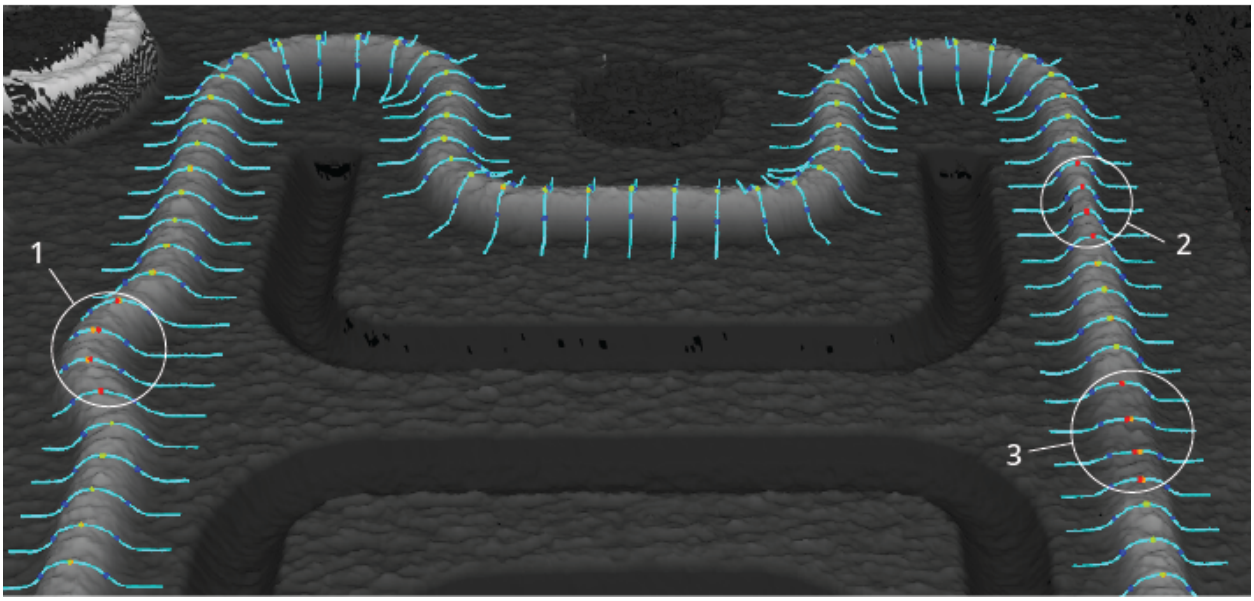
Type	Description
Tip Point	The center point of the stud tip.
Shaft Point	The center point of the stud shaft.
Base Point	The center point of the stud base.

For more information on geometric features, see *Geometric Features* on page 227.

Surface Track

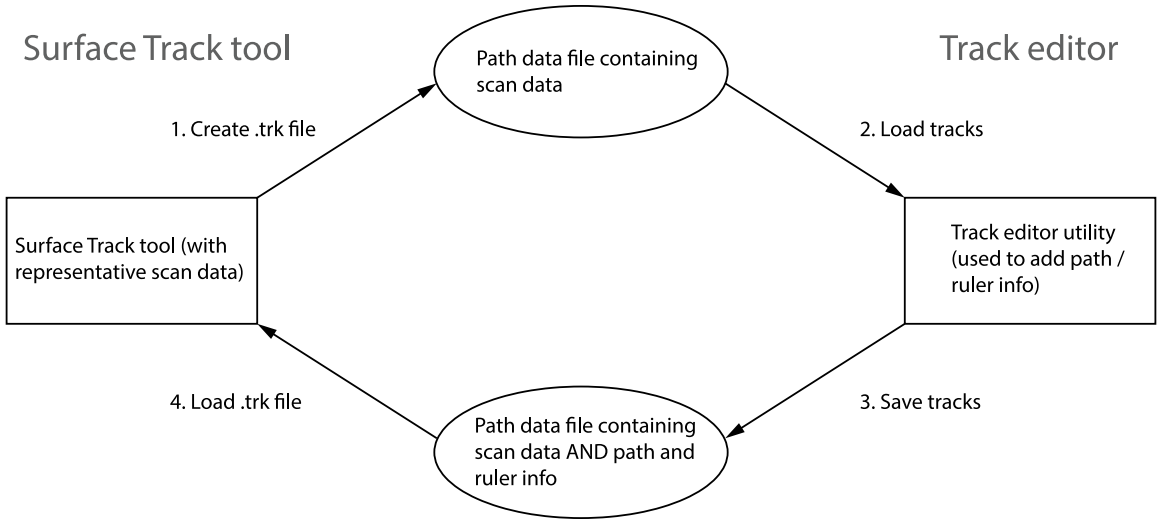
The Track tool lets you perform quality control and inspection along a path you define on representative scan data. The Track tool is especially useful for inspecting materials such as glue / sealant beads. The tool returns width and height measurements of the material, as well as OK and NG ("no good") 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 227).

> 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 through a PC instance of GoPxL. For more information, see *Running GoPxL on a Windows PC* on page 1188.



	Element	Description
1	Offset fails	The center points in this area of the track are too far off the defined path.
2	Width fails	The center points in this area of the track are too narrow.
3	Height failes	The center points in this area of the track are too low. (Note how the profiles are flatter than other profiles.)

You define the path along which the tool performs its measurements using a separate, PC-based utility (the "track editor"). The following shows the relationship between the Surface Track tool and the track editor.



For more information on the track editor, see *Track Editor* on page 1325.

> All instances of the Track tool on a sensor or in a PC instance of GoPxL share the same path file set in **File** (ending in .trk). 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, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Key Concepts

The following are important concepts for using both the track editor (see *Track Editor* on page 1325) and the Surface 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.

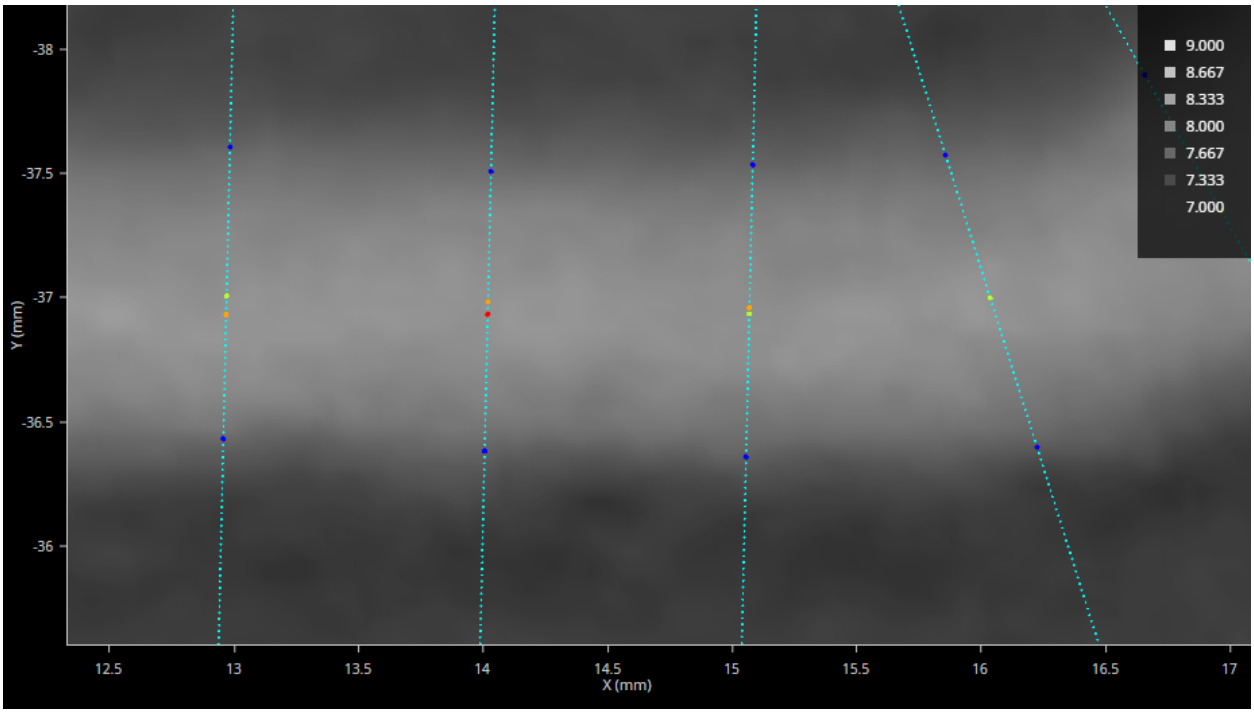
Path: The ideal center line of the track. You define the path in the track editor. You can define more than one path for use on scanned targets, but the Surface Track tool returns measurements for all paths.

Caliper / ruler: A caliper (in the tool) or ruler (in the track editor) is an area perpendicular to the path you define. You define the size (length and width) and spacing of the rulers in the track editor. The Surface Track tool extracts a profile from the surface data beneath a caliper and performs measurements based on the values you set in the tool's parameters.

Caliper / ruler profiles: The profiles extracted from the surface data under a caliper. The tool's 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 calipers. Colored dots provide important information (see below).



Track tool in data viewer, showing a track (lighter grey horizontal strip in the middle), calipers running perpendicular to the track (light blue dots). Dots of other colors provide additional information (see below).

When you enable **Show Measurement Results**, the tool displays dots on the calipers to provide the following information (see also the images below):

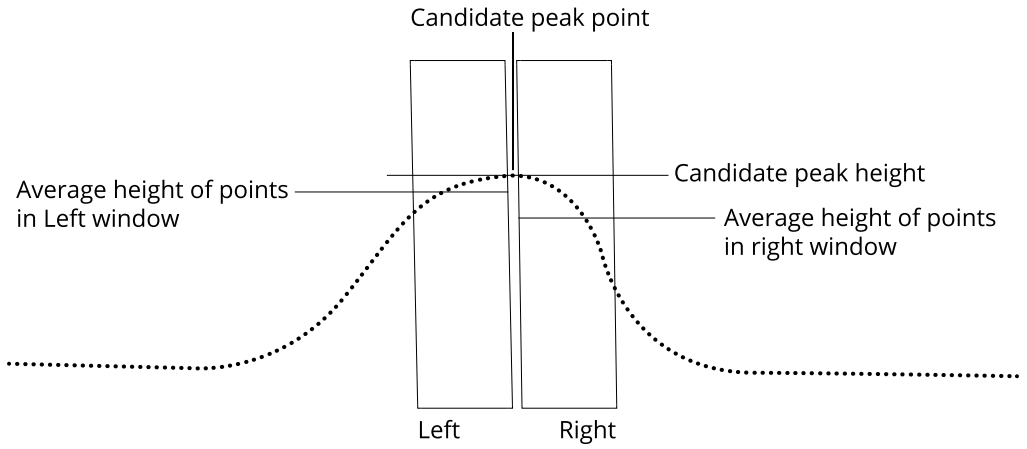
- Light blue dots: The data points in the caliper profile. When you enable **Show Path and Calipers**, the tool displays a white line centered on these dots to indicate the location of the caliper.
- Dark blue dotes: The detected sides of the track. These represent the width of the track under that caliper.
- Green dot: The center point (between the two sides of the track) on a caliper that *passes* all criteria set in the tool. These count toward the "OK Count" measurement.
- Red dots: The center point (between the two sides of the track) on a caliper that *fails* at least one of the criteria set in the tool. These count toward the "NG Count" measurement. Causes for failures include the width and the height of the track, as well as the area under the caliper profile.
- 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.

Track Location

The tool attempts to locate the track using the profile data it extracts under each caliper, and does this by first locating the "peak" (the highest point on the caliper 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 caliper 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* 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 caliper profile the tool searches for edge points. The tool uses this value to limit where the edge of the track might be detected.

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 Surface 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 on 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 981.

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 and Working with Scan Data* on page 104.

2. Add a Surface Track tool.

   For more information on adding a tool, see *Adding a Tool* on page 164.

3. In the Surface Track tool, choose **Create** from the **Operation** drop-down.

   The tool creates a file (for example, SurfaceTrack-0000.trk) containing scan data. Creating the file may take a few seconds.

4. Launch the track editor and configure the path or paths.

   For information on using the track editor, see *Track Editor* on page 1325.

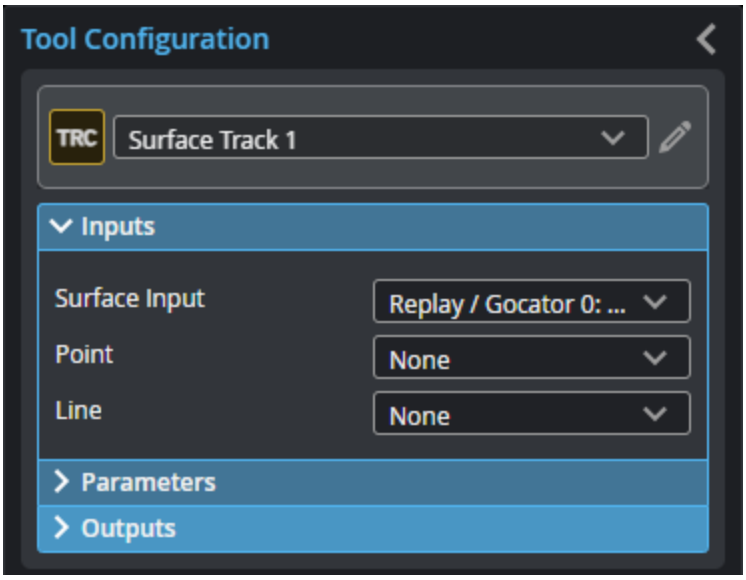
5. After you have finished editing the track data in the track editor, in the Surface Track tool, choose **Load** in the **Operation** drop-down to load the path data you just created.

6. Configure the Track tool as required.

   For information on the tool's measurements and settings, see the below.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

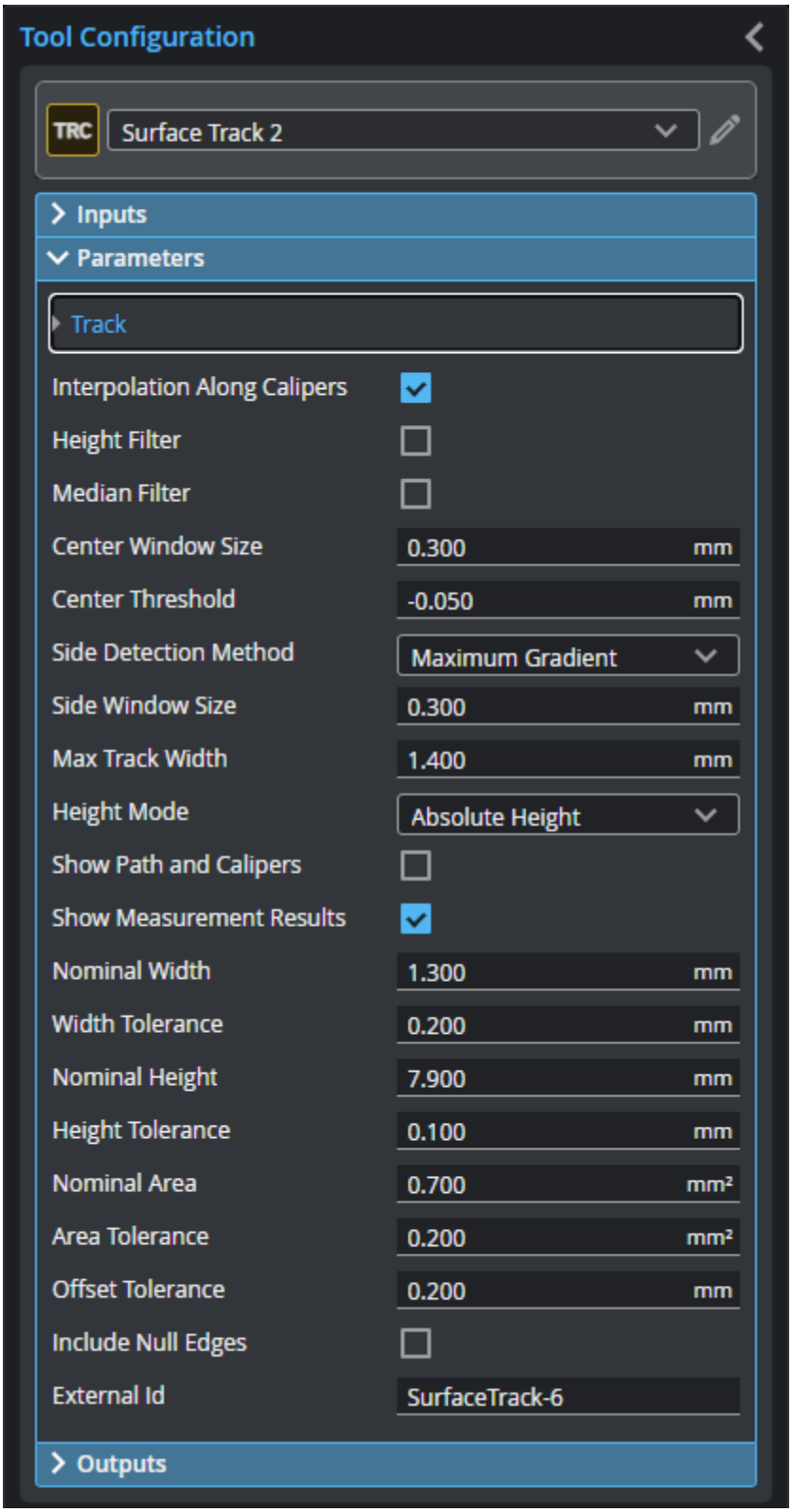


Inputs

Name	Description
Surface Input	The data the tool applies measurements to or processes.
Point Line	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 227.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.


Tool Configuration
TRC
Surface Track 2
Inputs
Parameters
Track
Interpolation Along Calipers
Height Filter
Median Filter
Center Window Size
0.300
mm
Center Threshold
-0.050
mm
Side Detection Method
Maximum Gradient
Side Window Size
0.300
mm
Max Track Width
1.400
mm
Height Mode
Absolute Height
Show Path and Calipers
Show Measurement Results
Nominal Width
1.300
mm
Width Tolerance
0.200
mm
Nominal Height
7.900
mm
Height Tolerance
0.100
mm
Nominal Area
0.700
mm²
Area Tolerance
0.200
mm²
Offset Tolerance
0.200
mm
Include Null Edges
External Id
SurfaceTrack-6
Outputs

Main Parameters

Parameter	Description
Track	When expanded, displays file-related operations.
File	The track file that contains scan and path data. You add paths and calipers to the file using the track editor. For more information on the track editor, see *Track Editor* on page 1325.
Operation	Provides operations related to the track file. One of the following: • **Normal**: Selected by the tool after you perform another file operation. • **Create**: Creates a new track 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**: Refreshes the list of files.
Interpolation Along Calipers	Enables linear interpolation on the profile extracted from the calipers to achieve sub-pixel accuracy in the width and height measurements.
Height Filter *Dependent settings* 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 caliper profiles.
Median Filter *Dependent settings* Window Size	When **Median Filter** is enabled, specify the window the tool will use to smooth the height values of the points in the caliper profiles in the **Window Size** setting.
Center Window Size	The size of the left and right windows the tool moves along the caliper profile to detect whether the point centered between the two is the highest point along a caliper (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 caliper 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.

Parameter	Description
	It may be necessary to use a negative number in some cases, for example, when the top point slightly dips below its surroundings.
Side Detection Method *Dependent settings* Side Window Size Max Track Width Side Height Threshold	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 *Side Detection* on page 983.
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 (all of the scan data). **Step Height** - Height values are relative to the surrounding area of the track.
Show Path and Calipers	Displays the path and calipers (as defined in the track editor) on the scan data.
Show Measurement Result	Shows dots on each caliper representing the results of the measurements on the profile extracted from the surface data under the caliper. For more information, see *Key Concepts* on page 981.
Nominal Width	The expected width of the track.
Width Tolerance	The tolerance applied to the nominal width.
Nominal Height	The expected height of the track. The **Height Mode** setting determines how the height is interpreted (absolute or step); for more information, see *Height Mode* above. This setting applies to the peak point, not the center point.
Height Tolerance	The tolerance applied to the nominal height. This setting applies to the peak point, not the center point.
Nominal Area	The expected cross-sectional area under the calipers on the track.
Area Tolerance	The tolerance applied to the nominal area.
Offset Tolerance	The maximum allowed distance between the center (highest) point on a ruler and the path. This setting applies to the center point.
Include Null Edges *Dependent settings* Null Fill Value	When enabled, displays a **Null Fill Value** parameter you can set to replace null values with the value.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

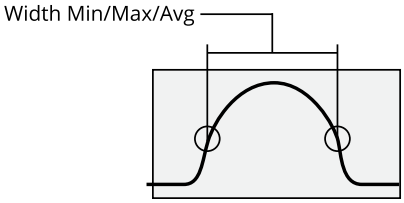
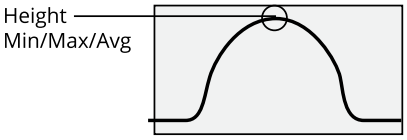
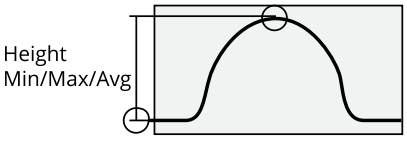


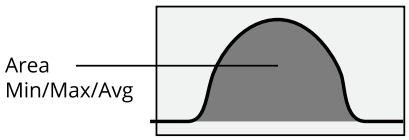
To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

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.")	
Min Width **Max Width** **Mean Width** These measurements return the minimum, maximum, and mean width of the track.	 *Width measurements on a ruler profile. The Track tool's settings determine the locations of the "sides" of the track.*
Min Height **Max Height** **Mean Height** These measurements return the minimum, maximum, and mean 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.*

Measurement	Illustration
Min Area **Max Area** **Mean Area** These measurements return the minimum, maximum, and mean area under the rulers.	 *Area measurements under a ruler profile.*
Width **Peak Height** **Offset** Arrays containing the widths, peak heights, and offsets of each ruler.	

Features

Type	Description
Center Point	An array of Point geometric feature representing the center point of each ruler.

For more information on geometric features, see *Geometric Features* on page 227.

Data

Type	Description
Diagnostic Surface	Surface data created by combining the extracted profiles. Use for diagnostics.

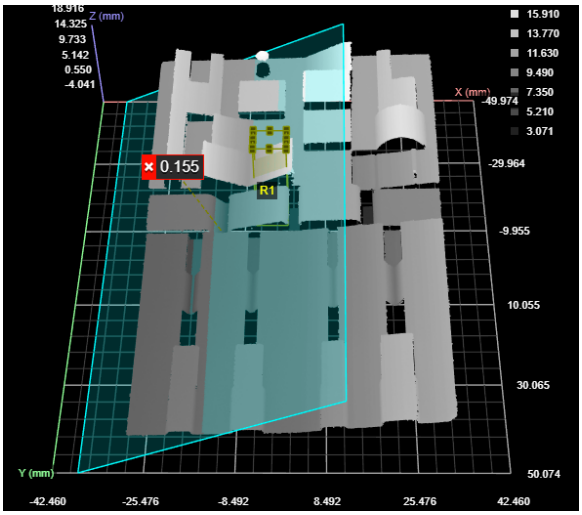
Training Archive

The Training Archive tool lets you conditionally add frames to a Surface Anomaly Detector project. Training Archive is only intended to be used with Surface Anomaly Detector. For more information, see *Training Archive* on page 742.

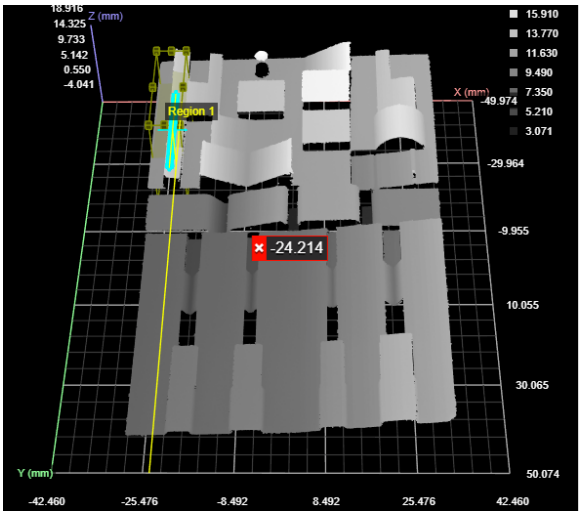
Surface Transform

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.

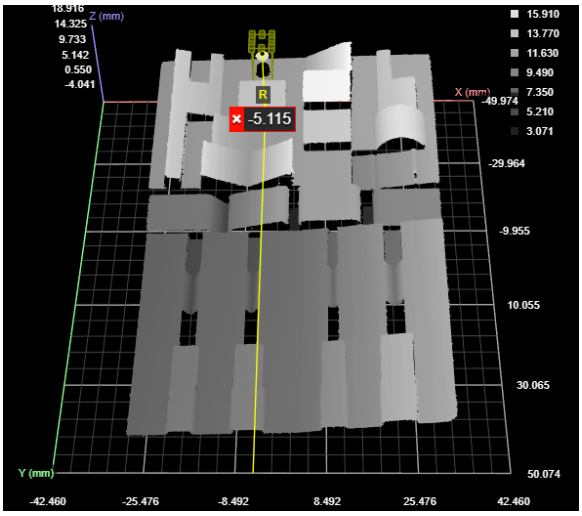
In *Combinations of geometric feature inputs and results* on page 995, the following geometric features are used by a Surface Transform tool in various combinations (a plane, a line, and a point).



A Surface Plane tool, with the region set to a small left-facing angled surface

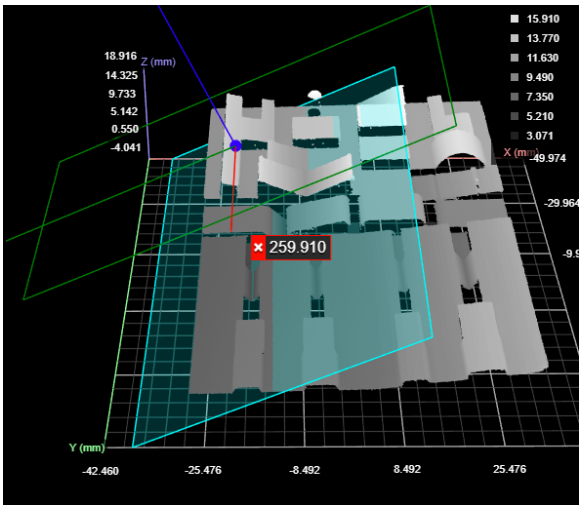


A Surface Edge tool, with the region set to the left edge of a raised surface (upper left of data viewer).



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

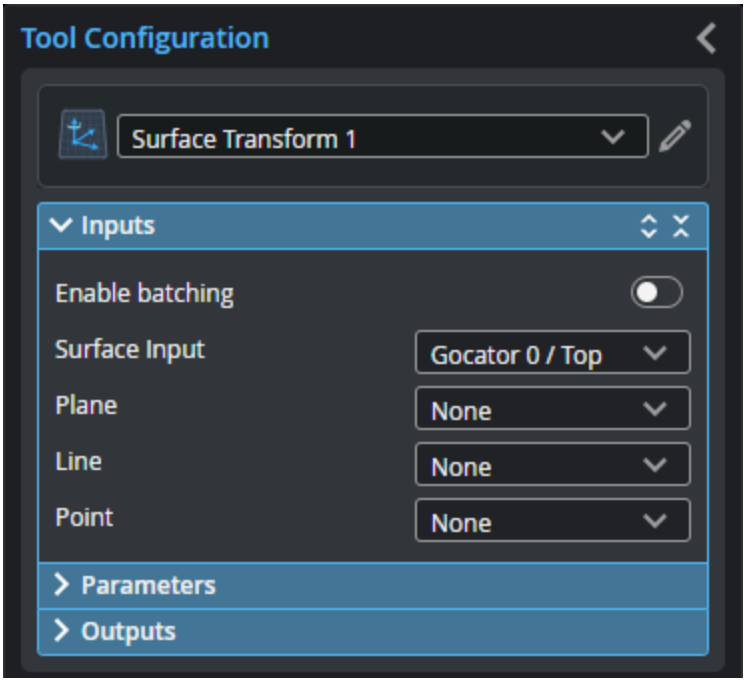
The new origin is represented by the dark blue dot at the intersection of the transformed axes.

Plane

The new plane is represented by the cyan rectangle.

> To switch between the original and transformed data, choose the output under the data viewer or in the tool's list of outputs.

Inputs



Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.
Plane Line Point	The Plane, Line, and Point geometric features the tool uses to perform transformations. The required geometric features depends on your application. For more information, see *Combinations of geometric feature inputs and results* below.

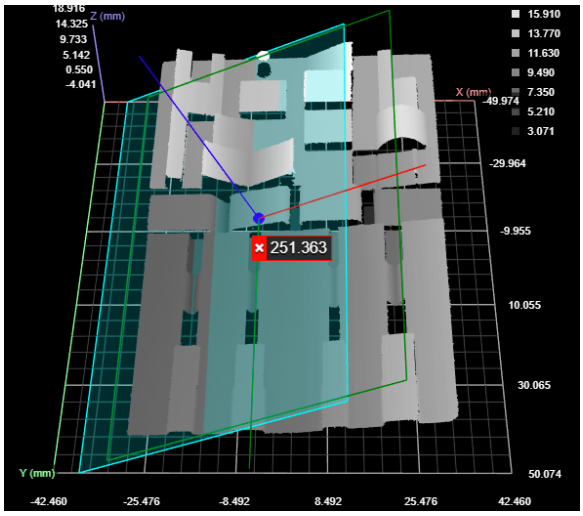
Combinations of geometric feature inputs and results

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.

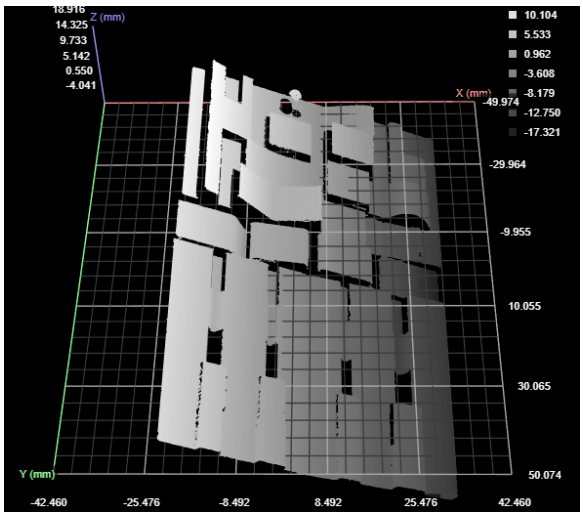
PLANE

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



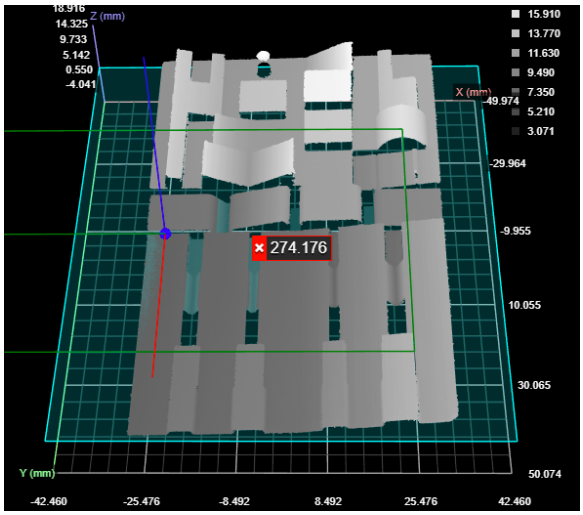
Transformed data



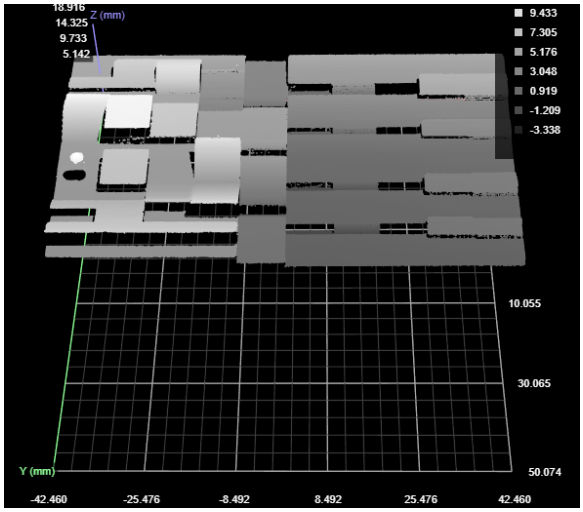
LINE

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

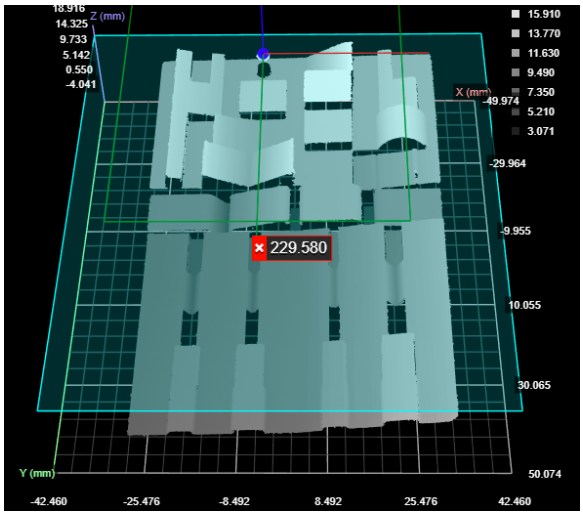


The direction of the X axis depends on the tool generating the line that Surface Transform takes as input. You may need to adjust the direction using the **Add Fixed Transform** settings.

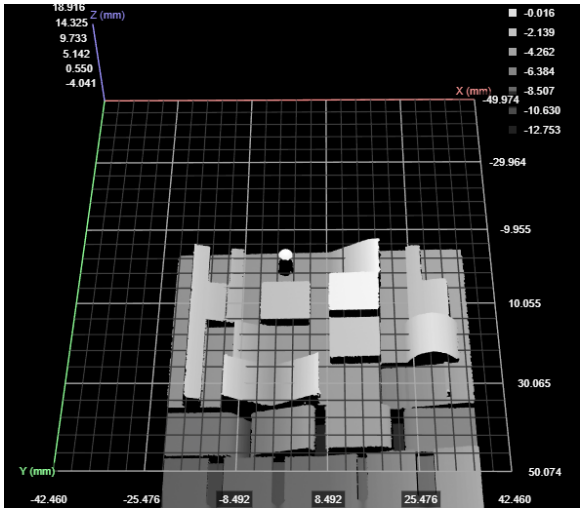
POINT

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



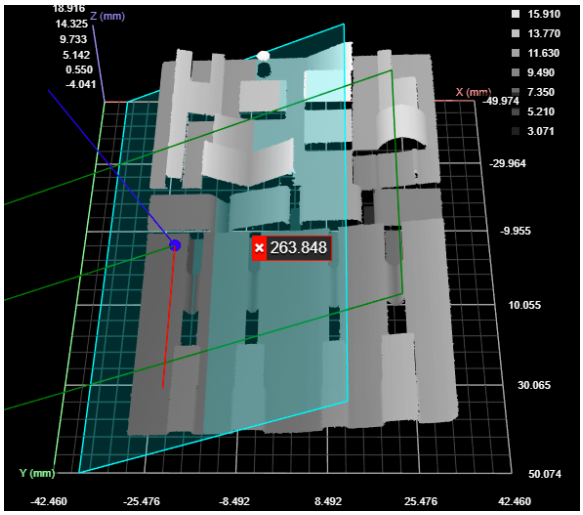
Transformed data



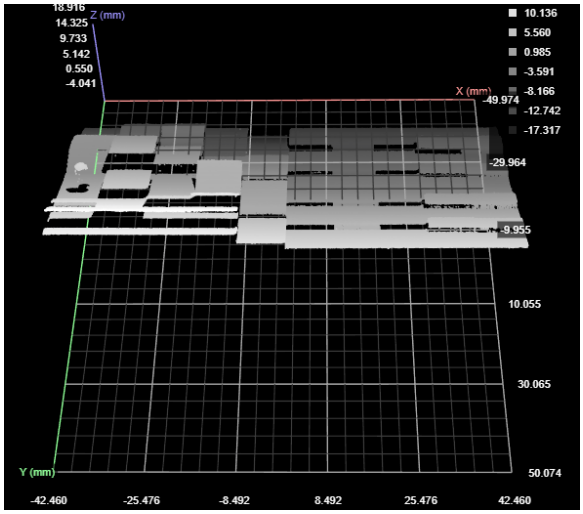
PLANE + LINE

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



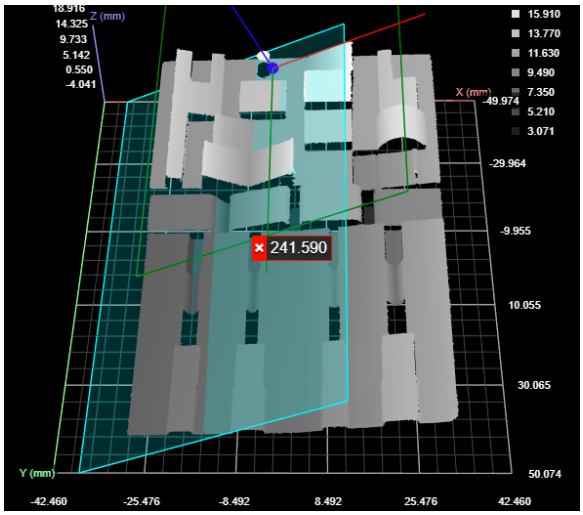
Transformed data



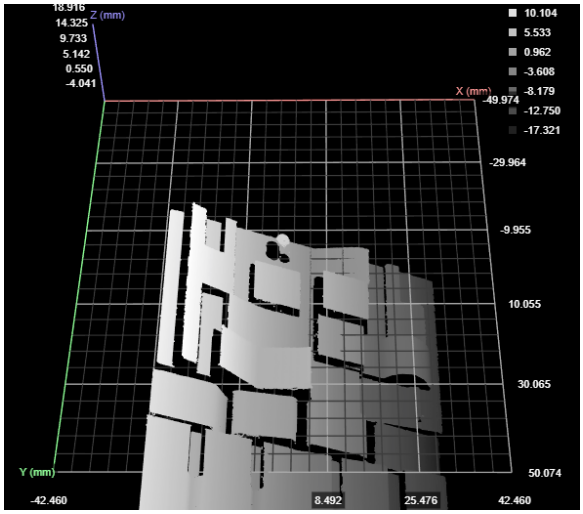
PLANE + POINT

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



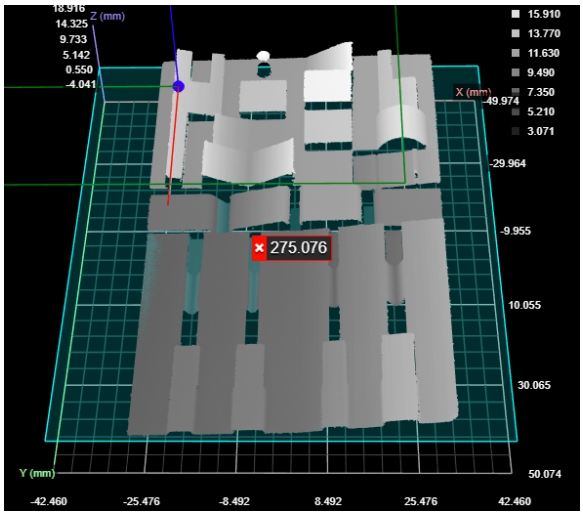
Transformed data



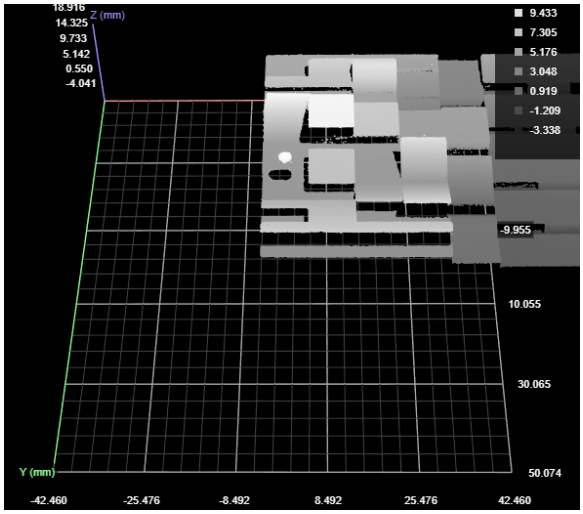
LINE + POINT

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



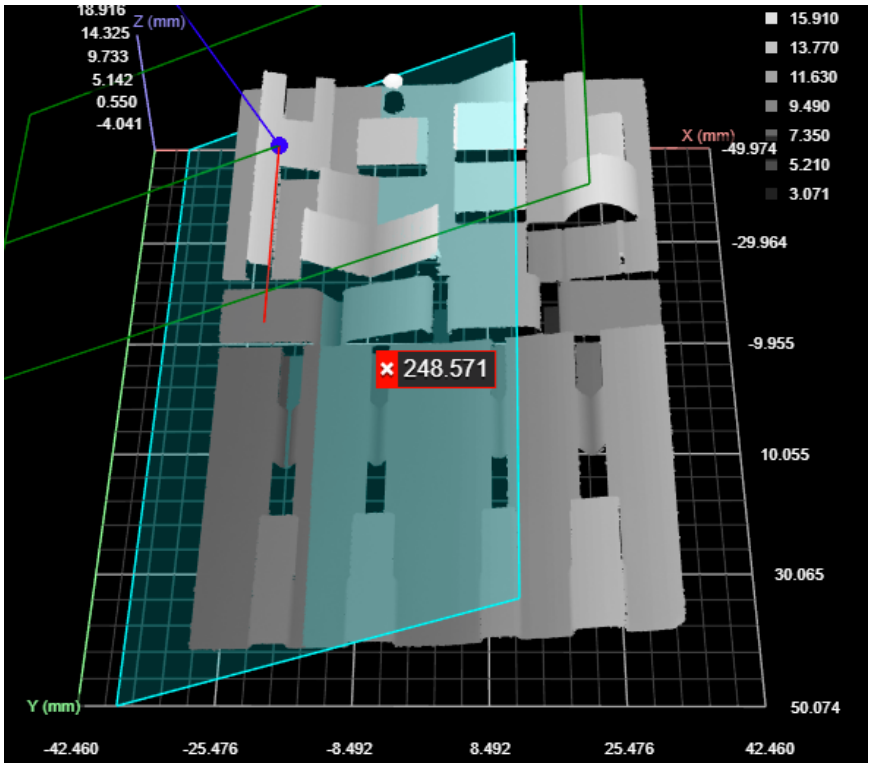
Transformed data



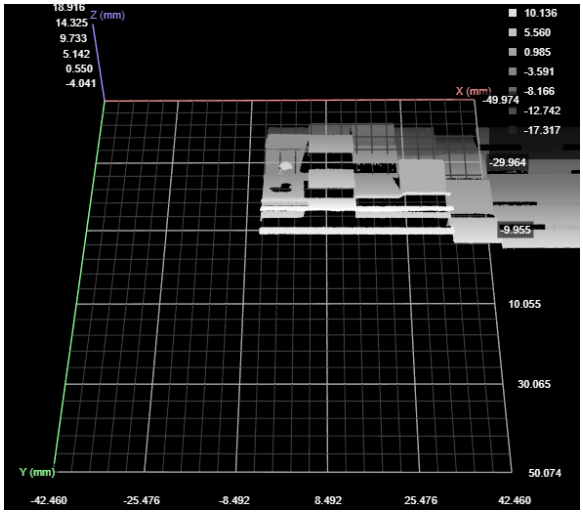
PLANE + LINE + POINT

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



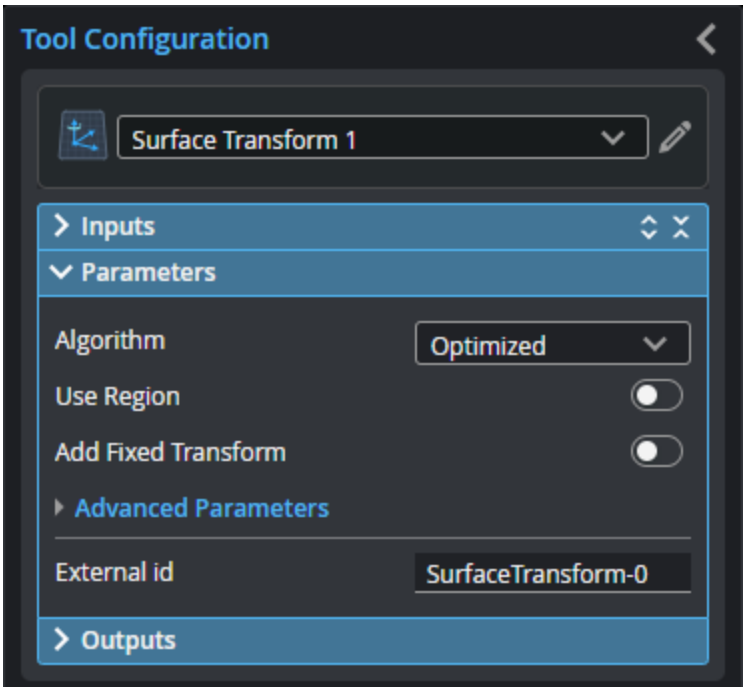
Transformed data



For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.

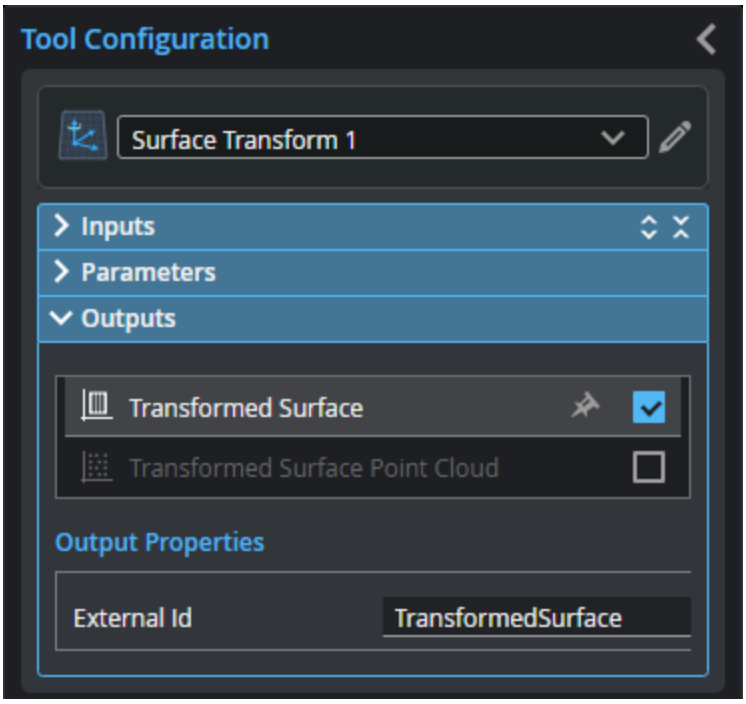


Parameters

Parameter	Description
Algorithm	One of the following: **Optimized** The tool uses an updated algorithm with additional parameters. This algorithm can be 30% to 40% faster than the "Classic" algorithm. This algorithm can also output point cloud data in addition to uniform data. For more information, see below. **Classic** The tool uses the algorithm from GoPxL 1.5 and earlier.
Use Region	When enabled, displays Region parameters (see below). When disabled, the tool uses all data.
Region	The region to which the tool's measurements will apply. For more information, see *Regions* on page 214.
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.
Resolution Mode	On G3 sensors, leave this set to the default **Original Resolution**.

Parameter	Description
(under expandable **Advanced Parameters** section when **Algorithm** is set to Optimized)	
Bounding Box Mode (under **Advanced Parameters** when **Algorithm** is set to Optimized)	Specifies the bounding box of the output surface data. One of the following: **Preserve Input Bounds** Uses the same bounding box from the input source. **Transform Input Bounds** Transforms the bounding box of the input message. Could be changed from frame to frame when the the geometric feature inputs are set in the **Inputs** section. **Fit to Valid Points** Calculates a tight bounding box based on the transformed valid points. The calculated bounding box will usually change from frame to frame.

Outputs



Data

Type	Description
Transformed Surface	The transformed surface.

Type	Description
Transformed Surface Point Cloud	

Surface Vibration Correction

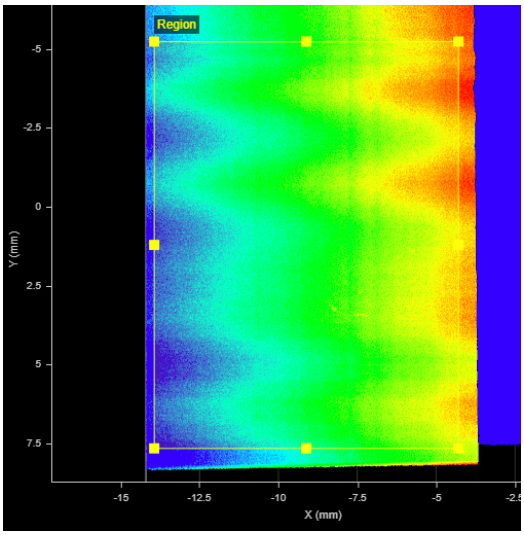
The Vibration Correction tool removes height (Z) variations along the Y and X axes in Surface data caused by vibrations in the transport or motion system, or unstable sensor mounting. The tool is useful for improving repeatability and accuracy of measurements. Corrected Surface data is used as input for other tools.

Vibration is primarily expected in the Y direction, as varying Z offsets of profiles or rows, which the tool corrects for.

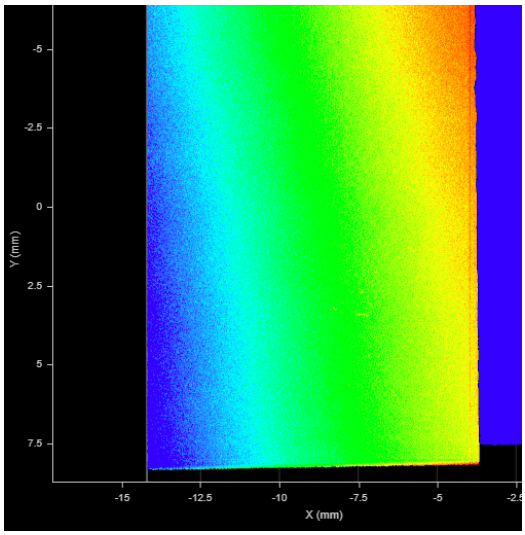
The tool also operates independently along the X axis, which corrects any consistent patterns of Z offset for entire columns of the Surface data.

There is no support for removing diagonal vibration patterns.

> 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

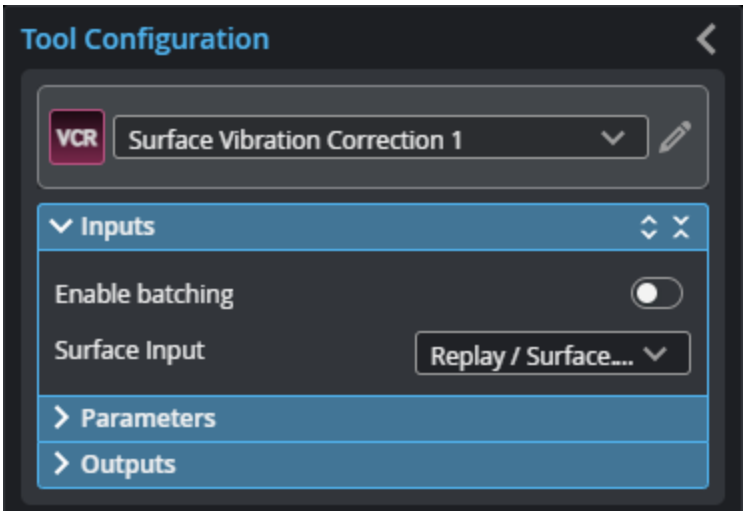


Corrected surface data: a better representation of the actual target

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.

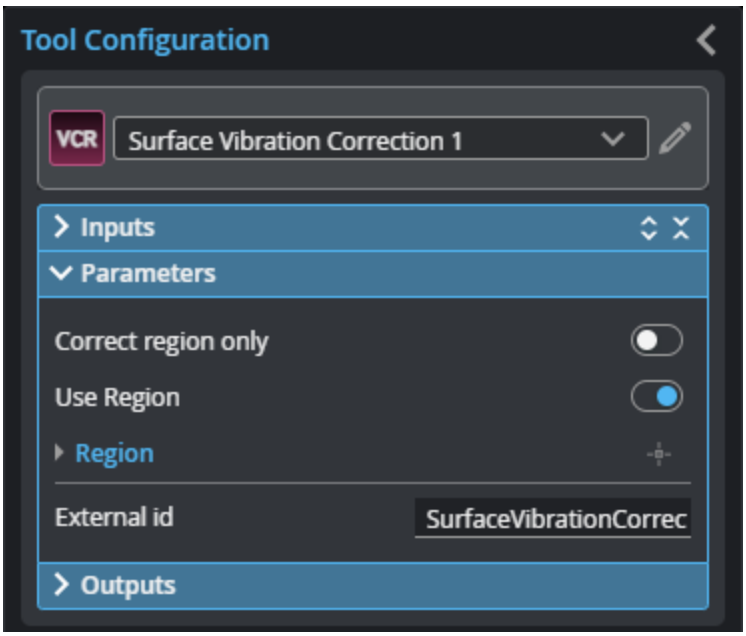


Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Surface Input	The data the tool applies measurements to or processes.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

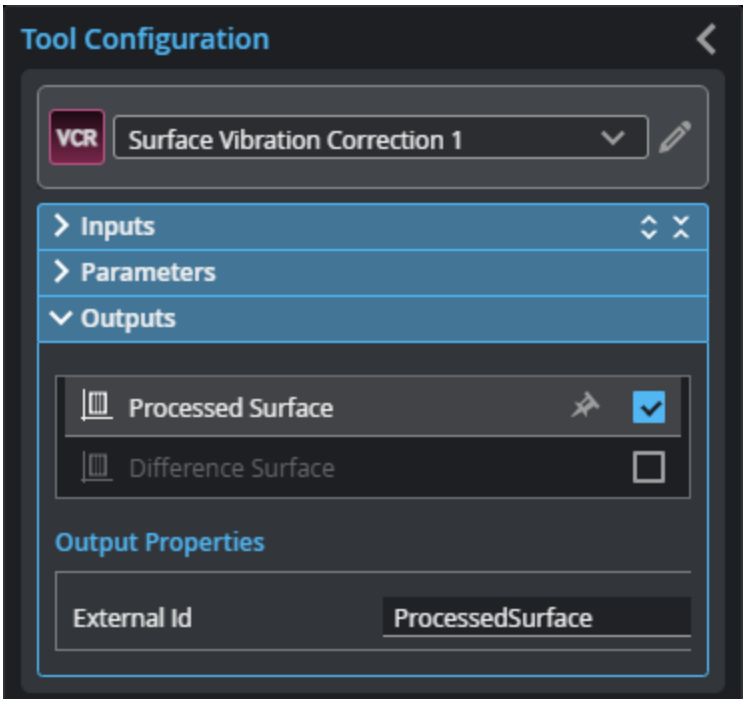


Parameters

Parameter	Description
Correct Region Only	If enabled, only the area under the region is corrected for vibration in the Surface data the tool outputs. This setting can be useful if vibration regularly occurs in a specific area of the scan data. When disabled, the tool applies correction to all 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 (by enabling **Correct Region only**).
Region	The region whose data the tool will use to calculate the vibration correction. If **Correct Region Only** is enabled, the tool applies vibration correction *only* to the region. If **Correct Region Only** is disabled, the tool applies vibration correction to all of the scan data.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.



Data

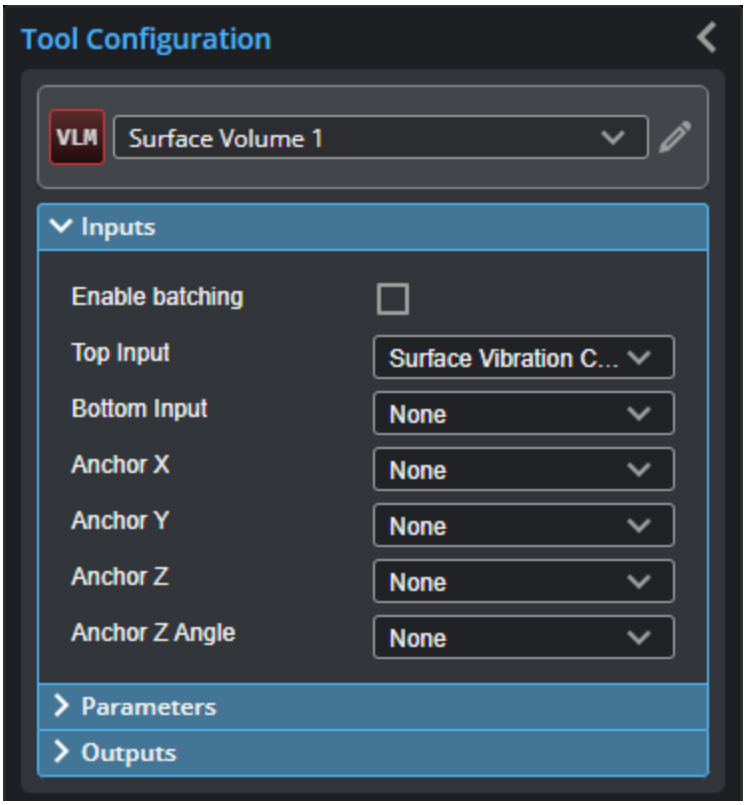
Type	Description
Processed Surface	Surface data corrected for vibration, available for use as input in other tools.
Difference Surface	Diagnostic Surface data showing the difference between the corrected surface and the original.

Surface Volume

The Volume tool returns the volume, area, and thickness of a part.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs



To use a measurement as an anchor, it must be enabled and properly configured in the tool providing the anchor. For more information on anchoring, see *Measurement Anchoring* on page 209.

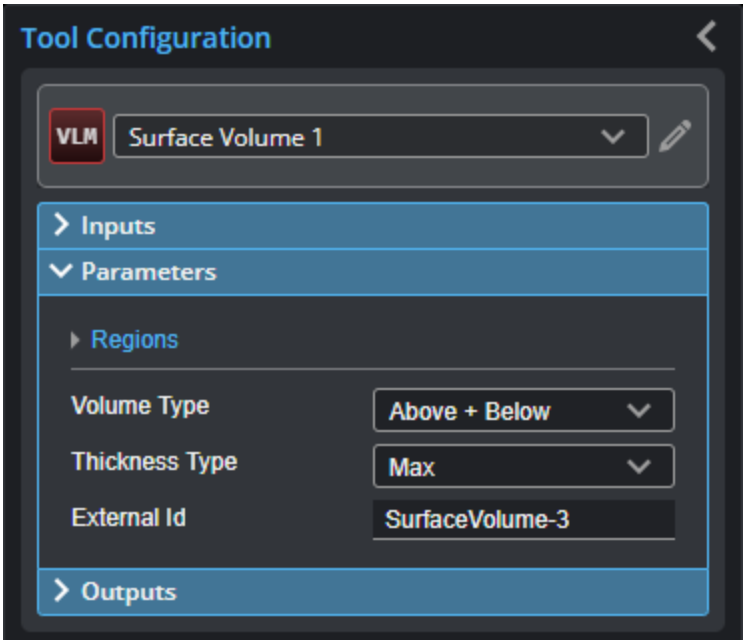
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Top Input	The data the tool applies measurements to or processes.
Bottom Input	This tool can optionally take a second, bottom input. When **Enable Batching** is checked, both inputs take an array.

Name	Description
Anchor X Anchor Y Anchor Z	Positional anchors are optional.
Anchor Z Angle	The angle measurement of another tool to use as a rotational anchor for this tool. Rotational anchors are optional.

Parameters

The following parameters are in the expandable **Parameters** section in the tool's configuration.



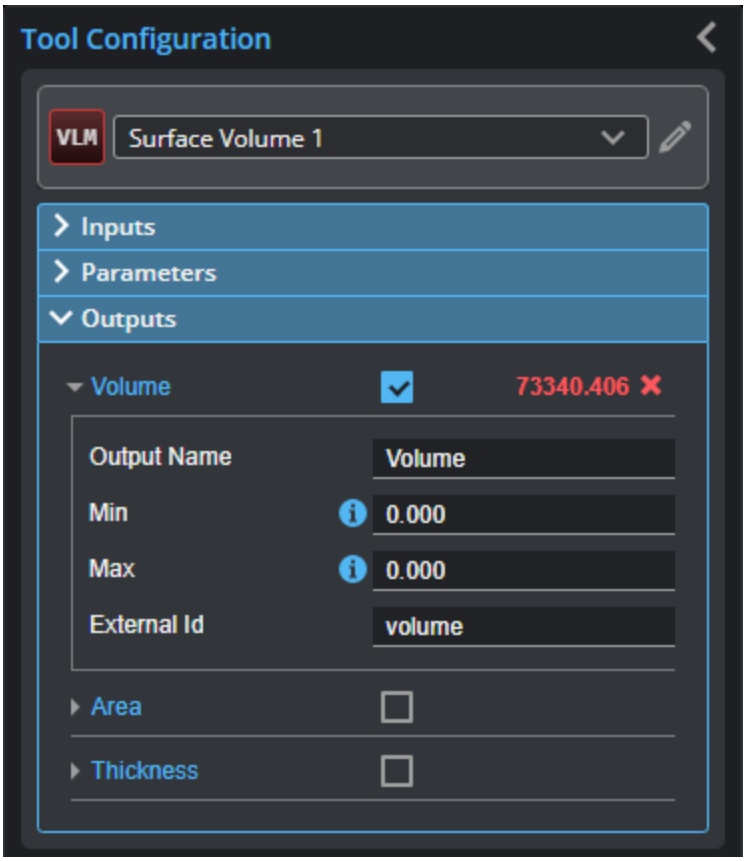
Parameters

Parameter	Description
Regions	When expanded, displays the region- and mask-related settings.
Enable	Enables regions and displays the region- and mask-related settings (see below).
Mask Mode Region Type 1 Region 1	When you enable regions (see above), the tool displays additional settings related to the region type. For details on the regions supported by this tool and their settings, see *Flexible Regions* on page 217. For general information on regions and the difference between standard and "flexible" regions, see *Regions* on page 214.
Volume Type	Used to compute different kinds of surface volumes. One of the following: • Above • Below • Above + Below (default)

Parameter	Description
	• Above - Below Above means the volume calculated using Surface data that is above the Z = 0 plane. Below means the volume calculated using Surface data that is below the Z = 0 plane. Note that Above and Below can't be negative, but Above - Below can be negative if Below > Above.
Thickness Type *(Thickness measurement only)*	Determines which feature in the Surface data the tool uses to calculate thickness. One of the following: • Max • Min • Average • Median • Centroid 2D (height of the centroid in the XY plane) • Centroid 3D (height of the centroid in the XYZ space).
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

Most tools provide measurements, geometric features, or data as outputs.

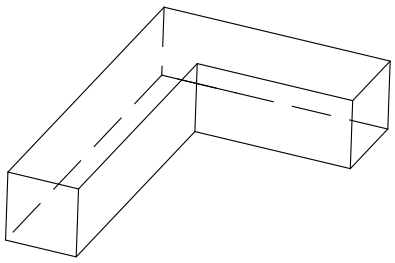


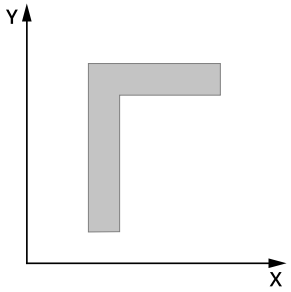
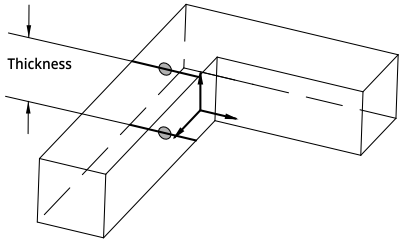
To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Measurements

Measurement	Illustration
Volume Measures volume in XYZ space.	

Measurement	Illustration
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 *Surface Segmentation* on page 951 and *Surface Blob* on page 760.	
Thickness Measures thickness (height) of a part. Uncheck this measurement if you don't need it.	

User Data Input Tool

The User Data Input tool lets you send data to GoPxL from PLCs over EtherNet/IP, PROFINET, Modbus TCP, or Ethernet ASCII, or via the SDK. You configure the tool to indicate the data types and their offsets GoPxL is receiving from the PLC, and the tool makes the data available in its outputs. A typical use case for this is to use the tool's string output as part of file names output with the data storage service, for example, part ID information. For information on the data storage service, see *Data Storage Service* on page 1158.

Note that the user data input buffer is shared by *all* protocols.

For information on EtherNet/IP, see *EtherNet/IP Protocol* on page 1038.

For information on PROFINET, see *PROFINET Protocol* on page 1126.

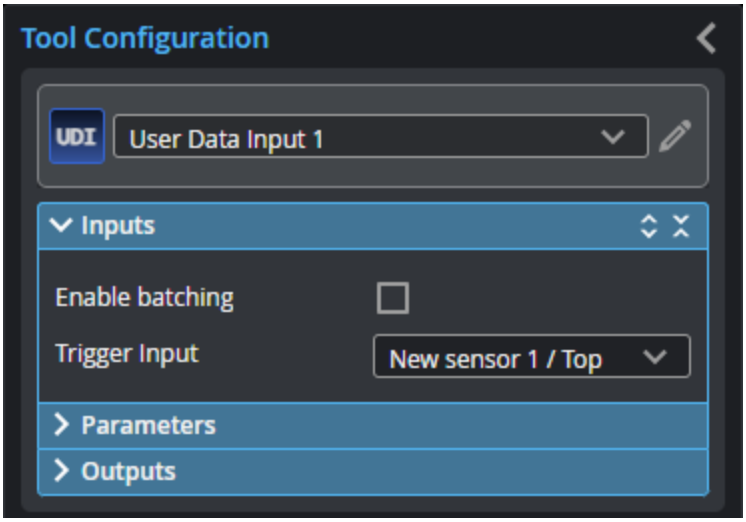
For information on Modbus, see *Modbus TCP Protocol* on page 1027.

For information on Ethernet ASCII, see *Ethernet ASCII Protocol* on page 1107.

For information on adding, managing, and removing tools, as well as detailed descriptions of settings common to most tools, see *Tool Configuration* on page 196.

Inputs

You configure the tool's inputs in the expandable **Inputs** section.



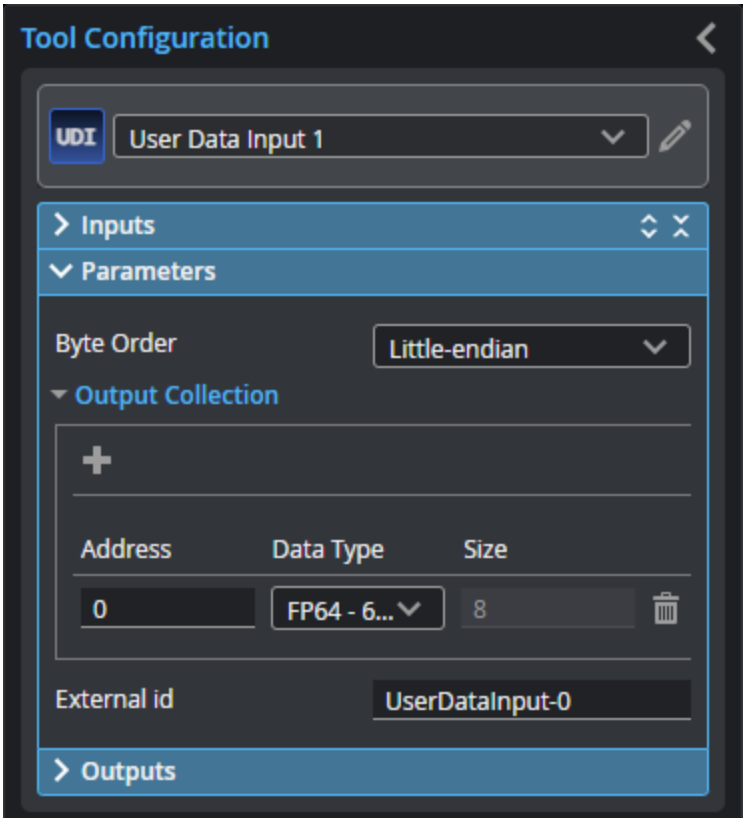
Inputs

Name	Description
Enable Batching	When **Enable Batching** is checked, the tool takes an array as input and processes each element in the array individually. There is no limit to the size of the array, other than the processing limitations of the sensor. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with a maximum of two elements. When **Enable**

Name	Description
	Batching is unchecked, the tool takes either a single surface, or an array of surfaces with a maximum of two elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. When **Enable Batching** is unchecked, the tool takes either a single profile, or an array of profiles with any number of elements. When **Enable Batching** is unchecked, the tool takes either a single surface, or an array of surfaces with any number of elements. If the input is an array, the tool will aggregate the data in the array, and process that data as a whole. For more information on arrays, batching, and aggregating, see *Arrays, Batching, and Aggregation* on page 199.
Trigger Input	The input that triggers the tool processing. This tool can optionally take an array as input. For more information, see Arrays, Batching, and Aggregation.

Parameters

You configure the tool's parameters in the expandable **Parameters** section.

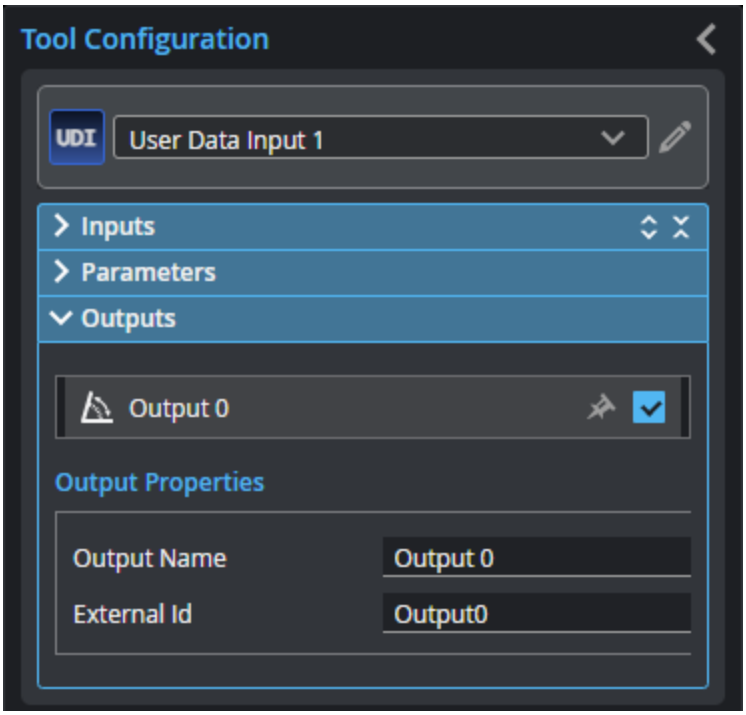


Parameters

Name	Description
Byte Order	The "endianness" of the data being injected into the tool chain. For PLCs, consult the PLC manufacturer's documentation.
Output Collection	An expandable section containing a list of the data GoPxL is receiving. For each piece of data the tool is receiving, set the address and data type.
External ID	The external ID of the tool that appears in GoHMI Designer. For more information, see *GoHMI and GoHMI Designer* on page 1210.

Outputs

This tool outputs data received from a PLC.



To set whether measurement values are considered a Pass or Fail, select a measurement in the outputs list, and then enable and configure its **Min** and **Max** parameters in the **Output Properties** section; you can enable one or the other, or both.

By default, **Min** and **Max** are both disabled, and a measurement is considered a Pass. For more information, see *Decisions* on page 229.

You can also apply scale and offset factors to measurement values. For more information, see *Output Filters* on page 228.

Data

Type	Description
Output {n}	The outputs available to other tools.

Communicate - PLC, Digital, GDP, HMI

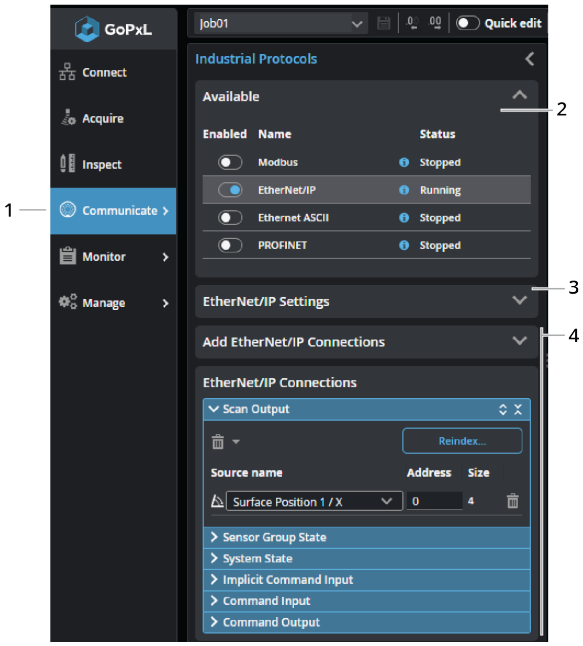
External devices such as PLCs (programmable logic controllers) can control Gocator sensors using various communication protocols and hardware interfaces. Sensors can also send scan data, measurement results, and measurement decisions to these devices. PLCs can in turn control other devices such as ejection chutes.

You can also use the GoPxL Data Protocol (GDP) with software written using the GoPxL SDK and REST API to control a sensor and receive data.

Lastly, on sensors and smart cameras, GoPxL can send acquired data to a remote FTP server. On GoMax Smart Vision Accelerators and GoPxL for PC, GoPxL can also save data to a local folder.

> If you are using a GoMax Smart Vision Accelerator, only connect Ethernet-based PLCs or PCs to the ETH1 port on the GoMax device (whether via a switch or not).

The following sections describe the **Communicate** category and the available protocols and interfaces.



To select the communication type, expand **Communicate**, and choose the required type. The last selected type remains if you switch to a different category or page.

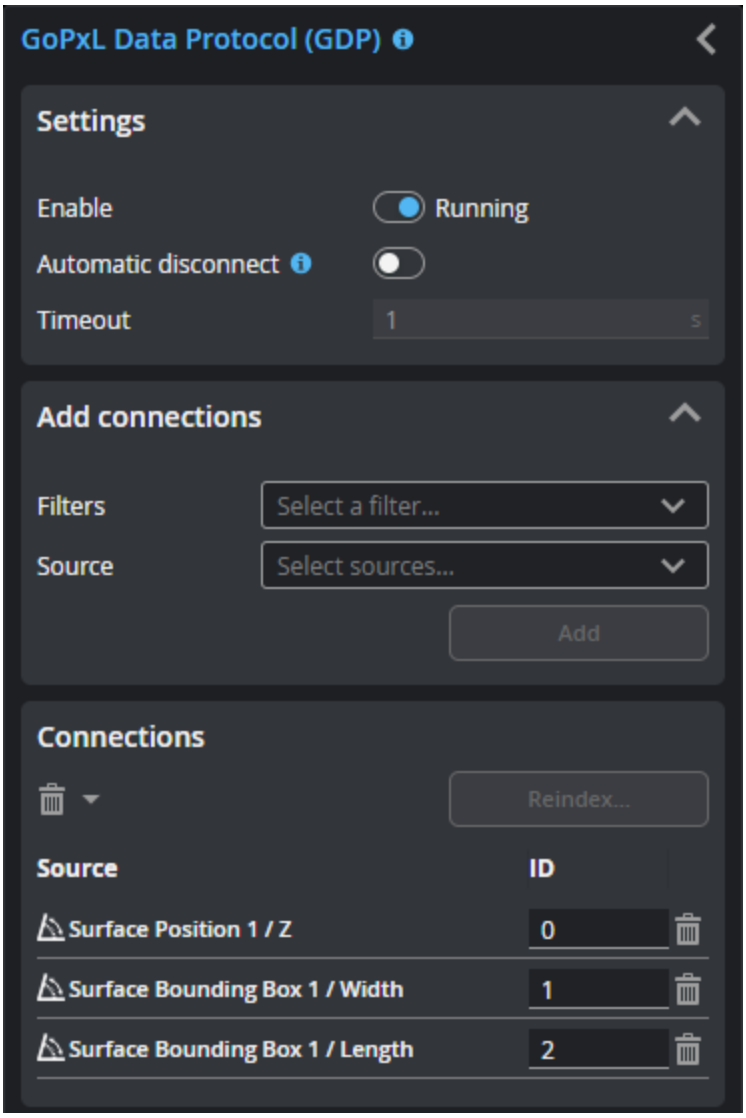
	Element	Description
1	Communicate category	Pages corresponding to the different control and data output methods available in GoPxL. For information on the available control methods, see *Communication interfaces* on the next page.

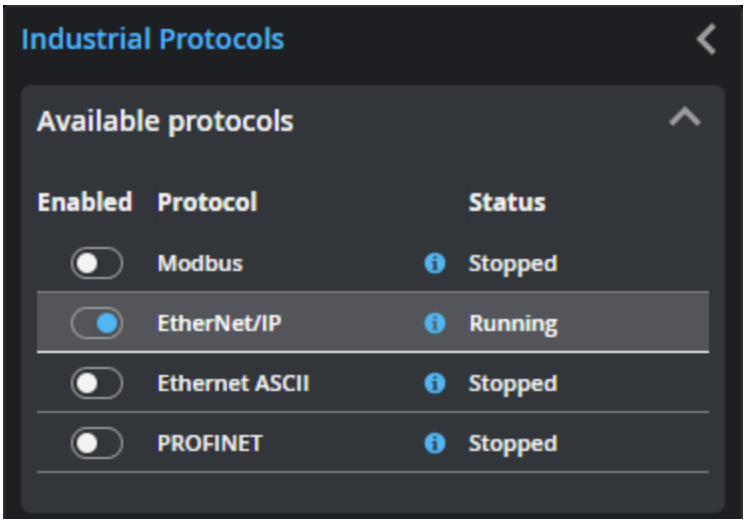
	Element	Description
2	Available Protocols	When you are on the Industrial page, this displays the available industrial protocols and whether a protocol is enabled. Use the toggles to start a protocol.
3	Settings	Lets you configure protocols.
4	Connections	Lets you add and remove the connections (such as stamps, measurements and decisions) to the connections map (industrial protocols) and outputs (Gocator protocol).

For more information on the available communication methods, see *Communication interfaces* below.

Communication interfaces

Interface	Description
GDP	The **Communicate** > **GDP** page lets you enable and configure the GoPxL Data Protocol, which sends to data and measurements to external applications that use the GoPxL SDK. For more information, see *GoPxL Data Protocol* on page 1023.

Interface	Description
	 For information on configuring the GoPxL Data Protocol in GoPxL, see *GoPxL Data Protocol* on page 1023. For information on the SDK and REST API, see *GoPxL SDK and REST API* on page 1234.
Industrial	The Industrial page lets you enable and configure the following protocols over an Ethernet connection: • Modbus • EtherNet/IP • Ethernet ASCII • PROFINET

Interface	Description
	 **Modbus** The Modbus protocol lets you communicate with a sensor from a PLC that supports Modbus. For more information on configuring the Modbus protocol in GoPxL, see *Modbus TCP Protocol* on page 1027. For the Modbus protocol specification, see *Modbus TCP Protocol Format - PLC Information* on page 1031. **EtherNet/IP** The EtherNet/IP protocol lets you communicate with a sensor from a PLC that supports EtherNet/IP. For more information on configuring the EtherNet/IP protocol in the GoPxL UI, see *EtherNet/IP Protocol* on page 1038. For the EtherNet/IP protocol specification, see *EtherNet/IP Protocol Format - PLC Information* on page 1045. **Ethernet ASCII** The Ethernet ASCII protocol lets you communicate with a sensor using an ASCII format. For more information on configuring the ASCII protocol over Ethernet in the GoPxL interface, see *Ethernet ASCII Protocol* on page 1107. For a list of the commands, see *Ethernet ASCII Protocol Format - PLC Information* on page 1112. **PROFINET** The PROFINET protocol lets you communicate with a sensor from a PLC that

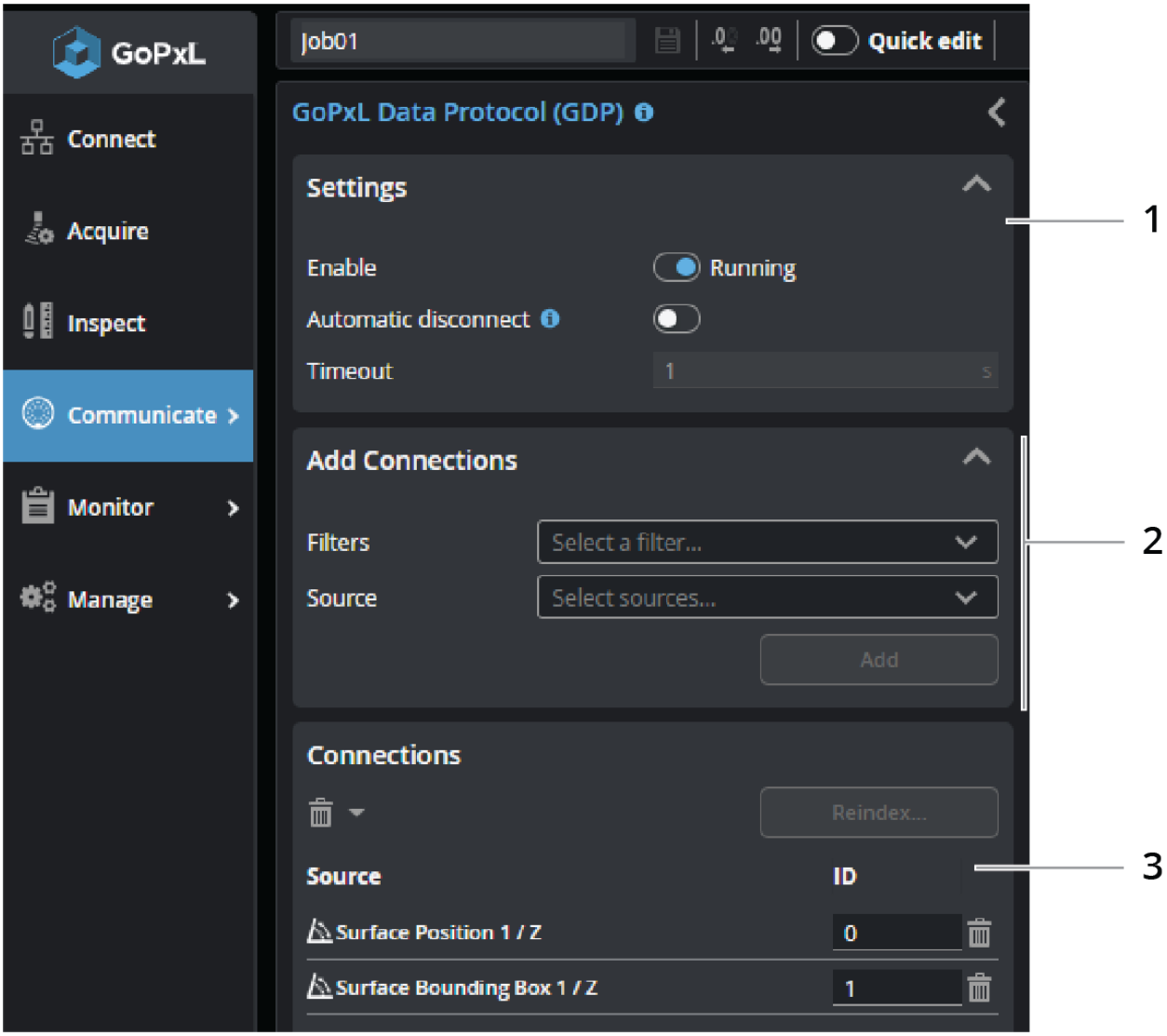
Interface	Description
	supports PROFINET. For more information on configuring the PROFINET protocol in the GoPxL UI, see *PROFINET Protocol* on page 1126. For the PROFINET protocol specification, see *Connections Map* on page 1130.
Digital	Lets you configure digital outputs. For more information, see *Digital Output* on page 1147.
GoHMI	GoHMI and the GoHMI Designer let you create a custom user interface connected to Gocator outputs that users can access from a browser on a PC, a touchscreen, or a mobile device. For more information, see *GoHMI and GoHMI Designer* on page 1210.
Data Storage	Lets you configure storing acquired data to a remote FTP server or to local storage. Note that sensors can only store data on a remote server. For more information, see .

GoPxL Data Protocol

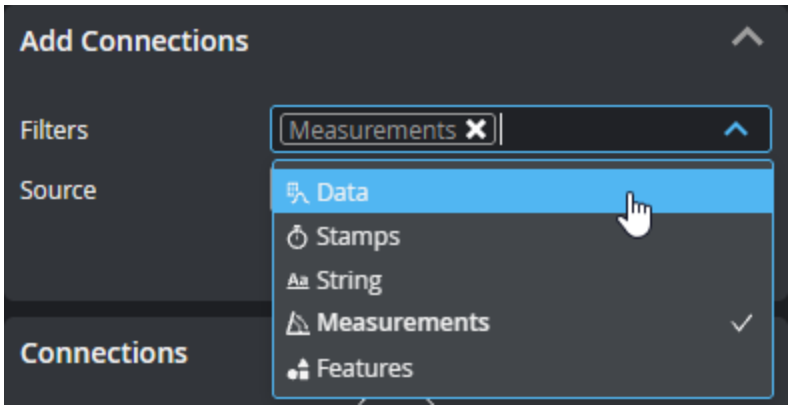
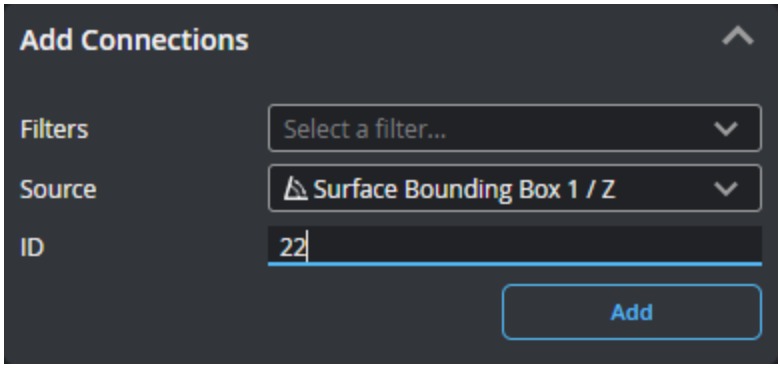
> If you are using a GoMax Smart Vision Accelerator, only connect Ethernet-based PLCs or PCs to the ETH1 port on the GoMax device (whether via a switch or not).

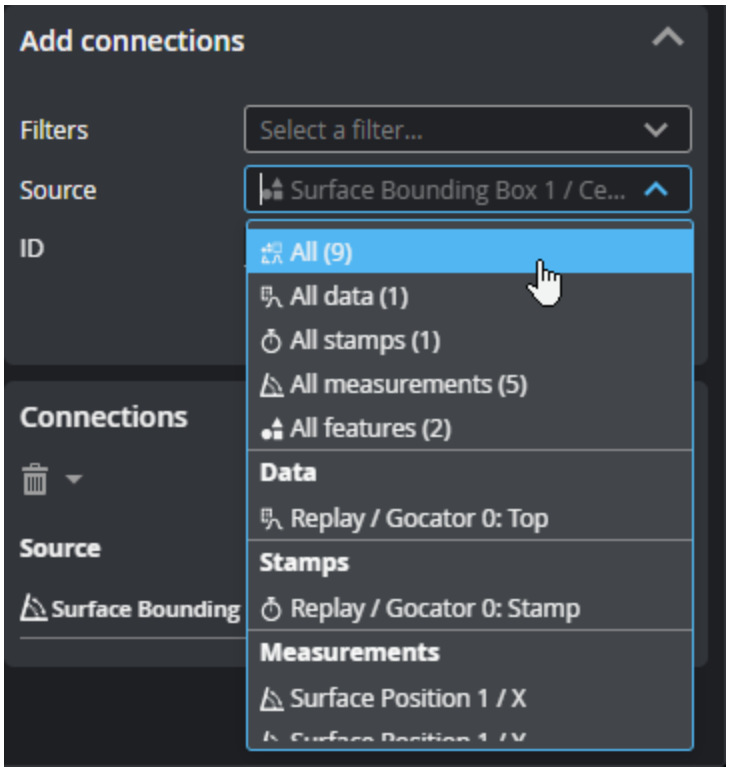
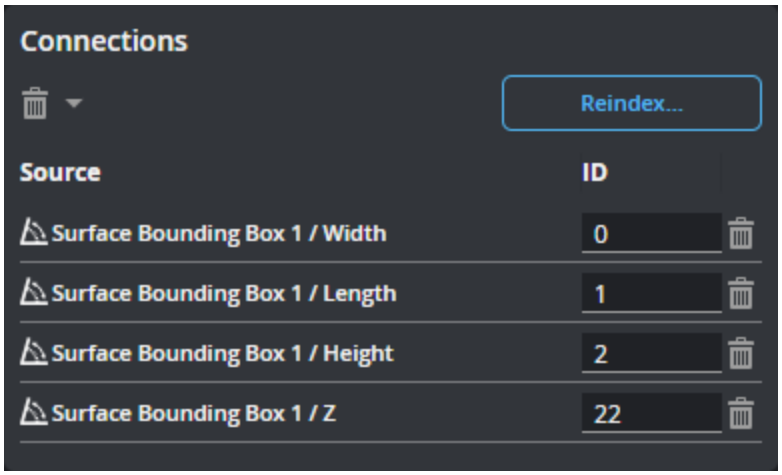
The GoPxL Data Protocol sends TCP messages containing data and measurement results to client applications that use the GoPxL SDK. You can select which measurements and what type of scan data to send (profile or surface, image, or intensity). Note that GDP does not have to be enabled for the client application to read and write configurations, only to receive data and measurements.

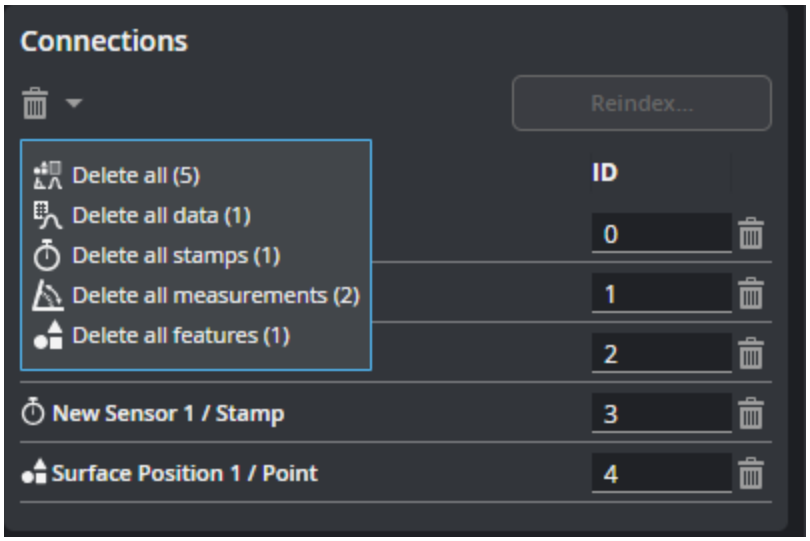
For information on the GoPxL SDK and the REST API, see *GoPxL SDK and REST API* on page 1234.



	Element	Description
1	Protocol settings	Lets you disable or configure the protocol. For information on the settings available here, see *Configuration area parameters* on page 1026.

	Element	Description
2	Add Connections	Lets you add sources (data, stamp, measurement, strings, or geometric feature) to send over the protocol. Added connections appear in the **Connections** section at the bottom of the panel If you have many data types in your enabled outputs, you can filter them by type to make it easier to find the source you want in the **Source** drop-down. You can enable more than one filter.  Choose a source in **Source**. If you leave ID as "Auto" GoPxL automatically increments the ID; otherwise, you can set it yourself. The field is hidden if more than one source is selected in **Source**.  You can also add all sources, or all sources of a given type. Adding multiple sources automatically increments the ID

	Element	Description
		 Click **Add** to add the source to the Connections table below.
3	Connections	The connections listed here will be sent by the GoPxL Data Protocol to an application developed using the GoPxL SDK. You can delete a connection using the trashcan icon in its row or change its ID. If you set an ID to the ID of another source, all other source automatically shift up to make room for it.  You can also use shortcut commands to delete all of the connections, or specific categories of connections.

Element	Description
	 If you set an ID to the index of another tool, all sources following the changed index are incremented. Click the **Reindex** button to reorder the connections and fill any gaps between index numbers.

Configuration area parameters

Parameter	Description
Enable	Toggles whether the Gocator communication protocol is enabled.
Automatic disconnect	When enabled, automatically disconnects from the client after the timeout value in **Timeout** if the sensor is unable to send data.
Timeout	Sets the timeout value when **Automatic disconnect** is enabled.

Modbus TCP Protocol

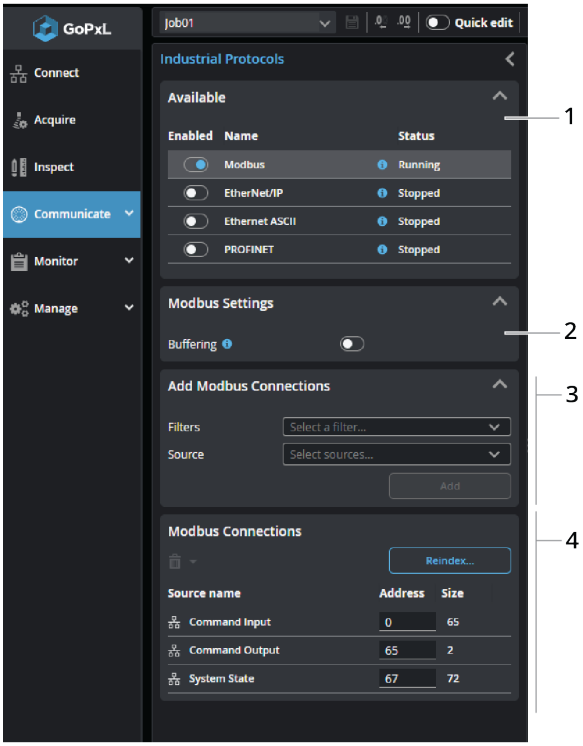
You can use a Modbus TCP programmable logic controller (PLC) over Ethernet to operate a sensor and receive values, such as measurements, stamps, or strings. Modbus only supports a subset of the tasks that can be accomplished in the web interface (for example, starting and stopping sensors, and switching jobs). Scan data can't be sent to the PLC.

You can send data from a PLC to GoPxL after adding and configuring an instance of the User Data Input tool; for more information on the tool, see *User Data Input Tool* on page 1015.

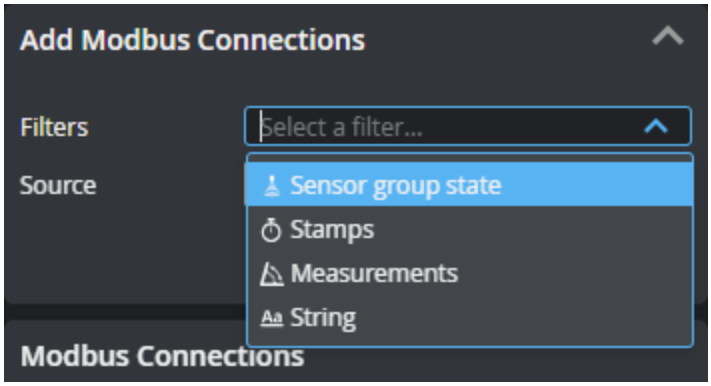
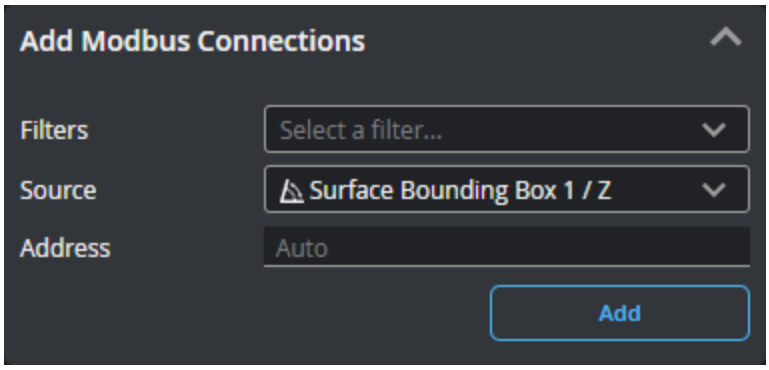
To send data over Modbus to the User Data Input tool, use command ID 9 and populate the Control Input block's Command Arguments field with the required data; a success/fail is returned. For more information, see the following: *Control Input* on page 1033.

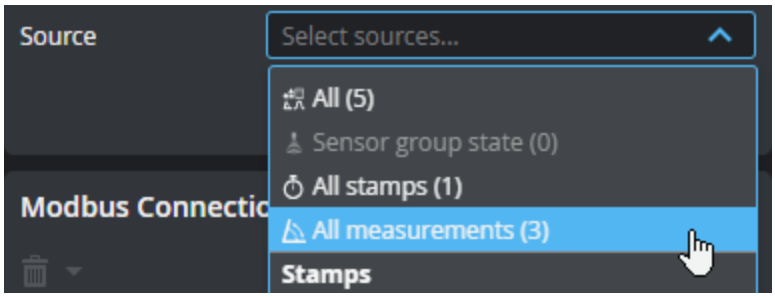
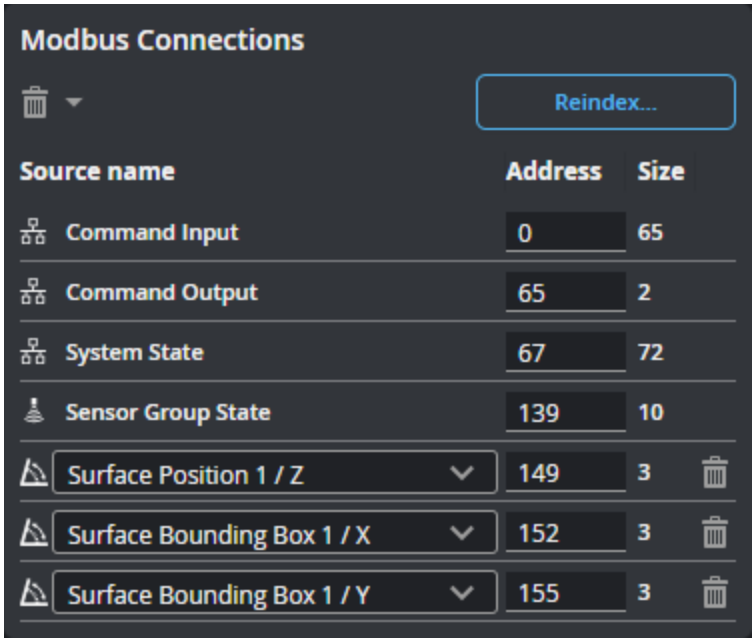
> If you are using a GoMax Smart Vision Accelerator, only connect Ethernet-based PLCs or PCs to the ETH1 port on the GoMax device (whether via a switch or not).

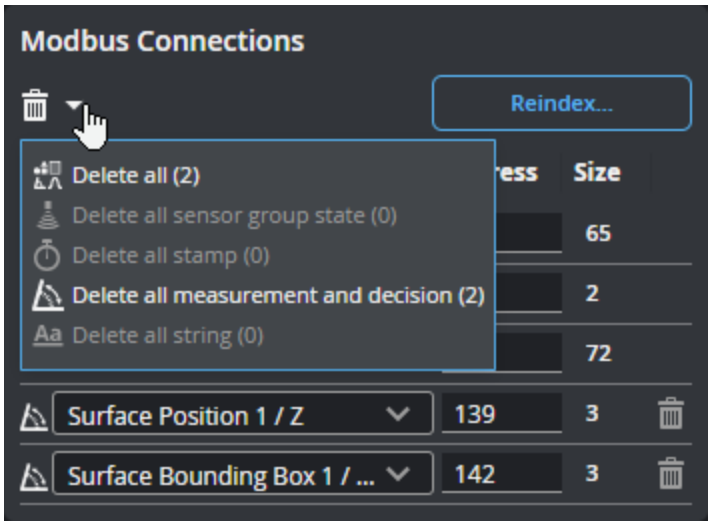
You configure the protocol in GoPxL on the **Communicate** > **Industrial** page. For information on the protocol's format and configuring your PLC, see the following: *Modbus TCP Protocol Format - PLC Information* on page 1031.



	Element	Description
1	Available Protocols	Click a row in the table to choose which protocol to configure. Enable a protocol by toggling the slider to the right.
2	Modbus Settings	**Buffering**

	Element	Description
		Toggle the slider to enable/disable buffering of scan outputs. Buffering should be enabled if multiple objects may be detected within a time frame shorter than the polling rate of the PLC. If buffering is enabled, the PLC must send the 'Buffer Advance' command to advance the queue before reading the measurement results. For more information, see the following: *Control Input* on page 1033.
3	Add Modbus Connections	Lets you add sources (such as stamps, measurements, or strings) to send over the protocol. Added connections appear in the **Modbus Connections** section at the bottom of the panel. If you have many sources, you can filter them by type to make it easier to find the source you want in the **Source** drop-down . You can enable more than one filter.  Choose a source in the **Source** drop-down. If you leave **Address** as "Auto", GoPxL automatically increments the value; otherwise, you can set it yourself.  You can also add all sources, or all sources of a given type.

	Element	Description
		 Click **Add** to add the source to the Modbus Connections table below.
4	Modbus Connections	The connections listed here will be sent by the protocol to a compatible PLC. You can delete a connection using the trashcan icon in its row, or change its ID. When changing an ID, make sure to use the size of the source to correctly calculate the minimum offset.  For string sources, you must specify a maximum expected length in the source's **Size** field. You can also use shortcut commands to delete all of the connections, or specific categories of connections.

Element	Description
	 Use **Reindex...** to remove gaps of unused addresses between connections. You can also use **Reindex...** to set the value of the first index.

Modbus TCP Protocol Format - PLC Information

GoPxL supports the Modbus TCP communication to provide system, sensor group, and scan output data to a PLC, as well as accept incoming commands. After enabling Modbus in the GoPxL web interface (for more information, see *Modbus TCP Protocol* on page 1027), the sensor starts listening on port 502 for one or more Modbus clients. GoPxL operates as a Modbus server (slave), and a PLC that connects to it acts as the Modbus client (master). All communication is initiated by the PLC, with GoPxL responding to the Modbus requests.

> The connection mappings for the industrial protocols (PROFINET, EtherNet/IP, and Modbus) have changed to correct a byte alignment issue. To allow communication between PLCs and GoPxL 1.2, you may need to update PLC logic created for GoPxL 1.1.
>
> For EtherNet/IP, the EDS file has been updated to reflect the changes.

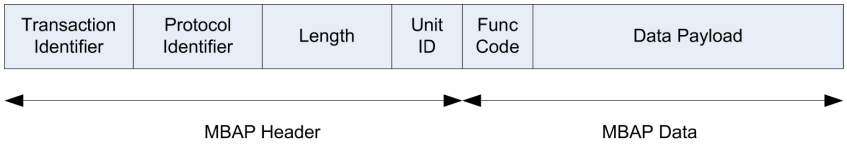
While the Modbus protocol includes several types of requests, the two basic operations are reading and writing registers. Commands are invoked by the PLC sending write-register requests. Data is requested by the PLC by sending read-register requests.

There are 16 bits per register in the Modbus protocol. 32-bit and 64-bit values are stored in consecutive registers in Big-Endian format.

If buffering is enabled with the Modbus protocol (when part detection is used and if multiple objects may be detected within a time frame shorter than the polling rate of the PLC), the PLC must send the Buffer Advance command to advance the queue before reading the measurement results (see *Control Input* on page 1033). For more information on part detection, see *Profile Part Detection* on page 603.

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

Field	Length (Bytes)	Description
		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

Connections Map

The Modbus protocol supports various blocks types. Each block type can be either an input block or an output block. Input blocks contain write-only registers and are used by the Modbus client to send commands such as starting or stopping a scan. Output blocks contain read-only registers and are used to store various types information, such as control, system, and sensor group state data as well as scan output data.

You can send data from a PLC to GoPxL after adding and configuring an instance of the User Data Input tool; for more information on the tool, see *User Data Input Tool* on page 1015.

To send data over Modbus to the User Data Input tool, use command ID 9 and populate the Control Input block's Command Arguments field with the required data; a success/fail is returned. For more information, see the following: *Control Input* below.

> The Modbus implementation uses the term "scanner" instead "sensor group." The two are interchangeable.

The starting address location of each block within the Modbus register map may be assigned using the GoPxL web interface.

The register address of a specific field within a block is equal to the block's address plus the field's address offset. For example, the command identifier in the Control Input block has an address offset equal to 1. If the Control Input block is located at register address 100, the command identifier value is located at register address 101.

SUPPORTED BLOCK TYPES

The block types supported by the Modbus protocol are listed in the table below. The size field indicates the number of registers that each block type requires.

Supported Block Types

Supported Block Type	Read or Write only?	Size (number of Modbus registers)	Description
"ControlInput"	WO	65	Command message set by the Modbus client.
"ControlOutput"	RO	2	Result data corresponding to a command that was written to the Control Input block.
"Measurement"	RO	3	Measurement data of a scan output measurement data message.
"Scanner"	RO	10	Information about a specific sensor group.
"Stamp"	RO	17	Stamp message information.
"String"	RO	4 (by default)	String data of a scan output string data message. Size can be configured through the GoPxL web interface.
"System"	RO	72	Information about the LMI device.

CONTROL INPUT

The input of the Control Input block contains the fields described below:

Control Input Fields

Field	Type	Address offset	Description
Command Sequence Number	16u	0	The command sequence number is set by the external client to uniquely identify a command request. This number

Field	Type	Address offset	Description
			is sent back in the Control Output. This is an optional field but is recommended that Modbus clients use this to help the clients correlate the Command Status in the Control Output with a specific command request in Control Input.
Command Identifier	16u	1	Identifies the requested action from the external client. Supported actions: 0 = No Command 1 = Start Scanning 2 = Stop Scanning 3 = Align Sensor Group 4 = Clear Alignment 5 = Load Job File 6 = Software Trigger 7 = Buffer Advance 9 = Set User Data Input. For use with the User Data Input tool; for more information see *User Data Input Tool* on page 1015.
Command Arguments	16u	2	Arguments for commands that require arguments. **Load Job File**: The argument contains the *NULL terminated* job file name. The job file name must *not* include the job file extension. The job extension is automatically appended by the Modbus server. Each register holds a single character; maximum 63 characters including the null terminator. **Set User Data Input**: Use up to the full 63 registers available in the block (a 126-byte payload) for data input.

CONTROL OUTPUT

The output of the Control Output block contains the fields described below:

Control Output Fields

Field	Type	Address offset	Description
Command Sequence Number	16u	0	The command sequence number received from the client in the Control Input.
Command Status	16u	1	Status of the command, referenced by the Command Sequence Number. 0 = No Command Received 1 = In Progress

Field	Type	Address offset	Description
			2 = Success 3 = Fail This field takes up 1 register.

SYSTEM STATE

The output of the System State block contains the fields described below:

System State Fields

Field	Type	Address offset	Description
System Uptime	64u	0	Amount of time the application has been running, in seconds.
System State	16u	4	Whether system is running or not. 0 = System is stopped 1 = System is running
Current Job File Name Length	16u	5	Number of characters in the current job file name.
Current Job File Name	16u	6	Name of the job file currently loaded into the system, without the extension. Maximum length of job file name is 64 characters. Each register contains a single character. Job file name is *not* null terminated.
Buffer Count	16u	70	The current amount of buffered scan output data when buffering is enabled.
Buffer Overflow	16u	71	A value of 0 indicates that no overflow has occurred. A value of 1 indicates that overflow occurred and data is being lost. This field takes up 1 register.

SENSOR GROUP STATE

The output of the Sensor Group State block contains the fields described below:

Sensor Group State Fields

Field	Type	Address offset	Description
Current Encoder Position	64s	0	The current encoder position.
Current Sync Time	64u	4	Current time of the clock used to stamp all scan output messages, in microseconds.
Alignment State	16u	8	Indicates whether all devices under the sensor group are aligned (that is, devices in the sensor group all have a non-

Field	Type	Address offset	Description
			identity transform matrix). 0 = Not Aligned 1 = Aligned 2 = Aligning
Laser Enabled	16u	9	Whether one or more devices under a scanner has its laser enabled to be able to turn on. 0 = All devices' lasers are disabled and cannot turn on 1 = One or more devices' lasers can be turned on This field takes up 1 register.

STAMP

The output of the Stamp block contains the fields described below:

Stamp Fields

Field	Type	Address offset	Description
Z Position	64s	0	Encoder position at time of last index pulse of last frame.
Encoder Position	64s	4	Encoder position of last frame when the image data was scanned/taken.
Time	64u	8	The timetick of the last frame. Timetick is a 64-bit counter that increases with time, so each scan from the same device has a unique timetick value. On Gocator 3D sensors, to convert this value to microseconds, use the following: `time (µs) = timetick / 1.024.`
Frame Index	64u	12	The frame number of the last frame. This field takes up 4 registers.
Digital Input States	16u	16	Digital input states of last frame. Bit 0: State of digital input 1 on master or sensor Bit 1: State of digital input 2 on master or sensor

MEASUREMENT

The output of the Measurement Block contains the fields described below:

Measurement Fields

Field	Type	Address offset	Description
Value	32s	0	Measurement value. This value is the actual measurement value multiplied by 1000. If the measurement value is invalid, the first register of the measurement value is set to 0x80000000.

Field	Type	Address offset	Description
			This means the first register of the measurement value is set to 0x8000 and the second register value is 0x0000.
Decision	16u	2	Measurement decision. Bit 0: 0 = Fail 1 = Pass Bit 1: 0 = Decision value is valid 1 = Decision value is invalid This field takes up 1 register.

STRING

The output of the String Block contains the fields described below:

Measurement Fields

Field	Type	Address offset	Description
String	string	0	String value. Unused space will be set to null (0x0). The string is otherwise not null terminated. If invalid, the block's buffer is entirely set to null, equivalent to an empty string. The string will be truncated if the block's size is smaller than the length of the string.

EtherNet/IP Protocol

You can use an EtherNet/IP programmable logic controller (PLC) over Ethernet to operate a sensor and receive values, such as measurements, stamps, or strings. EtherNet/IP only supports a subset of the tasks that can be accomplished in the web interface (for example, starting and stopping sensors, and switching jobs). Scan data can't be sent to the PLC.

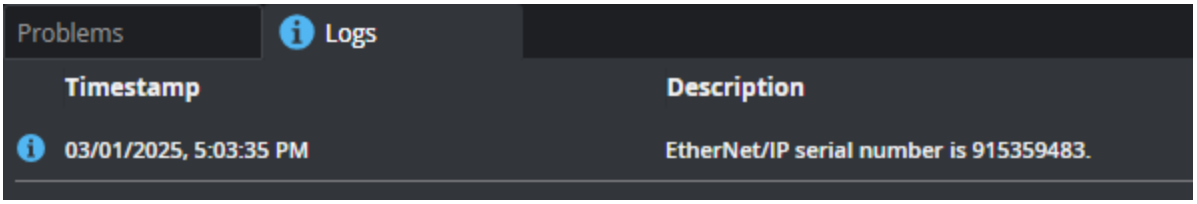
You can send data from a PLC to GoPxL after adding and configuring an instance of the User Data Input tool; for more information on the tool, see *User Data Input Tool* on page 1015.

To send data over EtherNet/IP to the User Data Input tool using Explicit messaging, you must use a specific assembly; see the following: *Explicit Messaging User Data Input* on page 1055.

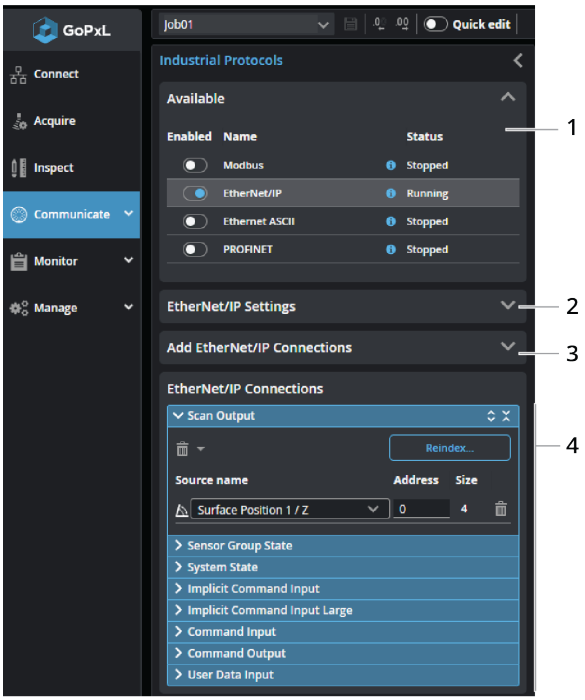
For Implicit messaging, you must use command 9 with one of the two Input objects; see the following: *Implicit Messaging Input* on page 1056 or *Implicit Messaging Input Large* on page 1057.

> If you are using a GoMax Smart Vision Accelerator, only connect Ethernet-based PLCs or PCs to the ETH1 port on the GoMax device (whether via a switch or not).

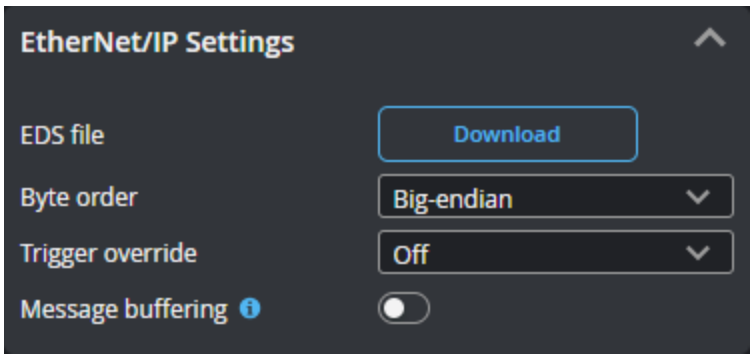
When you enable the protocol, a log entry displays the EtherNet/IP serial number you should use in your PLC.



You configure the protocol in GoPxL on the **Communicate** > **Industrial** page. For information on the protocol's format and configuring your PLC, see the following: *EtherNet/IP Protocol Format - PLC Information* on page 1045.



	Element	Description
1	Available Protocols	Click a row in the table to choose which protocol to configure. Enable a protocol by toggling the slider to the right.
2	EtherNet/IP Settings	You configure the protocol's settings in this section.



EDS file

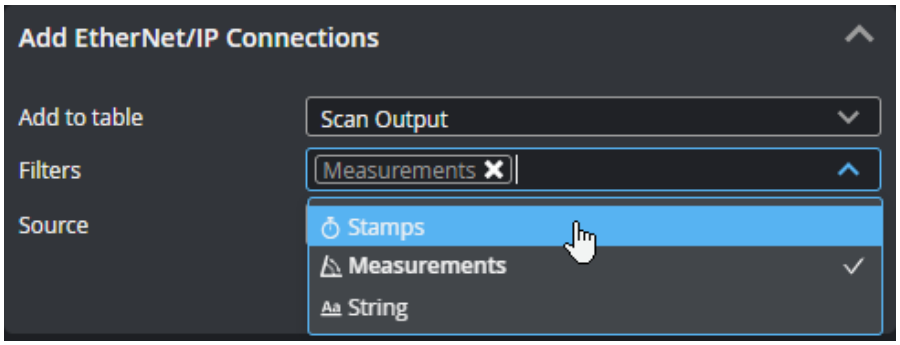
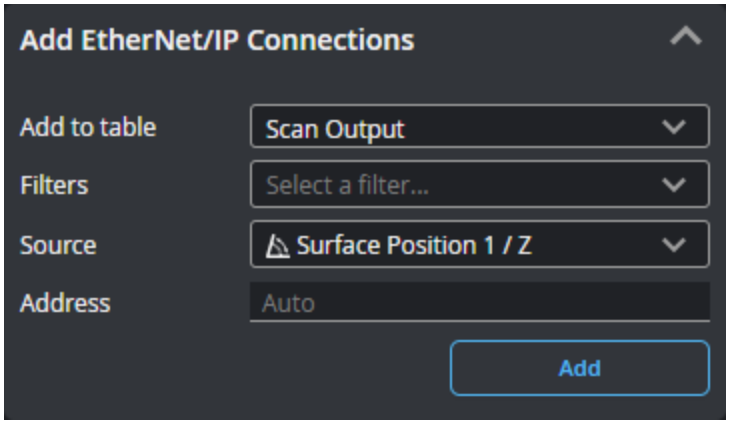
Click **Download** to save an EDS file to the client computer for later installation on the PLC.

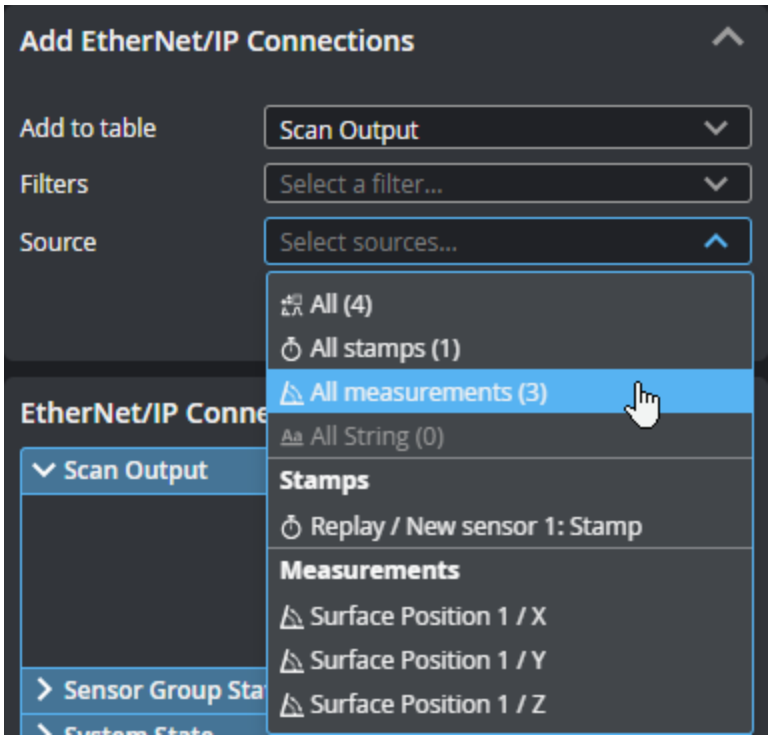
Byte order

One of the following:

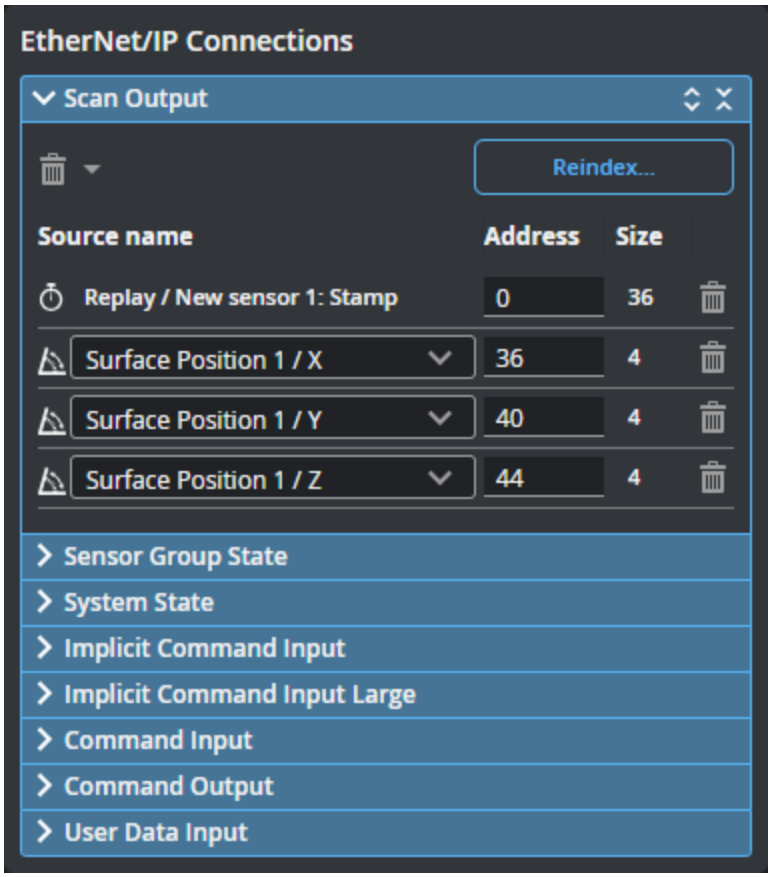
Big-endian: The most significant byte (the "big end") of the data is placed at the byte with the lowest address. The rest of the data is placed in order of decreasing significance in the next three bytes of memory.

	Element	Description
		Little-endian (default): 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. **Trigger override** One of the following: **Off**, **Cyclic**, and **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. **Message buffering** Used to enable and disable buffering of scan outputs. You should enable buffering if you are using part detection and if multiple objects may be detected within a period 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; for more information, see the following: *EtherNet/IP Protocol Format - PLC Information* on page 1045. Message buffering only applies to tool outputs sent over explicit messaging.
3	Add EtherNet/IP Connections	Lets you add sources (such as stamps, measurements, or strings) to send over the protocol. Added connections appear in the **EtherNet/IP Connections** section at the bottom of the panel. When GoPxL is running on a standalone sensor (one not controlling others), **Add to Table** only contains one value (Scan Output) that represents a register assembly. When GoPxL is running on a sensor controlling other sensors, or on a PC or GoMax, **Add to Table** contains an additional entry (Sensor Group State). Use this to choose which register a source is added to. If you have many sources, you can filter them by type to make it easier to find the source you want in the **Source** drop-down . You can enable more than one filter.

Element	Description
	 Choose the source in the **Source** drop-down. If you leave **Address** as "Auto", GoPxL automatically increments the value; otherwise, you can set it yourself. By default, connections are added directly after the last source in the connections map. You can optionally set the address at which a source is added using the **Address** parameter.  You can also add all sources, or all sources of a given type.

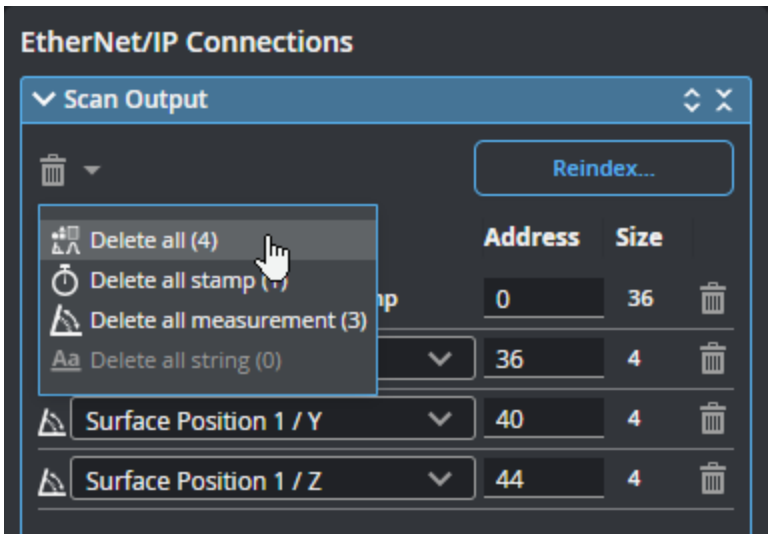
	Element	Description
		 Click **Add** to add the source to the EtherNet/IP Connections table below.
4	EtherNet/IP Connections	The connections listed here will be sent by the protocol to a compatible PLC. You can delete a connection using the trashcan icon in its row, or change its ID. When changing an ID, make sure to use the size of the source to correctly calculate the minimum offset.

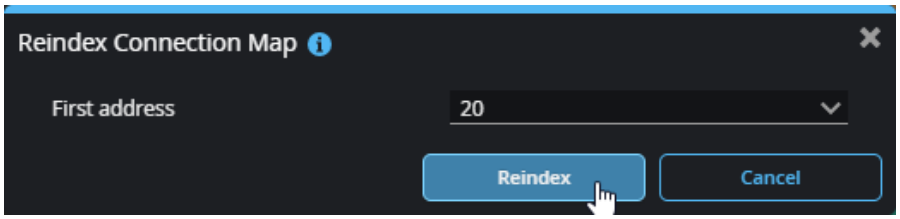
Element	Description



For string sources, you must specify a maximum expected length in the source's **Size** field.

You can also use shortcut commands to delete all of the connections, or specific categories of connections.



Element	Description
	Use **Reindex...** to remove gaps of unused addresses between connections. You can also set the value of the first index. 

EtherNet/IP Protocol Format - PLC Information

EtherNet/IP is an industrial protocol that allows bidirectional data transfer with PLCs that support the protocol. It encapsulates the object-oriented Common Industrial Protocol (CIP). EtherNet/IP communication enables the client to:

- Start and stop sensors
- Load job files
- Align sensors
- Receive sensor states, stamps, strings, and measurement results
- Retrieve data sent from a PLC to the User Data Input tool; for more information on this tool, see *User Data Input Tool* on page 1015

This section describes the EtherNet/IP message and data formats. The commands described in the sections below are those specific to the Gocator protocol and not the complete EtherNet/IP reference command set.

Before configuring your PLC, make sure to enable and configure EtherNet/IP in GoPxL. For more information on this, see *EtherNet/IP Protocol* on page 1038.



Implicit messaging is always enabled.

GoPxL supports EtherNet/IP communication to provide system state, sensor group state, and scan output data to a PLC, and also receive commands. After you enable EtherNet/IP in the GoPxL web interface, the sensor starts listening on the standard EtherNet/IP port 44818 for one or more EtherNet/IP clients. GoPxL operates as an EtherNet/IP server, and a PLC that connects to it acts as the EtherNet/IP client. All communication is initiated by the PLC, with GoPxL responding to the EtherNet/IP requests.

While the EtherNet/IP protocol includes several types of requests, the two core operations are reading and writing EtherNet/IP objects. Commands are invoked by the PLC sending *set* requests. Data is requested by the PLC by sending *get* requests for EtherNet/IP objects.

EtherNet/IP supports both explicit messaging and implicit messaging. Data can be read in Big-Endian or Little-Endian format based on the protocol's configuration selected in the web interface.

Supported EtherNet/IP Objects

All mandatory EtherNet/IP objects required for explicit messaging, including Assembly objects, are supported by the sensors. Except for mandatory objects (see below), the remaining objects are mapped to register assemblies.

Mandatory objects (the Identity Object Class 0x01, TCP/IP Object Class 0xF5, and Ethernet Link Object Class 0xF6) are always present in the EtherNet/IP service, regardless of the configured measurement object.

Implicit messaging supports only Assembly objects.

Object and Register Assembly Mapping

Object and Register Assembly Mapping

Object name	Register assembly	Supported Block Type Internal Identifier	Description	Service	Messaging Type	Instance ID	Max Size (bytes)
Explicit Messaging Command Input *EtherNet/IP Protocol* on page 1038	Control Input	ControlInput	Receives commands from the PLC.	Explicit	SetAttributeSingle (0x10)	112 (0x70) / 117 (0x75)	67 / 468
Explicit Messaging Command Output	Control Output	ControlOutput	Sends command response or feedback to the PLC.	Explicit	GetAttributeSingle (0x0E)	113 (0x71)	3
Sensor Group State	Sensor Group State	Scanner	Provides status of the sensor group.	Explicit	GetAttributeSingle (0x0E)	114 (0x72)	18
System State	System State	System	Reports overall system health and connection info.	Explicit	GetAttributeSingle (0x0E)	115 (0x73)	76
Sample State	Sample State	MeasurementValue String Stamp SystemStatus	Returns latest measurement,string,system status results and stamp.	Explicit	GetAttributeSingle (0x0E)	116 (0x74)	376
Implicit Messaging Input	Implicit Command Input	ControlInput	Receives control commands via cyclic connection.	Implicit	SetAttributeSingle (0x10)	100 (0x64)	67
Implicit Messaging Input Large	Implicit Command Input Large	ControlInputLarge	Receives control commands via cyclic connection. Provides additional space for use with the User Data Input tool. For information on the User Data Input tool, see *User Data Input Tool* on page 1015.	Implicit	SetAttributeSingle (0x10)	102 (0x66)	496

Object name	Register assembly	Supported Block Type Internal Identifier	Description	Service	Messaging Type	Instance ID	Max Size (bytes)
Implicit Messaging Output	Sample State	MeasurementValue String Stamp SystemStatus	Sends measurement, string, system status results and stamp to PLC on cycle.	Implicit	GetAttributeSingle (0x0E)	101 (0x65)	376
Config Assembly	N/A	N/A	Configuration Assembly for Implicit I/O Connection Heartbeat	Implicit	N/A	151 (0x97)	0

[1] For more information on the supported block types, see below.

Supported Block Types

The block types supported by the EtherNet/IP protocol are listed in the table below. The size field indicates the number of registers that each block type requires.

Supported Block Types

Block Type Internal Identifier	Read or Write only?	Size (number of EtherNet/IP registers)	Description
ControlInput	Write-Only	67	Accepts control commands from the EtherNet/IP client.
ControlInputLarge	Write-Only	496	Accepts control commands and user data from the EtherNet/IP client. For information on the User Data Input tool, see *User Data Input Tool* on page 1015.
ControlOutput	Read-Only	3	The result of the last command written to the ControlInput block.
MeasurementValue	Read-Only	4	Measurement values.
String	Read-Only	8 (by default)	String values. Size can be configured through the GoPxL web interface.
Scanner	Read-Only	18	The current state and configuration of the sensor group.
Stamp	Read-Only	36	Stamp message information.
System	Read-Only	76	Overall device state and configuration of the GoPxL system.
SystemStatus	Read-Only	16	System status and command status GoPxL system.

Identity Object (Class 0x01)

The Identity Object (Class 0x01, Instance ID 0x01) provides read-only device identification attributes used by EtherNet/IP clients during explicit messaging.

Identity object (Class 0X01)

Attribute ID	Attribute Name	Data Type	Value	Description	Access
1	Vendor ID	UINT	1256	ODVA-assigned vendor ID for LMI	Get
2	Device Type	UINT	43	Device type	Get
3	Product Code	UINT	1	Product code	Get
4	Revision	Structure (USINT, USINT)	Major: 2, Minor: 1	Firmware revision	Get
6	Serial number	UDINT	[device-specific]	Sensor serial number	Get
7	Product Name	SHORT_STRING[32]	Gocator	Gocator product name	Get

TCP/IP Object (Class 0xF5)

The TCP/IP Object (Instance ID 0x01) provides read-only attributes related to the network configuration of the device. This object is used by EtherNet/IP clients to retrieve information such as the current IP address and interface status.



GoPxL does not support writing or modifying TCP/IP settings via EtherNet/IP. Network parameters must be configured through the GoPxL web interface or external methods such as DHCP.

For complete structural definitions, see CIP Specification Volume 2, Chapter 5-3.

Attribute ID	Attibute Name	Data Type	Value	Description	Access
1	Status	UDINT	0	TCP interface status	Get
2	Configuration Capability	UINT	0	Indicates which configuration methods (e.g., BOOTP, DHCP) are supported.	Get
3	Configuration Control	UINT	0	Indicates which method is currently active.	Get
4	Physical Link Object	Structure	-	Contains: Path Size (UINT), Path (Padded EPATH) See §5.3.3.2.4 CIP V2.	Get
5	Interface Configuration	Structure	-	Contains: IP Address, Subnet Mask, Gateway, DNS, Domain See §5.3.3.2.5.	Get

Ethernet Link Object (Class 0xF6)

The Ethernet Link Object (Instance ID 0x01) exposes read-only attributes related to the physical Ethernet interface, such as link speed, status, and MAC address. These attributes provide diagnostic and identification information for EtherNet/IP clients.

For full attribute definitions and behavior, see CIP Specification Volume 2, Chapter 5-4.

Attribute ID	Attribute Name	Data Type	Value	Description	Access
1	Interface Speed	UDINT	1000	Data rate of the Ethernet interface, in Mbps.	Get
2	Interface Flags	UDINT	[bitfield]	Bit-level status flags: Bit 0 – Link Status (0 = Inactive, 1 = Active) Bit 1 – Duplex (0 = Half, 1 = Full)	Get
3	Physical Address	Array[6] OF USINT	e.g., 00 16 20 00 2E 42	Device MAC address.	Get

Assembly Object (Class 0x04)

All EtherNet/IP assembly objects in GoPxL are implemented using Class `0x04` and Attribute `0x03`, with each object distinguished by a unique Instance ID. Each instance represents a specific data exchange structure, either for input (write-only) or output (read-only).

A register assembly in GoPxL is a container that holds one or more register blocks. There is a strict one-to-one mapping between EtherNet/IP assembly objects and GoPxL register assemblies. This mapping allows PLCs to interact with specific control, system, or measurement data structures through EtherNet/IP.

Assembly objects are always exposed in the GoPxL EtherNet/IP interface, whether used in explicit or implicit messaging. However, data will only be populated if the corresponding register assembly is actively selected and configured in the GoPxL web interface. If the register assembly is not enabled, the object remains accessible but contains no meaningful data.

For Explicit Messaging, each assembly is read or written by the PLC using `GetAttributeSingle` or `SetAttributeSingle` services on demand.

For Implicit Messaging, each assembly is mapped to a cyclic connection. Once a connection is established by the PLC, data is exchanged automatically at the configured RPI (Requested Packet Interval).

Explicit Messaging Command Input

The Command Input block is used by an EtherNet/IP client to send control instructions to GoPxL. It maps to the `ControlInput` register assembly and uses Class `0x04`, Attribute `0x03`, Instance ID 112 (`0x70`).

This assembly supports the `SetAttributeSingle (0x10)` service and has a maximum size of 67 bytes.

Information	Value
Register Assembly	Control Input register assembly
Supported Block Type Identifier	ControlInput
Class	0x04
Instance	112 (0x70)
Attribute Number	3
Max Size	67 bytes

INPUT STRUCTURE

The input data block consists of the following fields:

Attribute	Name	Type	Value
Command Sequence Number	UINT16	0	Optional. A unique value set by the client to identify each command. Echoed back in the Control Output.
Command Identifier	UINT8	2	Identifies the action to perform: 0 = No Command 1 = Start Scanning 2 = Stop Scanning 3 = Align Sensor Group 4 = Clear Alignment 5 = Load Job File 6 = Software Trigger 9 = Set User Data Input
Unused	UINT8	3	Reserved unused field.
Command Arguments	ARRAY [UINT8] (max 63)	4	Optional. **Load Job File** The argument must be a null-terminated string containing the job filename. Do not include the file extension, as the control service server appends it automatically. Each register holds a single character, with a limit of 63 characters including the null terminator. **Set User Data Input** Use up to the full 63 registers for data input.

Explicit Messaging Command Output

The Command Output block provides the result of the last command sent to the device via the Control Input block. It maps to the `ControlOutput` register assembly and uses Class `0x04`, Attribute `0x03`, Instance ID 113 (`0x71`).

This assembly supports the `SetAttributeSingle (0x0E)` service and has a maximum size of 3 bytes.

Information	Value
Register Assembly	Control Output register assembly
Supported Block Type Identifier	ControlInput
Class	0x04
Instance	113 (0x71)
Attribute Number	3
Max Size of Register Assembly	3 bytes

OUTPUT STRUCTURE

The fields below are fixed in size and structure:

Attribute	Name	Type	Value
Command Sequence Number	UINT16	0	The sequence number echoed from the most recent Control Input.
Command Status	UINT8	2	Status of the referenced command: 0 = No Command Received 1 = In Progress 2 = Success 3 = Failure

Explicit Messaging Sensor Group State

The Sensor Group State block provides real-time status data about the active sensor group (scanner). It maps to the `Scanner` register assembly and uses Class `0x04`, Attribute `0x03`, Instance ID 114 (`0x72`).

This assembly supports the `GetAttributeSingle (0x0E)` service and has a maximum size of 18 bytes.

Information	Value
Register Assembly	Sensor Group State register assembly
Supported Block Type Identifier	Scanner
Class	0x04
Instance	114 (0x72)

Information	Value
Attribute Number	3
Max Size	18 bytes

OUTPUT STRUCTURE

The output of the Sensor Group Block contains the following fields:

Attribute	Name	Type	Value
Current Encoder Position	INT64	0	The current position from the encoder, used to synchronize with motion systems.
Current Sync Time	UINT64	8	Current system time used to timestamp scan outputs, in microseconds.
Alignment State	UINT8	16	Indicates sensor group alignment: 0 = Not Aligned 1 = Aligned 2 = Aligning
Laser Enabled	UINT8	17	Indicates laser readiness: 0 = All lasers disabled 1 = One or more lasers enabled

Explicit Messaging System State

The System State block reports the current runtime status of the GoPxL system. It maps to the `System State` register assembly and uses Class `0x04`, Attribute `0x03`, Instance ID 115 (`0x73`).

This assembly supports the `GetAttributeSingle (0x0E)` service and has a maximum size of 76 bytes.

Information	Value
Register Assembly	System State register assembly
Supported Block Type Identifier	System
Class	0x04
Instance	115 (0x73)
Attribute Number	3
Max Size	76 bytes
Supported Service	GetAttributeSingle (0x0E)

OUTPUT STRUCTURE

This block contains system-level runtime information and file state useful for diagnostics and monitoring:

Field	Type	Offset (bytes)	Description
System Uptime	UINT64	0	Number of seconds the GoPxL application has been running. On sensor or GoMax, this is the amount of time the GoPxL software on the device has been running. This will be close to the amount of time the sensor or GoMax has been powered on. On PC, this is the amount of time the GoPxLService instance has been running on the PC. This can be very different from the amount of time the PC has been powered on.
System State	UINT8	8	Indicates if the system is actively running: 0 = Stopped 1 = Running
Buffer Count	UINT8	9	Number of buffered scan output messages (when buffering is enabled).
Buffer Overflow	UINT8	10	0 = No overflow 1 = Overflow has occurred, indicating data is being lost.
Job File Name Length	UINT8	11	Number of characters in the loaded job file name.
Job File Name	ARRAY [UINT8] (max 64)	12	Name of the loaded job file, without the extension. Not null-terminated. Each register stores one character.

Explicit Messaging Sample State

The Sample State block reports metadata and measurement data related to the most recent scan frame. It maps to the `Stamp`, `MeasurementValue`, and `String` register assemblies and uses Class `0x04`, Attribute `0x03`, Instance ID 116 (`0x74`).

This assembly supports the `GetAttributeSingle (0x0E)` service and has a maximum size of 375 bytes.

Information	Value
Register Assembly	Sample State register assembly
Supported Block Type Identifier	Stamp, MeasurementValue, String
Class	0x04
Instance	116 (0x74)
Attribute Number	3
Max Size of Register Assembly	375 bytes
Supported Service	GetAttributeSingle (0x0E)

STAMP BLOCK

The Stamp block provides frame-level metadata useful for alignment, timing, and synchronization across devices. Values reflect the most recent frame acquired by the system.

Field	Type	Offset (bytes)	Value
Z Position	INT64	0	Encoder position at time of last index pulse of the last frame.
Encoder Position	INT64	8	Encoder position at time of scan capture.
Time	UINT64	16	The timetick of the last frame. Timetick is a 64-bit counter that increases with time, so each scan from the same device has a unique timetick value. On Gocator 3D sensors, to convert this value to microseconds, use the following: `time (µs) = timetick / 1.024.`
Frame Index	UINT64	24	Frame number of the last scan.
Digital Input States	UINT8	32	Bitfield of digital inputs:Bit 0 = Digital Input 1Bit 1 = Digital Input 2
Buffer (reserved)	UINT8 [3]	33-35	Reserved. Unused padding bytes.

MEASUREMENT VALUE BLOCK

The MeasurementValue block returns the actual scan measurement value. GoPxL 1.2 does not support measurement + decision blocks—only measurement is reported.

Field	Type	Offset (bytes)	Value
[measurement value]	INT32	0	Measurement value ×1000. For example, 12.345 mm is sent as 12345. If the value is invalid, this field is set to 0x80000000.

STRING BLOCK

The String block returns the string scan output.

Field	Type	Offset (bytes)	Value
[string value]	STRING	0	String value. Unused space will be set to null (0x0). The string is otherwise not null terminated. If invalid, the block's buffer is entirely set to null, equivalent to an empty string. The string will be truncated if the block's size is smaller than the length of the string. Block size is 8 bytes by default and can be configured in the GoPxL UI.

SYSTEM STATUS BLOCK

The SystemStatus block provides system level status information also details of last executed command through EtherNet/IP.

Field	Type	Offset (bytes)	Value
System State	UINT8	0	Indicate if the system is in an active run state. 0= Stop 1 = Run
Alignment State	UINT8	1	Indicate sensor group alignment. 0 = Not aligned 1 = Aligned 2 = Align State Active
Buffer (Reserved)	UINT8	2	Reserved for Laser Safety Status.
Buffer (Reserved)	UINT8	3	Reserved. Unused padding byte. Read as 0.
Last Command Sequence Number	UINT16	4	Last command sequence number received through ethernet IP command input block.
Last Command Identifier	UINT16	6	Identifier of most recent command reference by Last command sequence number. This represent only the command received through ethernet IP only.
Command Status	UINT8	8	Status of the reference command: 0 = No Command Received 1 = In Progress 2 = Succes 3 = Failure
Buffer (Reserved)	UINT8	9	Reserved. Unused padding byte. Read as 0.
Error Code	UINT16	10	Failure code for the reference command when Command Status is 3 (Failure). When Command Status is not 3 , this filed is set to 0.
Buffer (Reserved)	UINT8 [4]	12-15	Reserved. Unused padding byte. Read as 0.

Explicit Messaging User Data Input

The User Data Input block lets a PLC exchange a section of memory buffer with the Gocator device. It uses Class `0x04`, Attribute `0x03`, and Instance ID 117 (`0x75`). This assembly supports the `GetAttributeSingle (0x0E)` service and has a maximum size of 468 bytes.

Only one instance of `UserDataInput` can be created and it is automatically created by the system.

Information	Value
Register Assembly	User Data Input register assembly

Information	Value
Supported Block Type Identifier	UserDataInput
Class	0x04
Instance	117 (0x75)
Attribute Number	3
Max Size of Register Assembly	468 bytes
Supported Service	SetAttributeSingle (0x10)

USER DATA INPUT BLOCK

The `UserDataInput` block represents a data buffer of 468 bytes. It allows a PLC user to define and use any sections within this buffer and exchange that with GoPxL.

Field	Type	Offset (bytes)	Value
[Data]	Any	0	A memory region that lets the user define and use it.

Implicit Messaging Input

The Implicit Messaging Input object receives control commands from a PLC. It maps to the `ControlInput` register assembly and is assigned Instance ID 100 (`0x64`)

Information	Valu
Register Assembly	Implicit Command Input register assembly
Supported Block Type Identifier	ControlInput
Class	0x04
Instance	100 (0x64)
Attribute Number	3
Max Size of Register Assembly	67 bytes
Supported Service	SetAttributeSingle (0x10)

INPUT STRUCTURE

The input data block consists of the following fields.

Field	Type	Offset (bytes)	Value
Command Sequence Number	UINT16	0	Optional. A unique value set by the client to identify each command. Echoed back in the Control Output.
Command Identifier	UINT8	2	Identifies the action to perform: 0 = No Command 1 = Start Scanning

Field	Type	Offset (bytes)	Value
			2 = Stop Scanning 3 = Align Sensor Group 4 = Clear Alignment 5 = Load Job File 6 = Software Trigger 9 = Set User Data Input. For use with the User Data Input tool; for more information see *User Data Input Tool* on page 1015.
Unused	UINT8	3	Reserved unused field.
Command Arguments	ARRAY [UINT8] (max 63)	4	Optional. **Load Job File** The argument must be a null-terminated string containing the job filename. Do not include the file extension, as the control service server appends it automatically. Each register holds a single character, with a limit of 63 characters including the null terminator. **Set User Data Input** Use up to the full 63 registers for data input.

Implicit Messaging Input Large

The Implicit Messaging Input Large object receives control commands from a PLC. It maps to the `ControlInputLarge` register assembly and is assigned Instance ID 102 (`0x66`)

The structures of the Implicit Messaging Input Large and Implicit Messaging Input block. The main differences are as follows:

The Implicit Messaging Input block allows up to 63 bytes to be used for a file name with the Load Job File command and user data for the Set User Input Data command.

The Implicit Messaging Input Large block allows up to 63 bytes for a file name with the Load Job File command, or up to 492 bytes for user data with the Set User Input Data command.

Information	Value
Register Assembly	Implicit Command Input register assembly
Supported Block Type Identifier	ControlInputLarge
Class	0x04
Instance	102 (0x66)
Attribute Number	3
Max Size of Register Assembly	496 bytes
Supported Service	SetAttributeSingle (0x10)

INPUT STRUCTURE

The input data block consists of the following fields.

Field	Type	Offset (bytes)	Value
Command Sequence Number	UINT16	0	Optional. A unique value set by the client to identify each command. Echoed back in the Control Output.
Command Identifier	UINT8	2	Identifies the action to perform: 0 = No Command 1 = Start Scanning 2 = Stop Scanning 3 = Align Sensor Group 4 = Clear Alignment 5 = Load Job File 6 = Software Trigger 9 = Set User Data Input. For use with the User Data Input tool; for more information see *User Data Input Tool* on page 1015.
Unused	UINT8	3	Reserved unused field.
Command Arguments	ARRAY [UINT8] (max 492)	4	Optional. **Load Job File** The argument must be a null-terminated string containing the job filename. Do not include the file extension, as the control service server appends it automatically. Each register holds a single character, with a limit of 63 characters including the null terminator. **Set User Data Input** Use up to the full 492 registers for data input.

Implicit Messaging Output

The Implicit Messaging Output object streams live scan data from GoPxL to the PLC. It maps to the `Sample State` register assembly, which includes Stamp, MeasurementValue, and String blocks `0x04`, Attribute `0x03`, Instance ID 101 (`0x65`).

The structures of the Implicit Messaging Input Large and Implicit Messaging Input block. The main differences are as follows:

The Implicit Messaging Input block allows up to 63 bytes to be used for a file name with the Load Job File command and user data for the Set User Input Data command.

The Implicit Messaging Input Large block allows up to 63 bytes for a file name with the Load Job File command, or up to 492 bytes for user data with the Set User Input Data command.

Information	Value
Register Assembly	Sample State register assembly
Supported Block Type Identifier	Stamp, MeasurementValue, String
Class	0x04
Instance	101 (0x65)
Attribute Number	3
Max Size of Register Assembly	376 bytes

STAMP BLOCK

The Stamp block provides frame-level metadata useful for alignment, timing, and synchronization across devices. Values reflect the most recent frame acquired by the system

Field	Type	Offset (bytes)	Value
Z Position	INT64	0	Encoder position at time of last index pulse of the last frame.
Encoder Position	INT64	8	Encoder position at time of scan capture.
Time	UINT64	16	The timetick of the last frame. Timetick is a 64-bit counter that increases with time, so each scan from the same device has a unique timetick value. On Gocator 3D sensors, to convert this value to microseconds, use the following: `time (µs) = timetick / 1.024.`
Frame Index	UINT64	24	Frame number of the last scan.
Digital Input States	UINT8	32	Bitfield of digital inputs:Bit 0 = Digital Input 1Bit 1 = Digital Input 2
Buffer (reserved)	UINT8 [3]	33-35	Reserved. Unused padding bytes.

MEASUREMENT VALUE BLOCK

The MeasurementValue block returns the actual scan measurement value.

Field	Type	Offset (bytes)	Value
[measurement value]	INT32	0	Measurement value ×1000. For example, 12.345 mm is sent as 12345. If the value is invalid, this field is set to 0x80000000.

STRING BLOCK

The String block returns the string scan output.

Field	Type	Offset (bytes)	Value
[string value]	STRING	0	String value. Unused space will be set to null (0x0). The string is otherwise not null terminated. If invalid, the block's buffer is entirely set to null, equivalent to an empty string. The string will be truncated if the block's size is smaller than the length of the string. Block size is 8 bytes by default and can be configured in the GoPxL UI.

SYSTEM STATUS BLOCK

The SystemStatus block provides system level status information, as well as details on the last executed command through EtherNet/IP.

Field	Type	Offset (bytes)	Value
System State	UINT8	0	Indicates if the system is in an active run state. 0= Stop 1 = Run
Alignment State	UINT8	1	Indicates sensor group alignment. 0 = Not aligned 1 = Aligned 2 = Align State Active
Buffer (Reserved)	UINT8	2	Reserved for Laser Safety Status.
Buffer (Reserved)	UINT8	3	Reserved. Unused padding byte. Read as 0.
Last Command Sequence Number	UINT16	4	Last command sequence number received through EtherNet/IP command input block.
Last Command Identifier	UINT16	6	Identifier of most recent command reference by "Last command sequence number". This represents only the command received through EtherNet/IP.
Command Status	UINT8	8	Status of the reference command: 0 = No Command Received 1 = In Progress 2 = Success 3 = Failure
Buffer (Reserved)	UINT8	9	Reserved. Unused padding byte. Read as 0.
Error Code	UINT16	10	Failure code for the reference command when Command Status is 3 (Failure). When Command Status is not 3 , this field is set to 0.
Buffer (Reserved)	UINT8 [4]	12-15	Reserved. Unused padding byte. Read as 0.

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. GoPxL 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 that in most applications you use a closed Ethernet subnet (that is, network switch, PLC, Gocator sensors, 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 of this guide. The exact hardware of your system may vary.

Requirements	Details
GoPxL firmware	1.0 and higher
Gocator families	The following families are supported: • laser line profiler sensors • fringe snapshot sensors • confocal line profile sensors
Required Files	GoPxL-EIP.eds GoPxL_EthernetIP.ACD
Other	Allen-Bradley L16ER-BB1B PLC Allen-Bradley Studio 5000 programming tool V21.03 or newer D-Link Unmanaged Industrial Gigabit Ethernet Switch DGS-108

The Ethernet card to which the Gocator device is connected should be added as a module to the Backplane. Verify that the IP Address is on the correct subnet. Note that the IP address should be that of the PLC's Ethernet modules, not that of the Gocator's.

Byte Order Options

GoPxL supports output in either Big Endian or Little Endian byte ordering options.

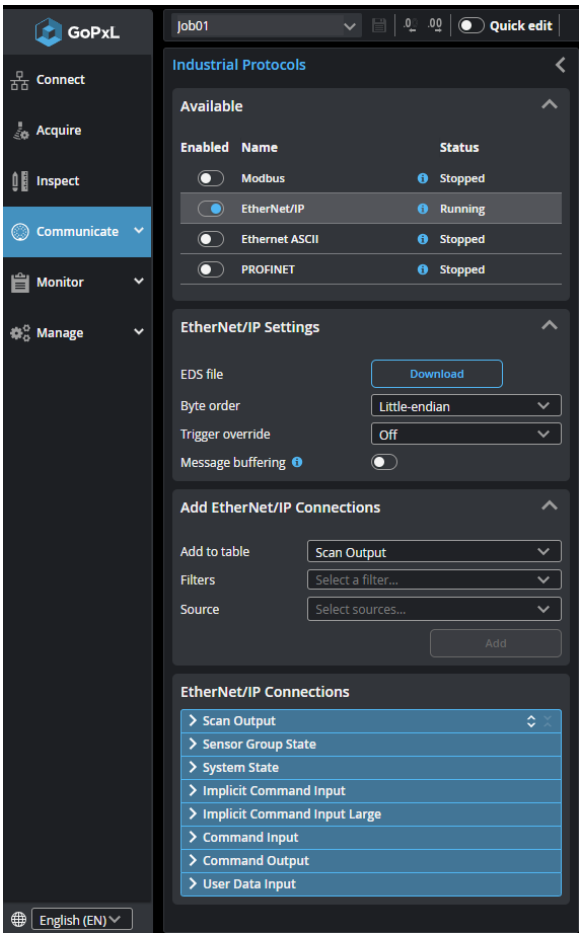
Big Endian Byte Order: The most significant byte (the "big end") of the data is placed at the byte with the lowest address. The rest of the data is placed in order of decreasing significance in the next three bytes of memory.

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.

Implicit Messaging - Setting Up GoPxL

To output in EtherNet/IP implicit messaging mode in GoPxL, you configure it in the **Communicate** > **Industrial** page. Note that the *type* of implicit messaging (cyclic versus change of state) is determined by the **Trigger Override** setting.



To configure GoPxL for EtherNet/IP implicit messaging mode:

1. On the **Industrial** page, select the EtherNet/IP row and enable it.

2. Set **Byte order** to **Little Endian**.
3. Make sure that **Message buffering** is disabled.
4. Check the **Implicit Messaging** option.
5. Set **Trigger override** to the type of implicit messaging you are using.

   For cyclic messaging, set **Trigger override** to **Force Cyclic**.

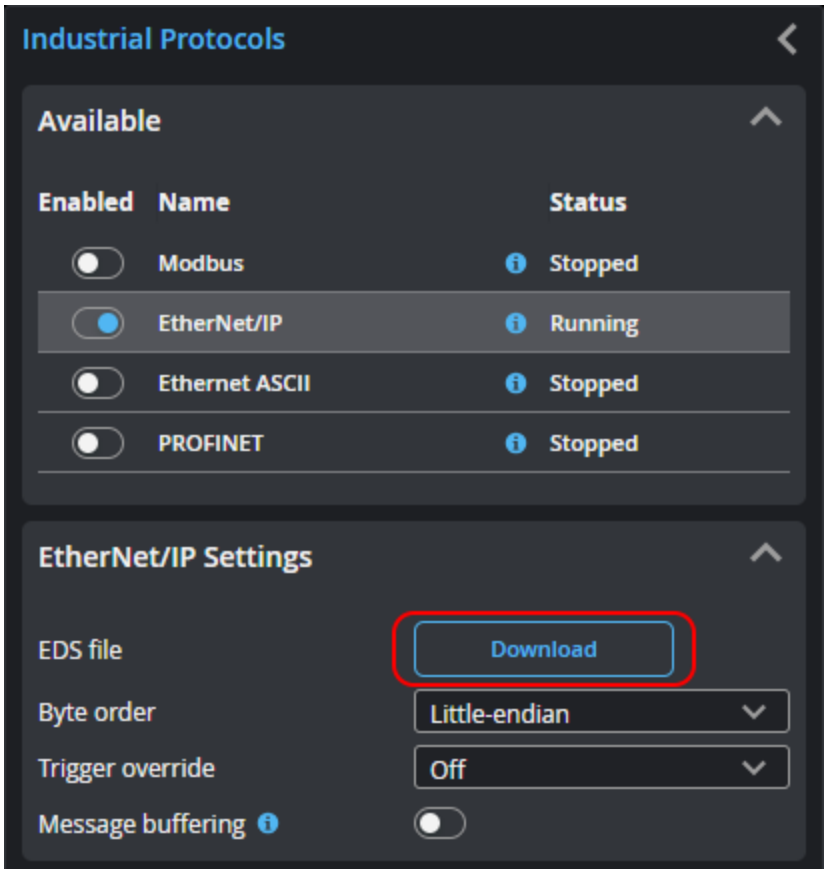
   For change of state messaging, set **Trigger override** to **Change of state**. When you set up the PLC to communicate with GoPxL using change of state implicit messaging, you must create an event task on the PLC to rapidly check whether the sensor is running; if the frame count increases, data is copied to an array. The event task period must allow the event task to be executed at a higher rate than the GoPxL frame rate. For more information, see *Implicit Messaging - Setting Up the PLC* on the next page.

   Under **Add EtherNet/IP Connections**, configure which values (such as stamps and measurements) to communicate to the PLC. For more details, see *EtherNet/IP Protocol* on page 1038.

Before setting up implicit messaging on the PLC, you must download the EDS file from GoPxL to the connected PC.

To download the EDS file:

1. In EtherNet/IP Settings, click **Download** to download the EDS file.



   Make note of where the file is saved.

   You may need to tell your browser to accept the download.

Implicit Messaging - Setting Up the PLC

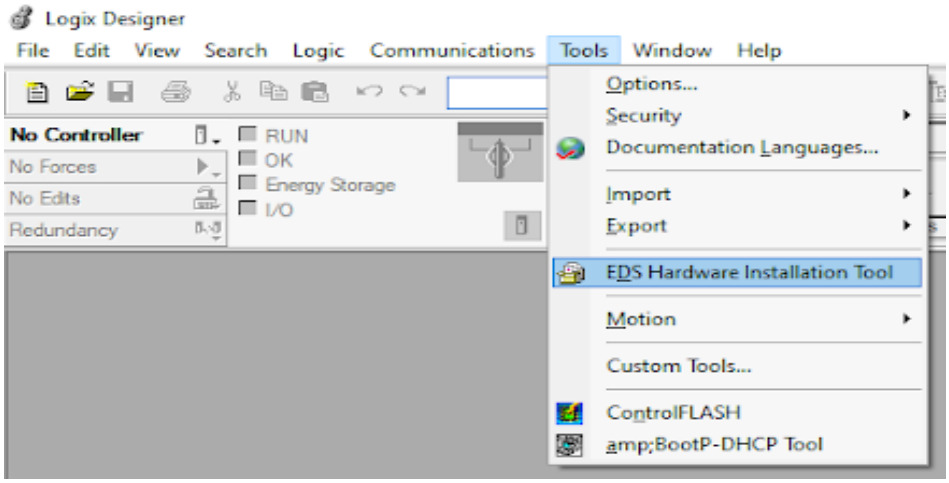
This section describes setting up implicit messaging on the PLC.

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**.

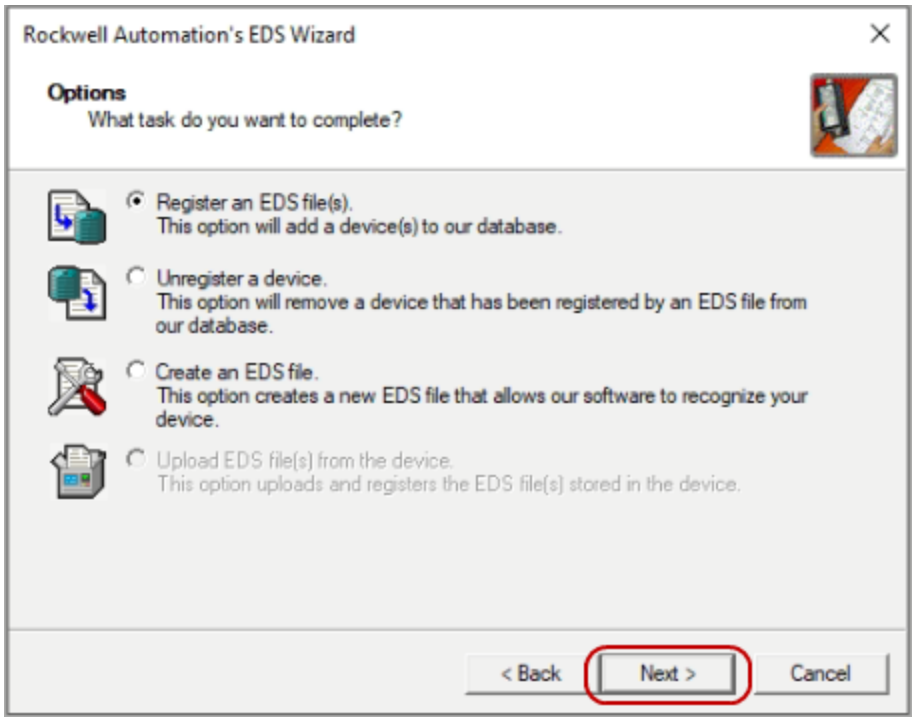


   The EDS setup tool (the Rockwell Automation EDS Wizard) launches.

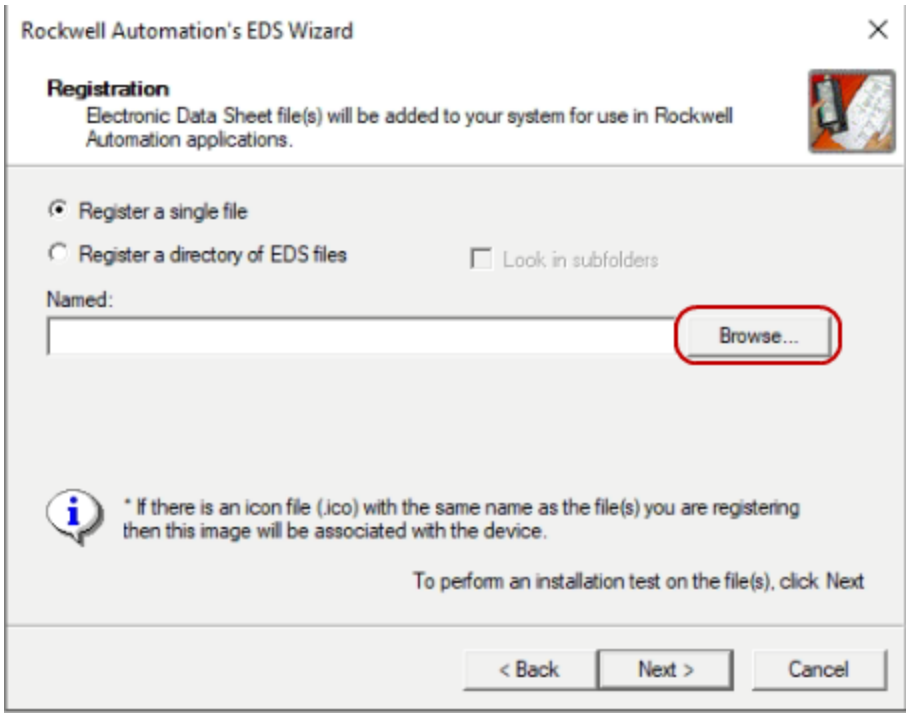
3. In the wizard, click **Next**.



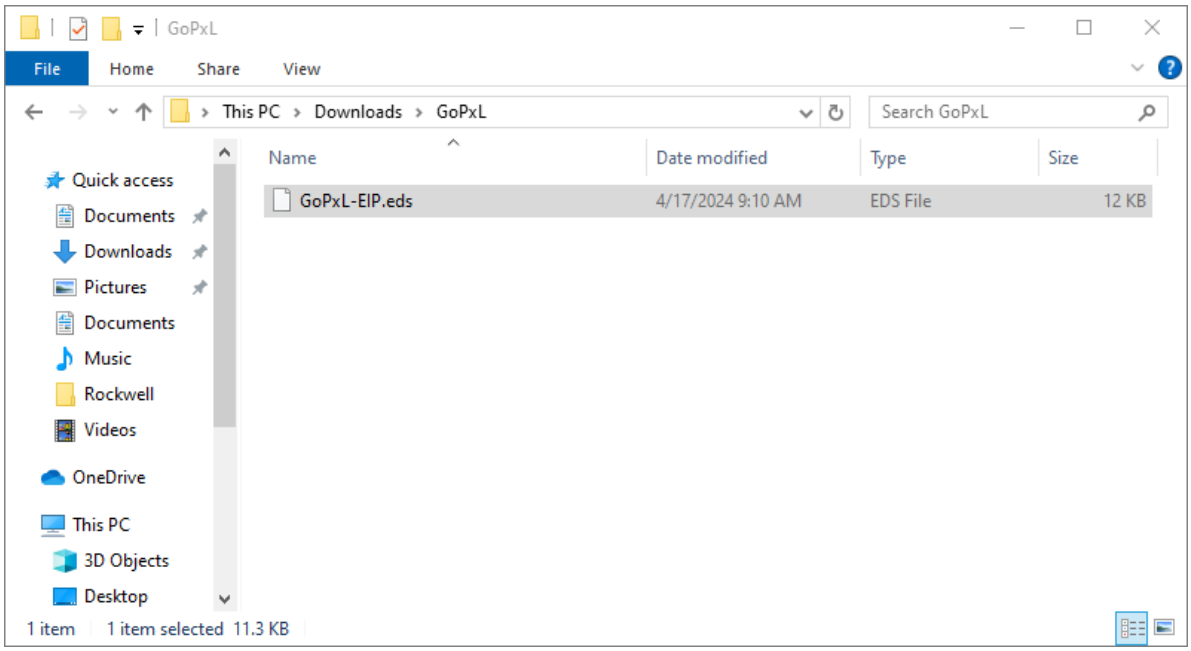
4. Choose **Register an EDS file(s)** and click **Next**



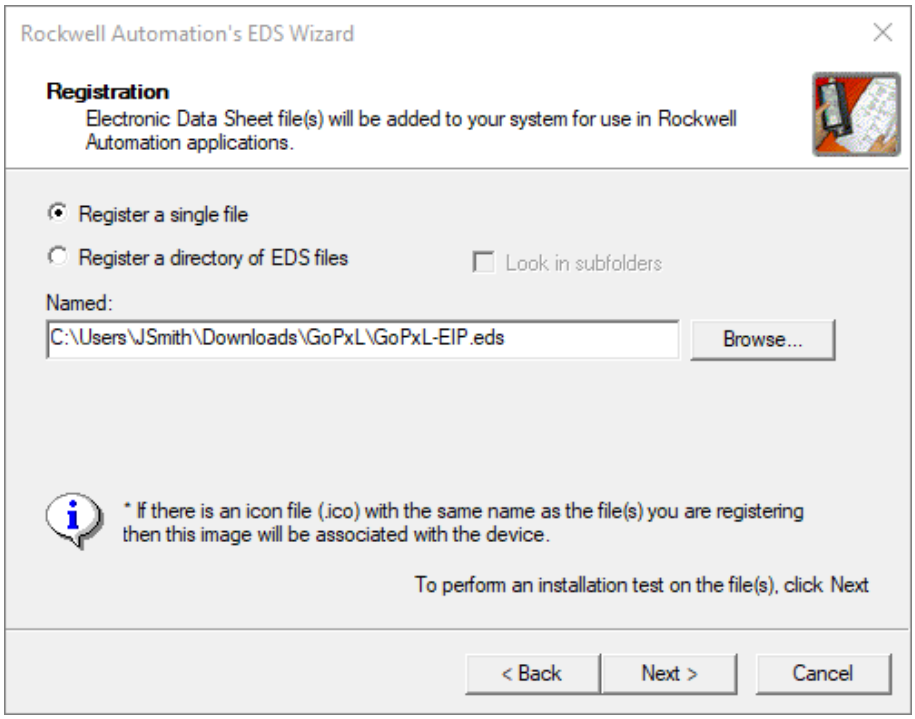
5. Choose **Register a single file** and then click **Browse**.



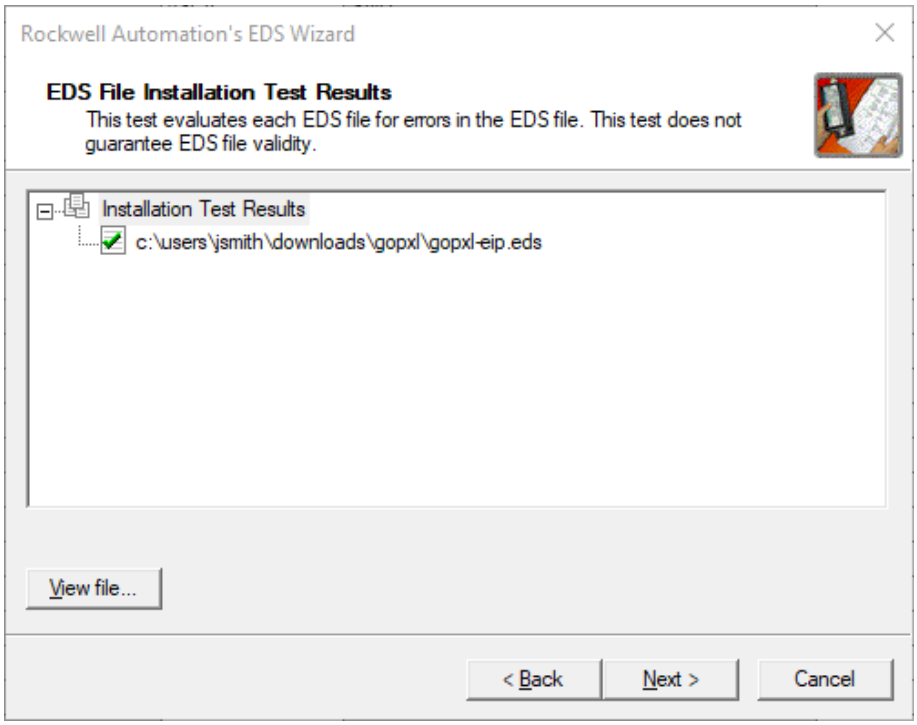
6. Browse to the .eds file you downloaded and unzipped, select it, and click **Open**.



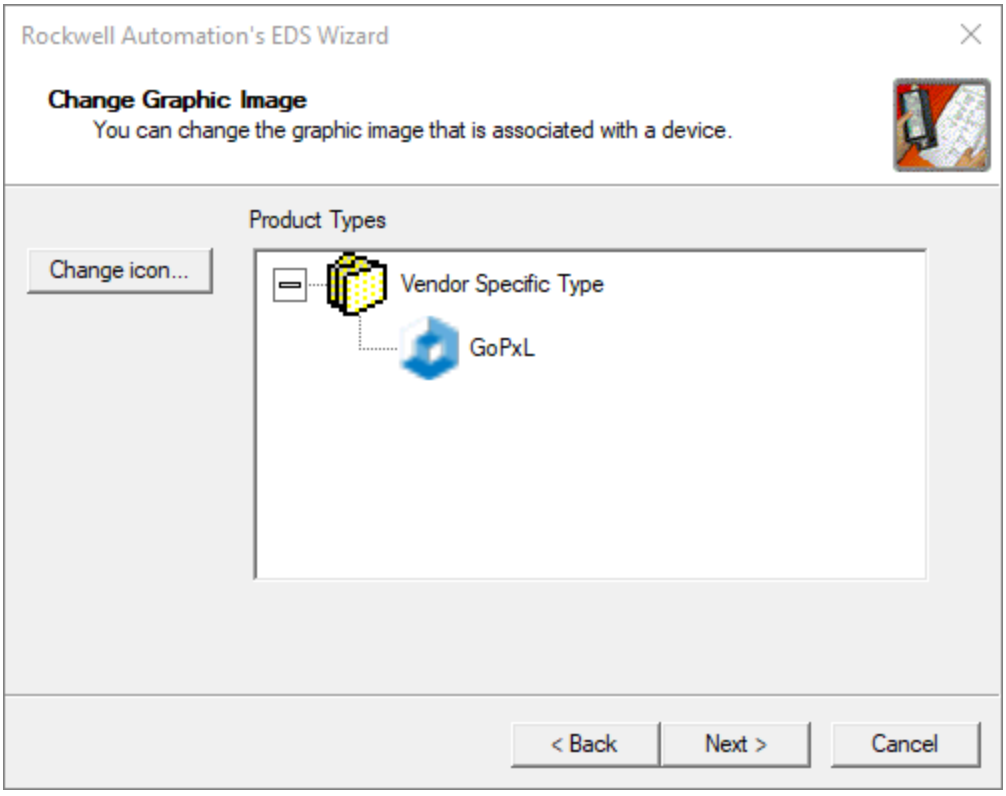
7. Click **Next**.



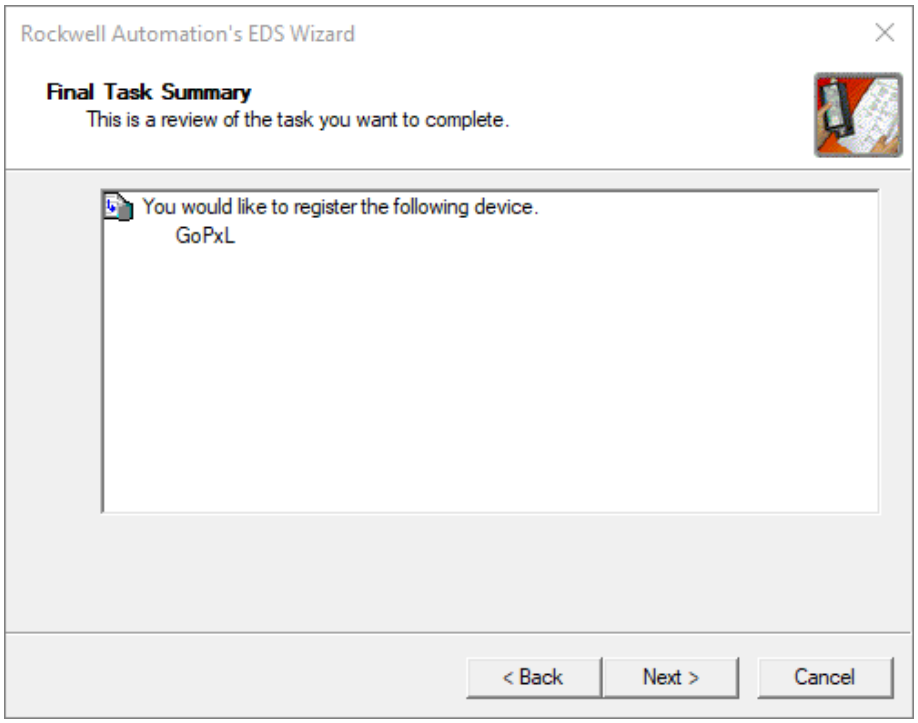
8. If your EDS file has no errors or conflicts (a green checkmark is displayed next to the .eds file), click **Next**.



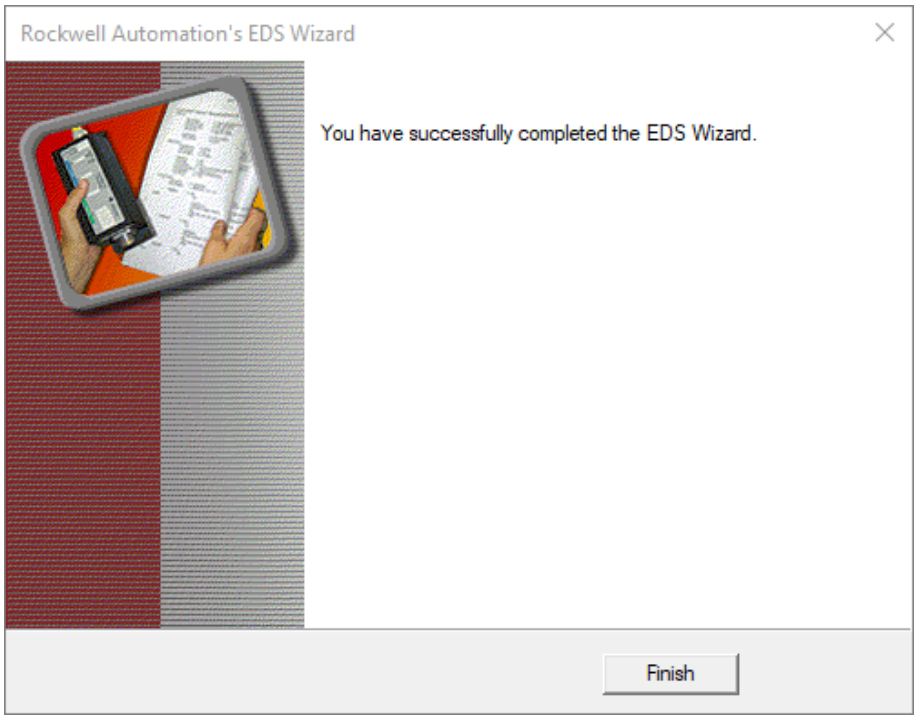
9. Verify that the tool automatically applies the GoPxL logo from the EDS file.



10. Click **Next**.

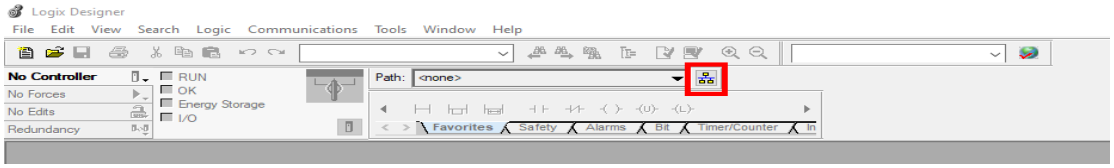


11. Click **Finish**.

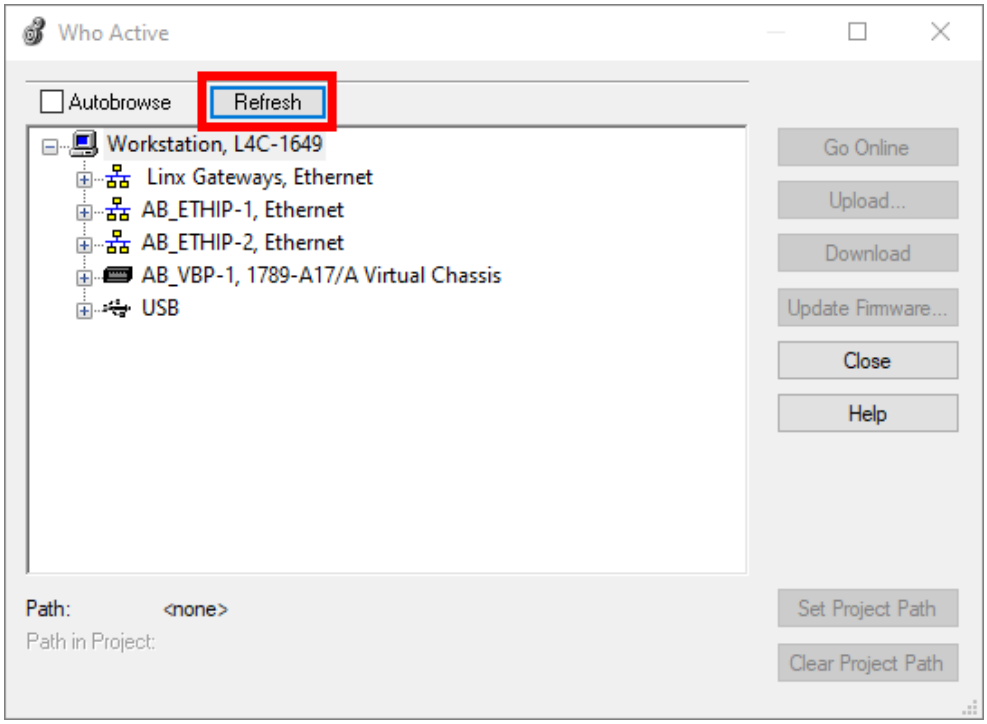


ADD GOCATOR IO DEVICE TO PLC PROGRAM

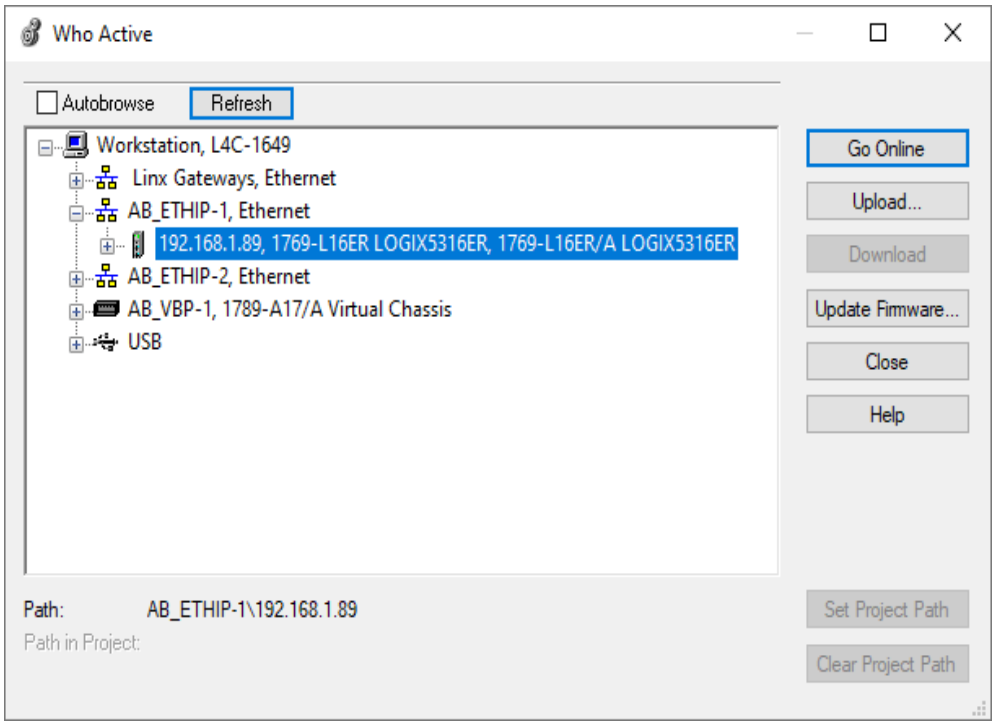
1. Click the **Who Active** button to the right of the **Path** field.



2. Click **Refresh** in the **Who Active** dialog to update your available devices.



3. In the tree structure, browse to and select your PLC controller Ethernet node.

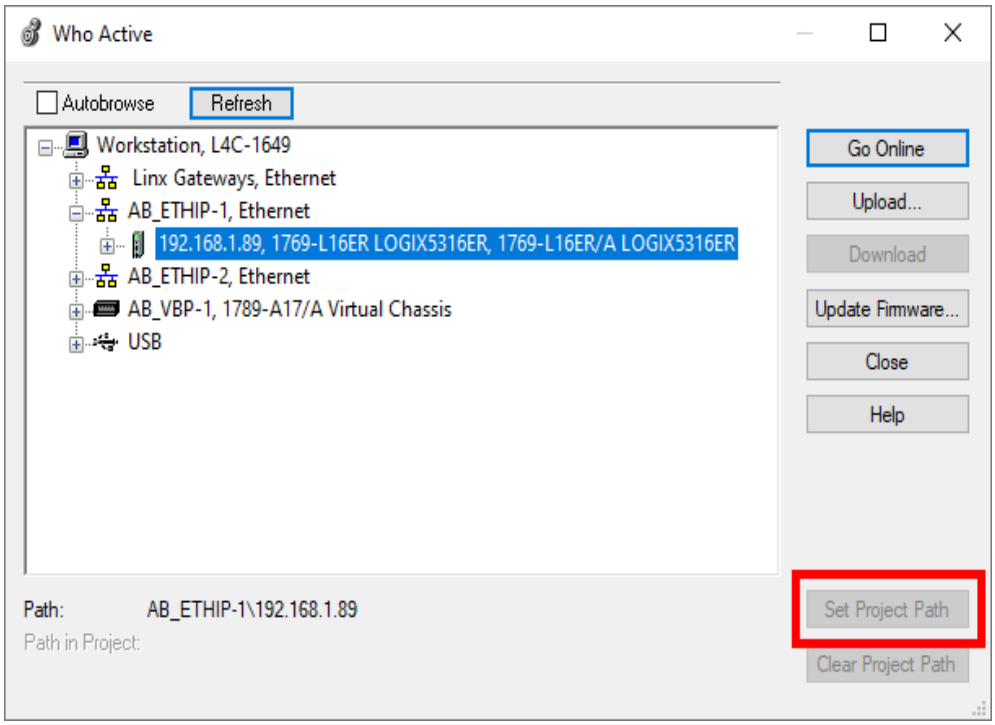


4. If you do not see the **Go Online** option at this point, make sure that RSLinx has been started and is running in the background on your setup PC.

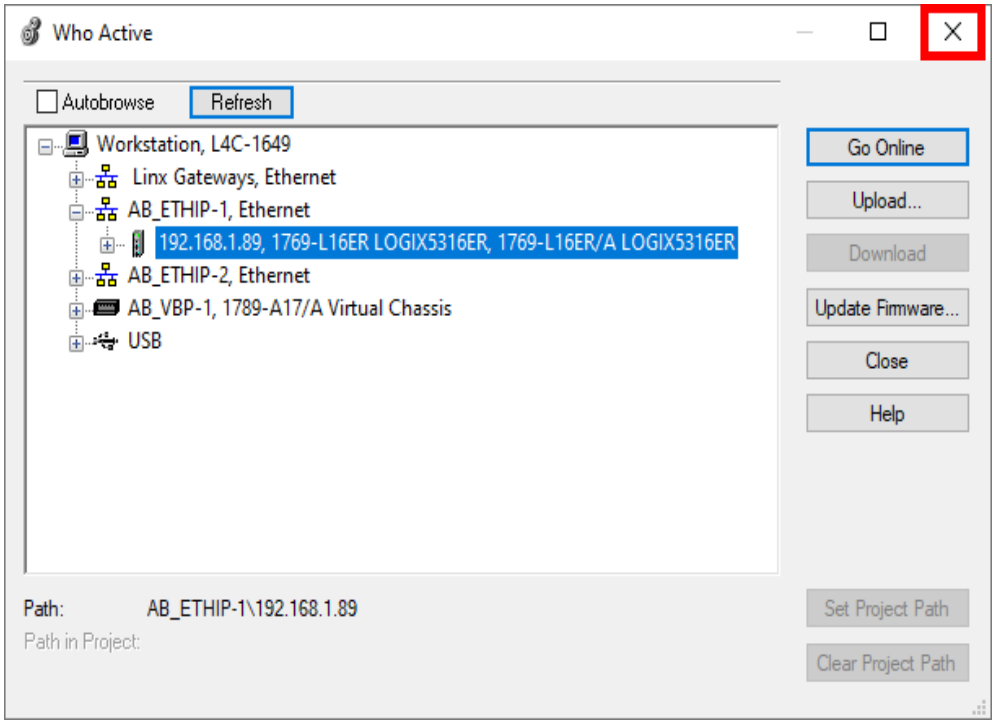
   Device discovery will not complete if RSLinx is not running.

5. Click **Set Project Path**.

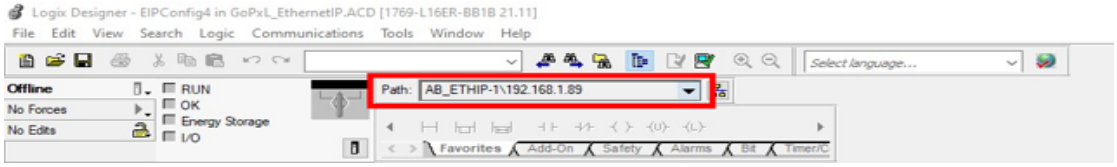
   This will set your project path when you attempt to download to the PLC later.



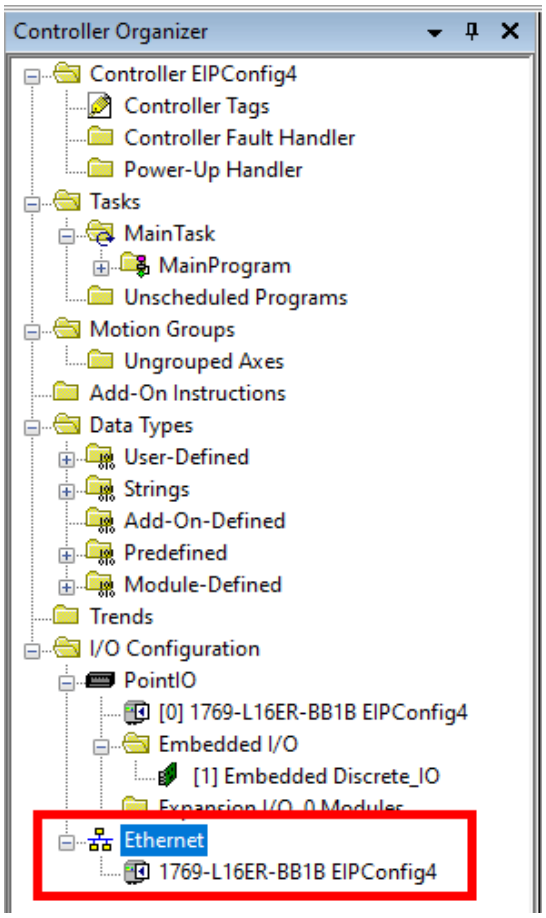
6. Click X to exit your node setup.



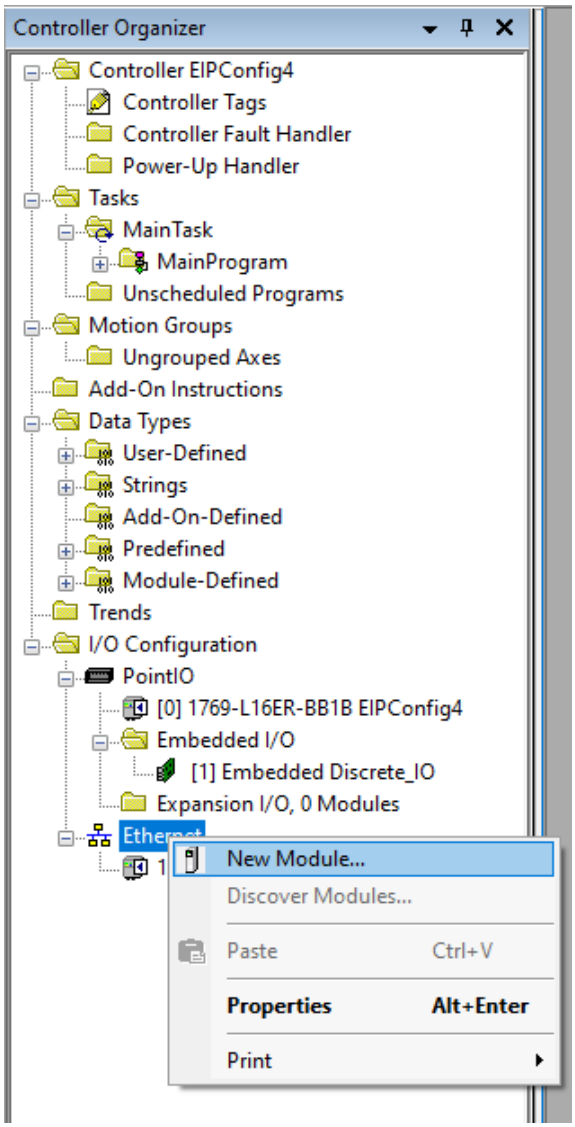
7. In Studio 5000, verify that the path is updated to the IP address of your controller.



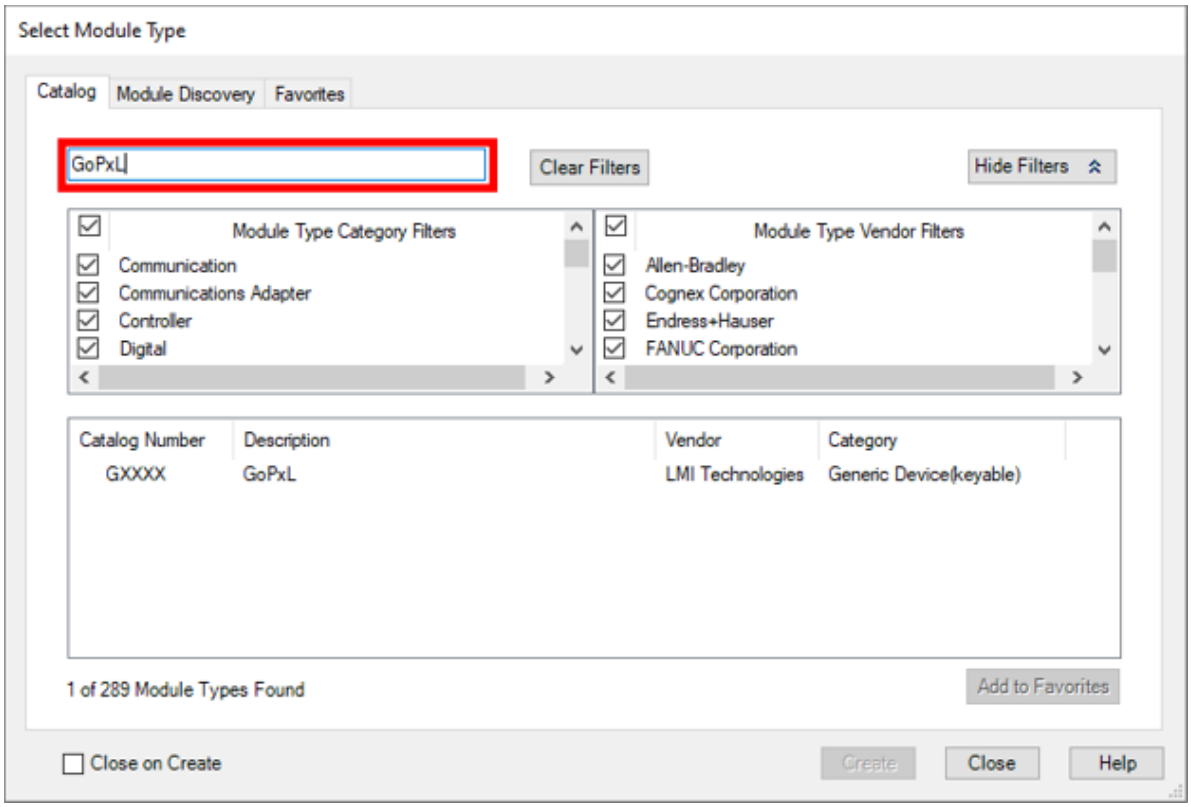
8. In the Controller Organizer, choose **Ethernet** under the **I/O Configuration** node.



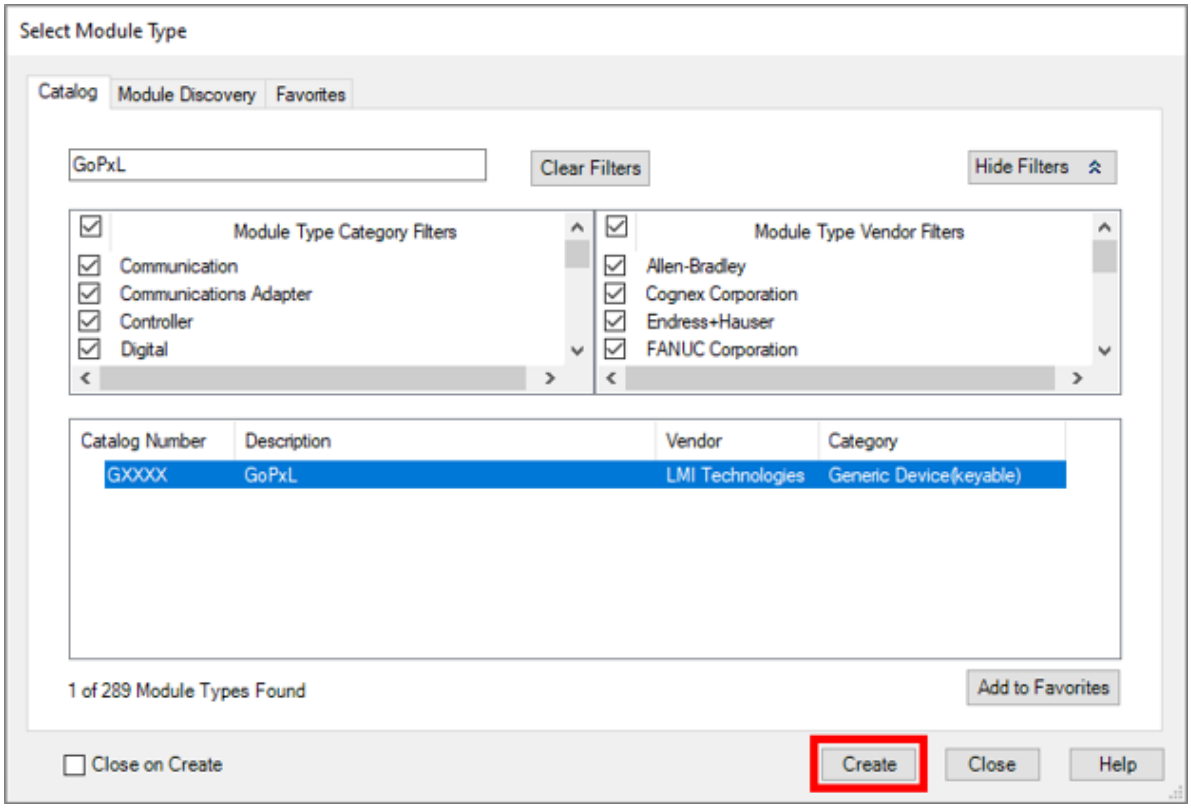
9. Right-click the Ethernet network node and click **New Module**.



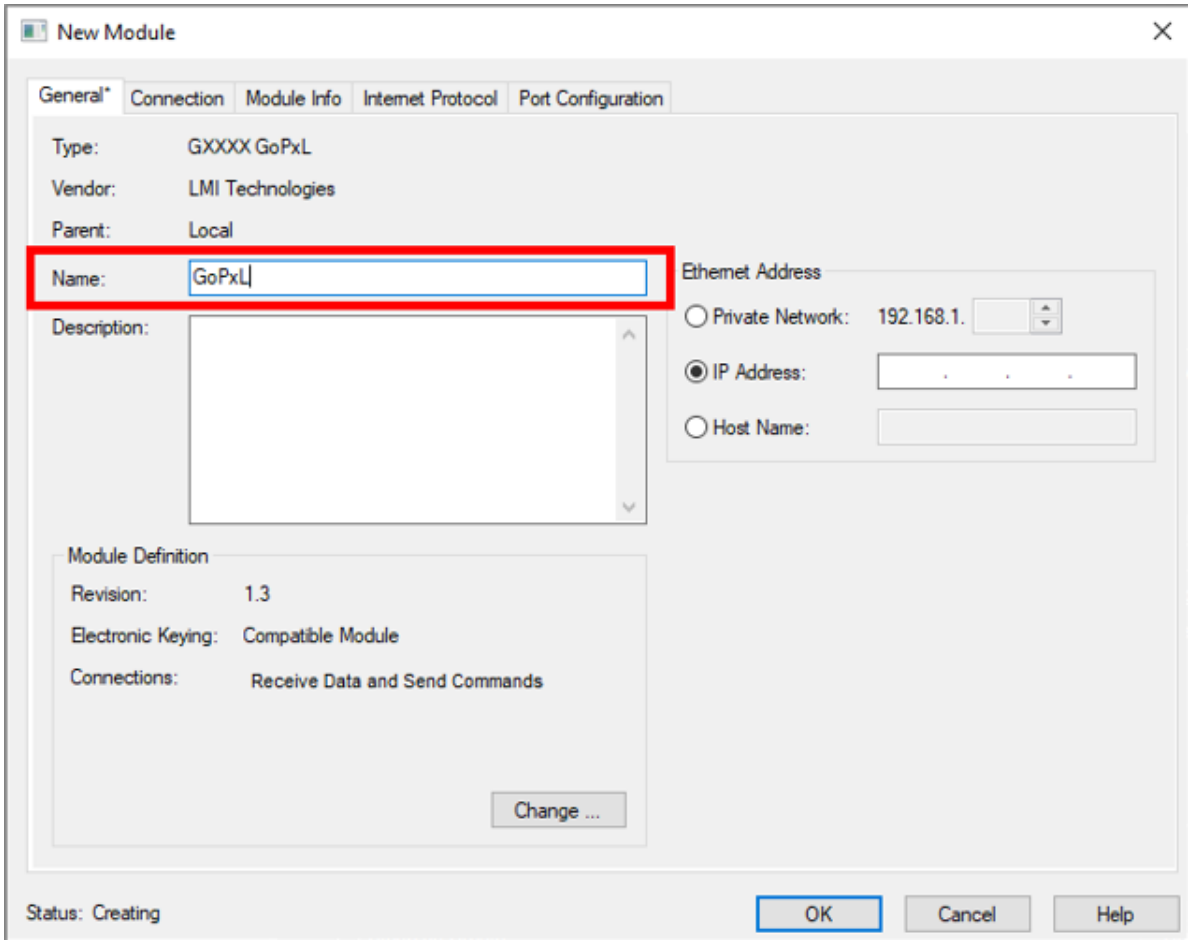
10. Type “GoPxL” into the search bar of the dialog that appears.



11. In the list under the search bar, select the new Gocator device file and click **Create**.

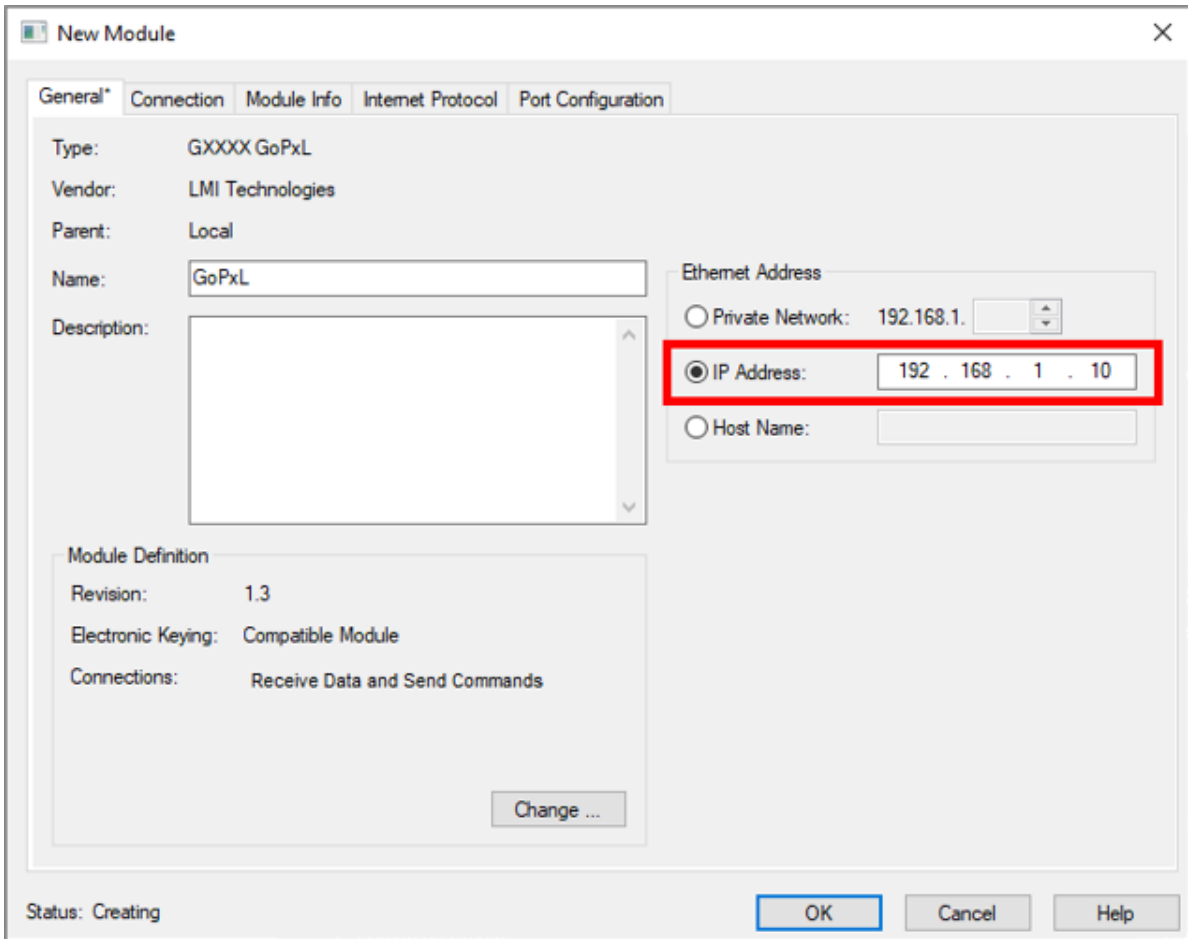


12. In the New Module dialog, in the **Name** field, give the new IO device a *unique* name.



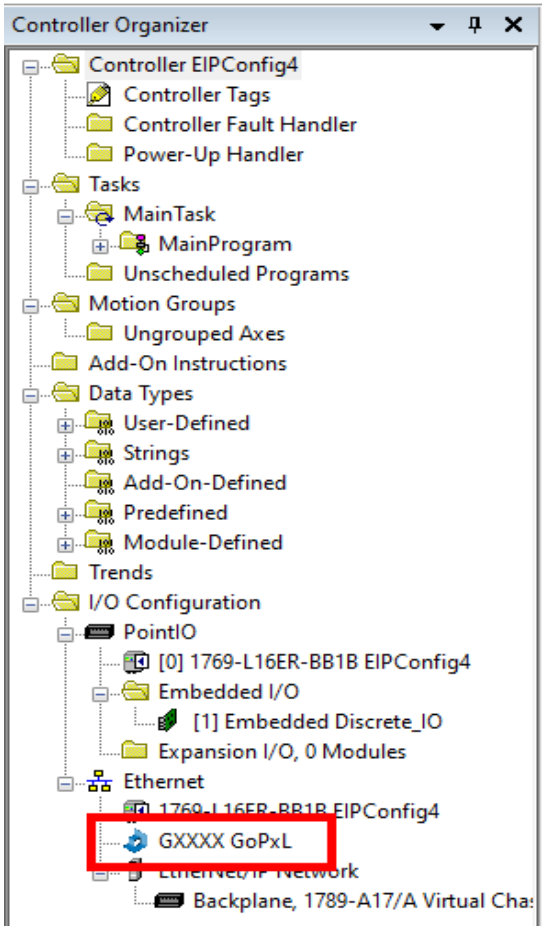
13. Type in the static IP address of the first sensor that you are trying to set up, and then click **OK**.

    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 connecting to the web interface in a browser or by using the Discovery tool (for more information, see *GoPxL Discovery Tool* on page 1290).

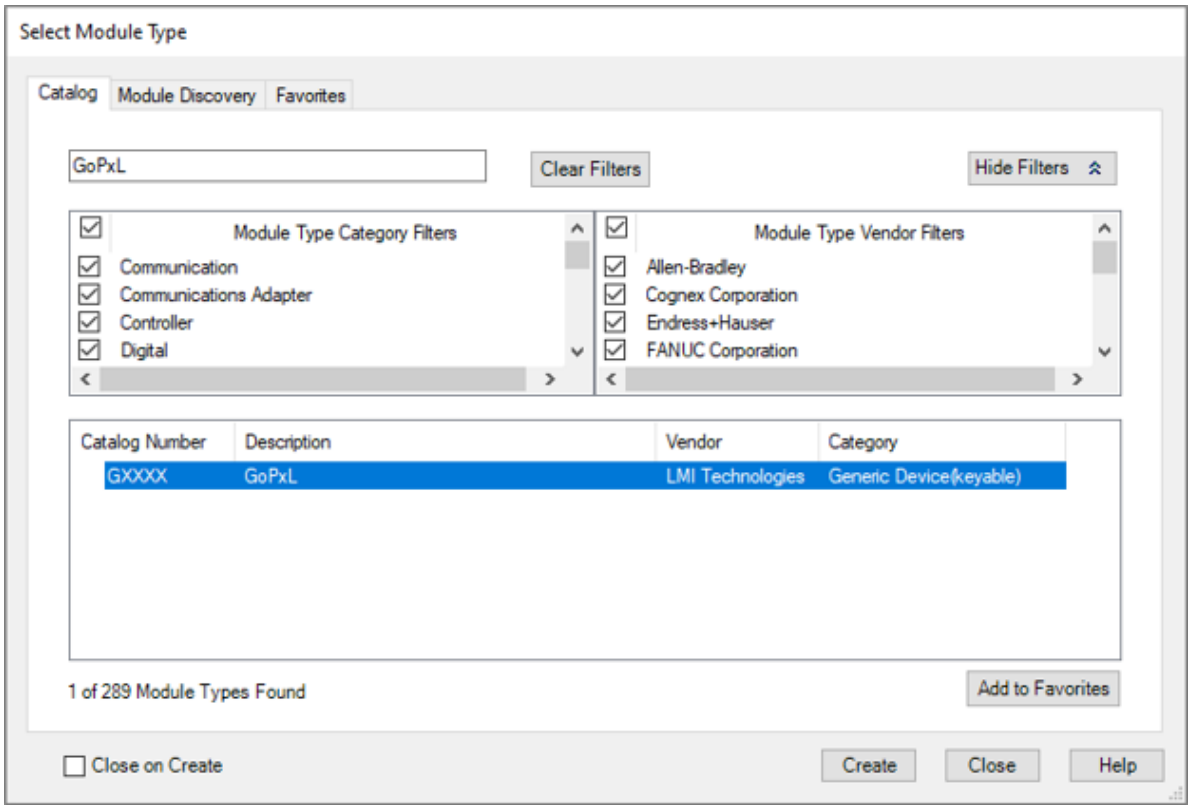


14. In your Program tree, verify that you now have a new IO device.

    The naming format shown should be device devicename (for example, GXXX GoPxL)

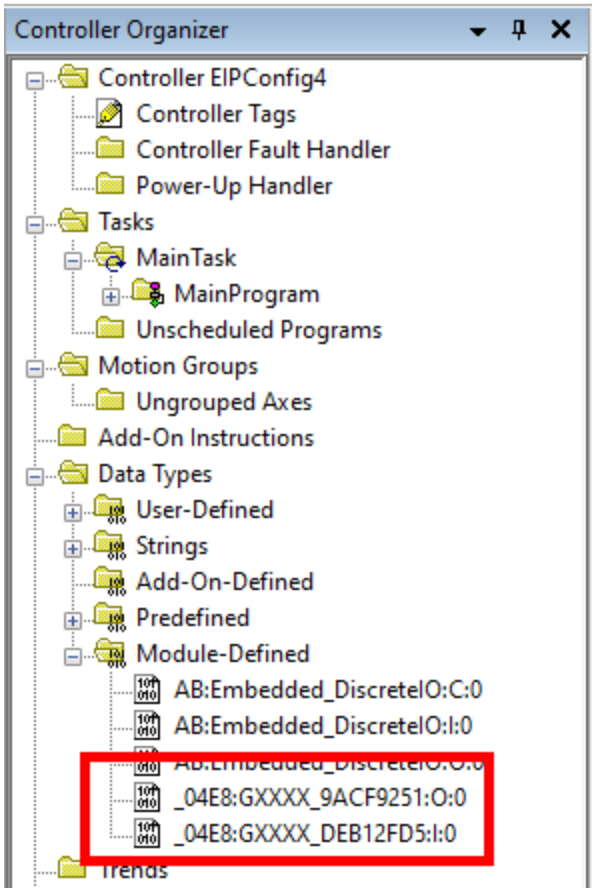


15. In the Select Module Type dialog, click **Close**.



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 GoPxL, respectively, for a *Monitor Data and Control Data* connection type.



When GoPxL is in Implicit Messaging mode, data will be streamed and stored in, for example, the G2510: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* on page 1058.

The EDS file now contains detailed tag descriptions as shown below that can be used directly in the PLC program.

Name	Data Type
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

Name	Data Type
ConnectionFaulted	BOOL
Z_Position_0	DINT
Z_Position_1	DINT
Encoder_Position_0	DINT
Encoder_Position_1	DINT
Time_0	DINT
Time_1	DINT
Frame_0	DINT
Frame_1	DINT
Digital_Input_States	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

Implicit Messaging - Using Gocator Command Assembly

The Output Message format (from PLC to GoPxL) is used to control GoPxL through implicit messaging. This message is sent continuously from the PLC to GoPxL at the user-requested Request Packet Interval (RPI) on the PLC side. The default Gocator RPI is 50 ms.

The total command message size is 67 bytes.

For information on the command assembly structure, see *Implicit Messaging Input* on page 1056 or *Implicit Messaging Input Large* on page 1057.

It's important to understand that because the Gocator is driven internally by its own clock, and because users can configure GoPxL for any frame rate, 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.

Before continuing, make sure you have set the data block used to send control messages to GoPxL properly; for more information, see *Implicit Messaging - Setting Up the PLC* on page 1064. It will appear in the Gocator Module-Defined data types as shown below:

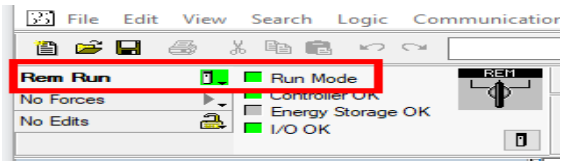
Name	Data Type
Command_Sequence_Number	INT
Command	SINT
Command_Argument1	SINT
Command_Argument2	SINT
Command_Argument3	SINT
Command_Argument4	SINT
Command_Argument5	SINT
Command_Argument6	SINT
Command_Argument7	SINT
Command_Argument8	SINT
Command_Argument9	SINT
Command_Argument10	SINT
Command_Argument11	SINT
Command_Argument12	SINT
Command_Argument13	SINT
Command_Argument14	SINT
Command_Argument15	SINT
Command_Argument16	SINT
Command_Argument17	SINT
Command_Argument18	SINT
Command_Argument19	SINT
Command_Argument20	SINT
Command_Argument21	SINT
Command_Argument22	SINT
Command_Argument23	SINT
Command_Argument24	SINT
Command_Argument25	SINT

STARTING A SENSOR

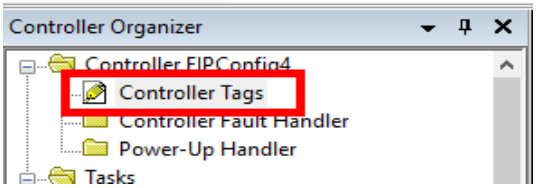
To start a sensor, do the following:

1. Make sure that you have downloaded your PLC program to the controller and that your controller is in Run mode.

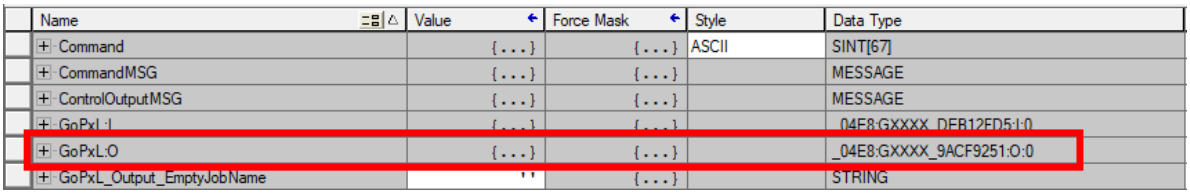
   For information on downloading the PLC program to the controller, see *Install EDS File* on page 1064.



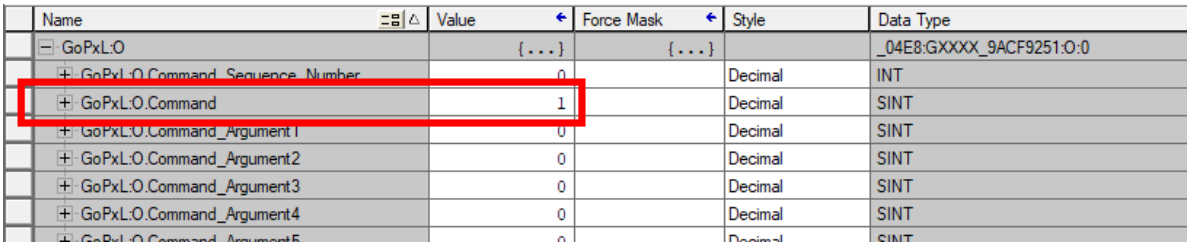
2. In the Controller Organizer, double-click **Controller Tags** to show them in the main screen



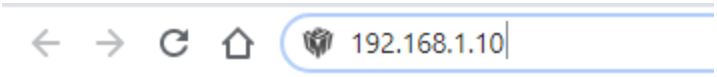
3. Click your Output data block to expand



4. Write the integer value 2 to the first byte named **Command**.



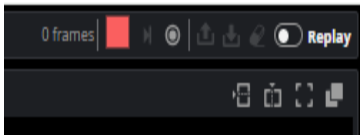
5. In a web browser, type in the sensor IP address to the URL bar.



This should load the GoPxL web interface. If not, verify that the sensor is powered up. If it is, the sensor may be configured to use a different IP address. Use the Discovery tool to find the sensor; for more information, see See *GoPxL Discovery Tool* on page 1290.

6. Verify that the sensor started.

   If the Run button is a red square, then the sensor was successfully started.



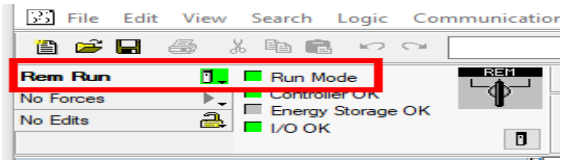
This process can be repeated to stop the sensor, clear alignment, start moving alignment, start stationary alignment, or issue a software trigger by typing the proper integer value into the Command byte of the Output assembly. For additional commands and control options, see *Implicit*

Messaging Input on page 1056 or *Implicit Messaging Input Large* on page 1057, or refer to the provided sample Studio 5000 job file.

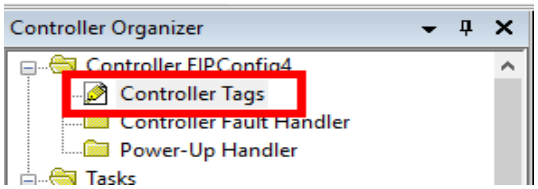
LOADING A SENSOR JOB FILE

1. Make sure that you have downloaded your PLC program to the controller and that your controller is in Run mode.

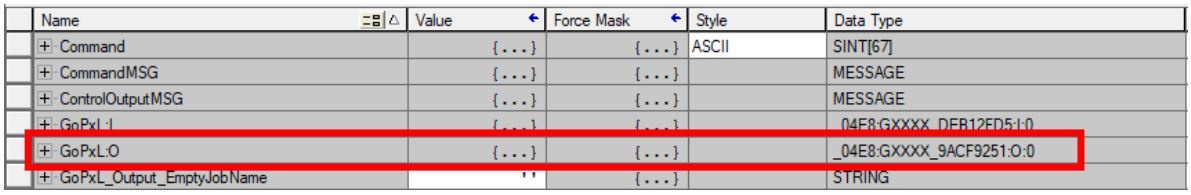
   For information on downloading the the PLC program to the controller, see *Install EDS File* on page 1064.



2. Double click **Controller Tags** to show them in the main screen

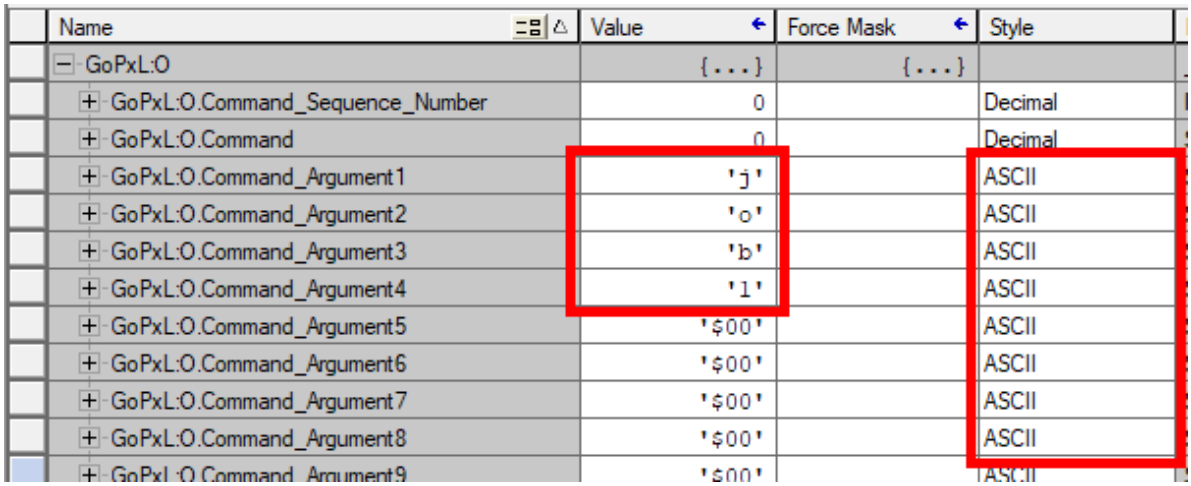


3. Click your Output data block to expand

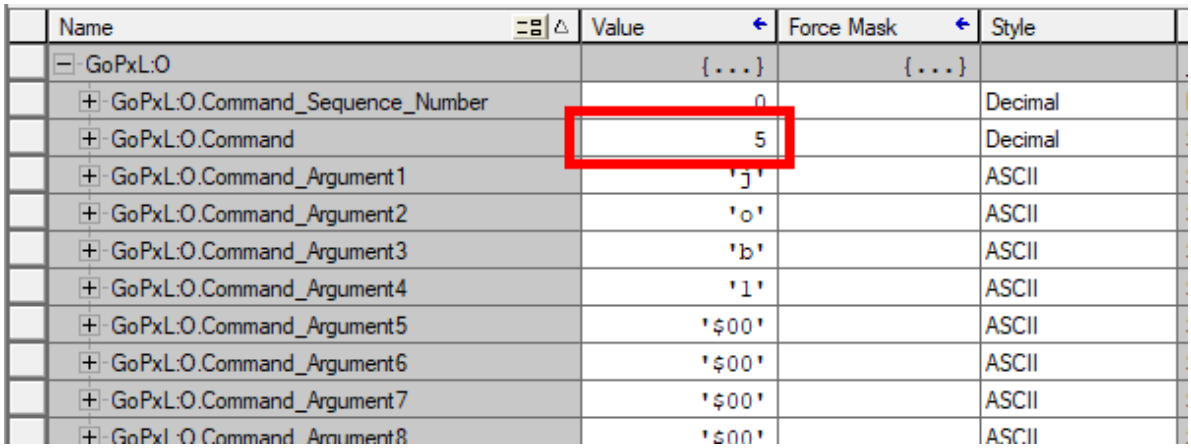


4. If `job1` is the name of the job file to be loaded on the sensor and it is not currently running, type each of the four characters making up the filename into the first four characters of the Command_Argument bytes of the Command assembly.

   The ASCII character inputs here are case sensitive and the extension, `.gpjob`, 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.



5. Type the integer value 5 into the **Command** byte to transmit the job name for loading.

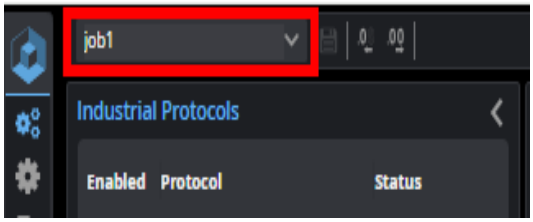


Name	Value	Force Mask	Style
GoPxL:O	{...}	{...}	
GoPxL:O.Command_Sequence_Number	0		Decimal
GoPxL:O.Command	5		Decimal
GoPxL:O.Command_Argument1	'j'		ASCII
GoPxL:O.Command_Argument2	'o'		ASCII
GoPxL:O.Command_Argument3	'b'		ASCII
GoPxL:O.Command_Argument4	'1'		ASCII
GoPxL:O.Command_Argument5	'$00'		ASCII
GoPxL:O.Command_Argument6	'$00'		ASCII
GoPxL:O.Command_Argument7	'$00'		ASCII
GoPxL:O.Command_Argument8	'$00'		ASCII

6. In a web browser, type the sensor IP address to the URL field.



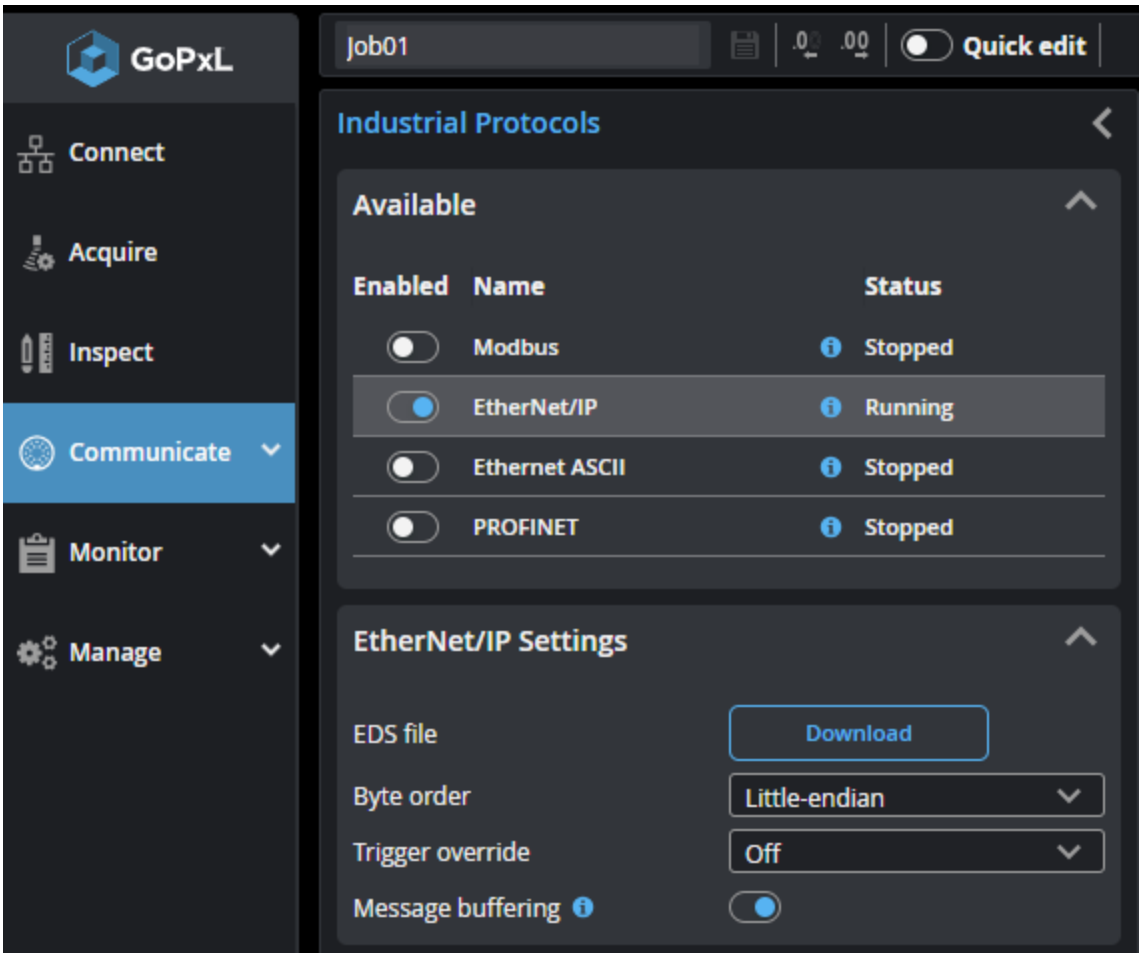
7. After the web GUI loads, verify that the job was loaded on GoPxL by looking at the job name box.



For additional commands and control options, see *Implicit Messaging Input* on page 1056 or *Implicit Messaging Input Large* on page 1057, or refer to the provided sample Studio 5000 job file.

Explicit Messaging - Setting Up GoPxL

To output data over EtherNet/IP in explicit messaging mode, you configure it in the **Communicate** > **Industrial** page.



To configure the sensor for EtherNet/IP explicit messaging mode:

1. On the **Industrial** page, select the EtherNet/IP row and enable it.
2. Choose **Little Endian** from the **Byte Order** dropdown box.
3. Enable the **Message Buffering**.

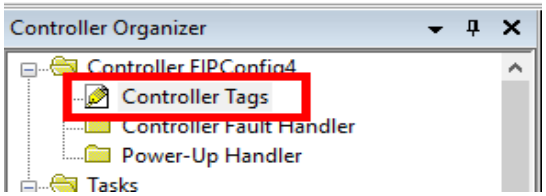
Explicit Messaging - Read Single Attribute on PLC

This section shows how to read the serial number from GoPxL, that is, attribute 6. (For more on the Identity Object, see *Identity Object (Class 0x01)* on page 1048.)

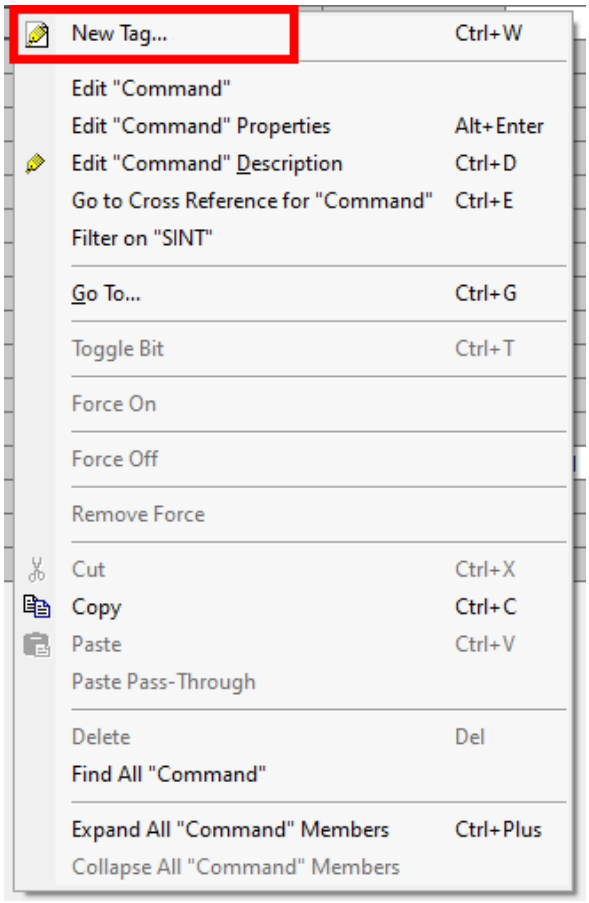
Before attempting to control and run the Gocator from the PLC, you should always verify the connection first by reading an attribute from the Identity Object, for example the sensor's serial number. LMI recommends following the steps described in this section before trying to control the sensor.

To read the sensor's serial number:

1. In Studio 5000, in the Controller Organizer, expand **Controller Tags** by double-clicking it.

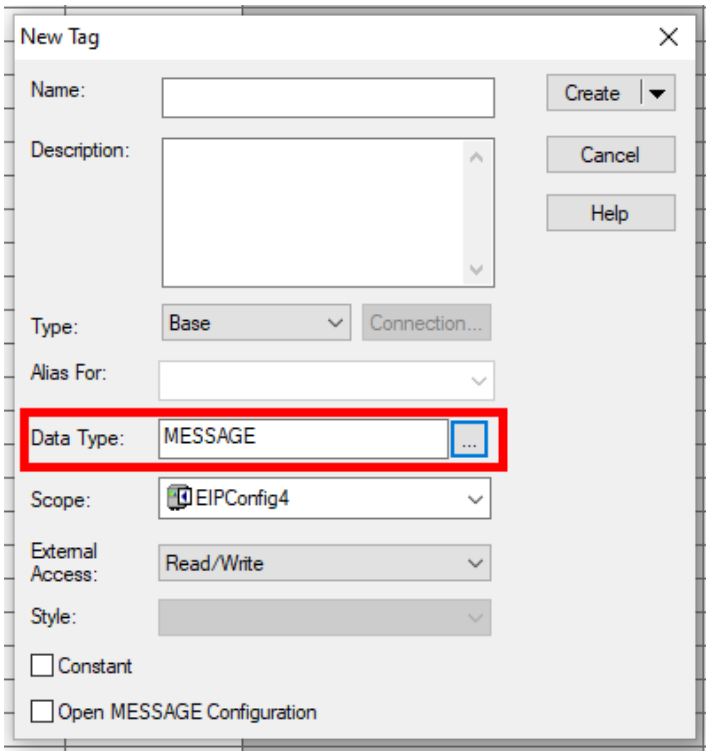


2. Right-click in the middle of the screen and choose **New Tag** from the context menu.

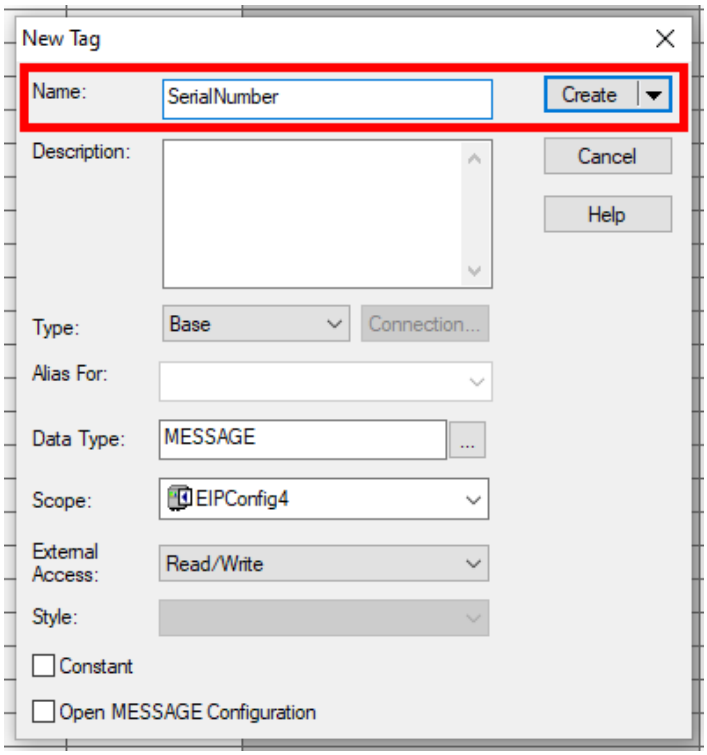


3. In the New Tag dialog, change the data type to MESSAGE.

   This creates a block to store parameters for requesting data from GoPxL.

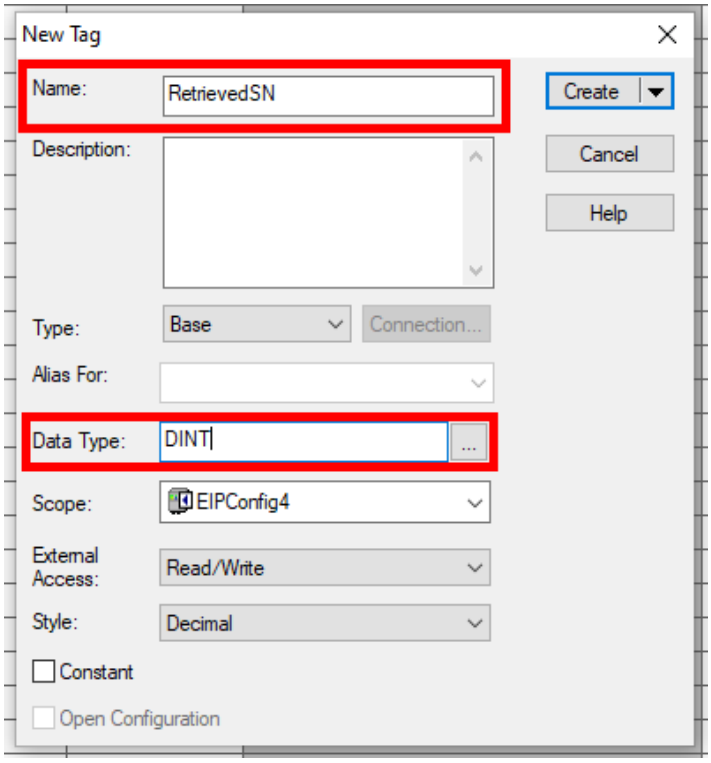


4. Name the tag and click **Create**.

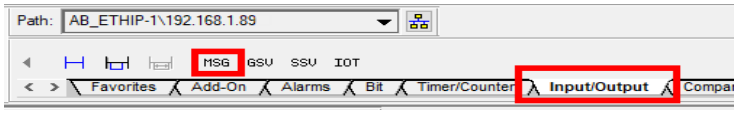


5. Right-click in the middle of the screen again and choose **New Tag** from the context menu.
6. Change the data type to DINT and name the tag.

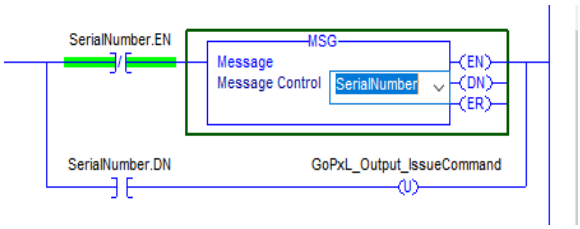
This will create a tag to store the serial number in. The type must match the data type of the attribute you want to get. To determine the type of the attribute, see *Identity Object (Class 0x01)* on page 1048.



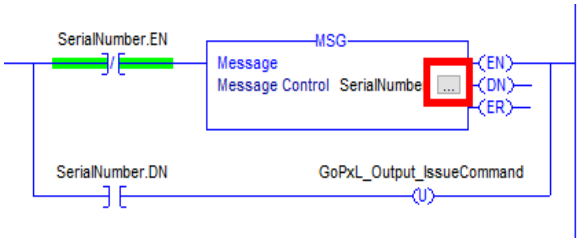
7. In the ladder, navigate to the Input/Output function blocks and click MSG to add a Message function block. You may need to add a new rung to allow this.



8. Once the new MSG function block has been added, click the tag drop-down and select the MSG tag you created earlier.

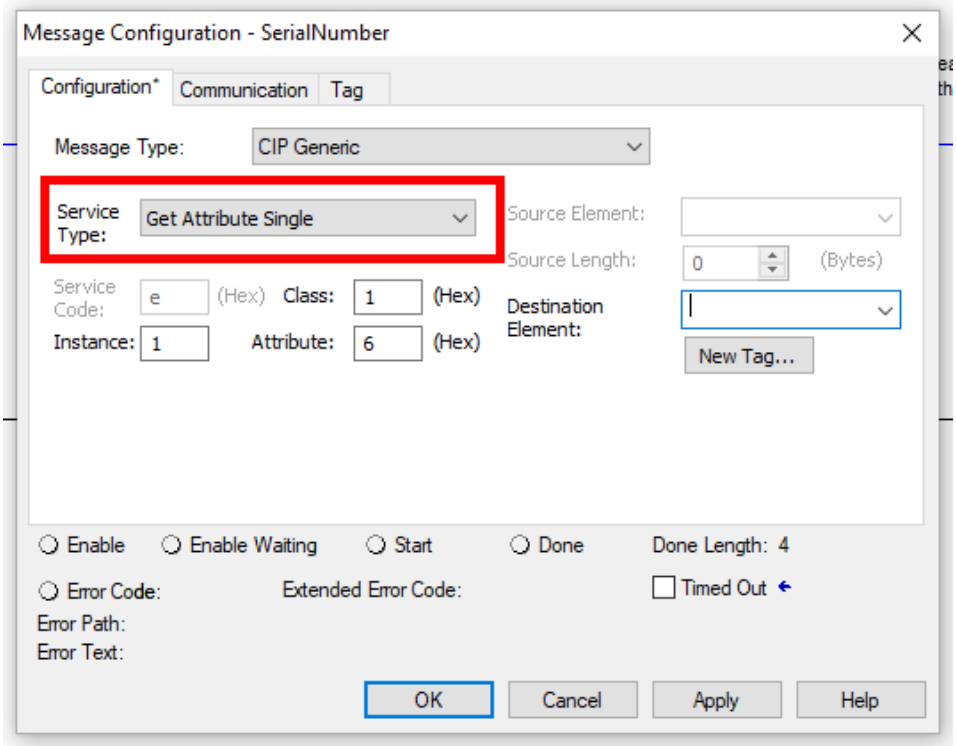


9. Click the grey box to open the Configuration Dialog box.



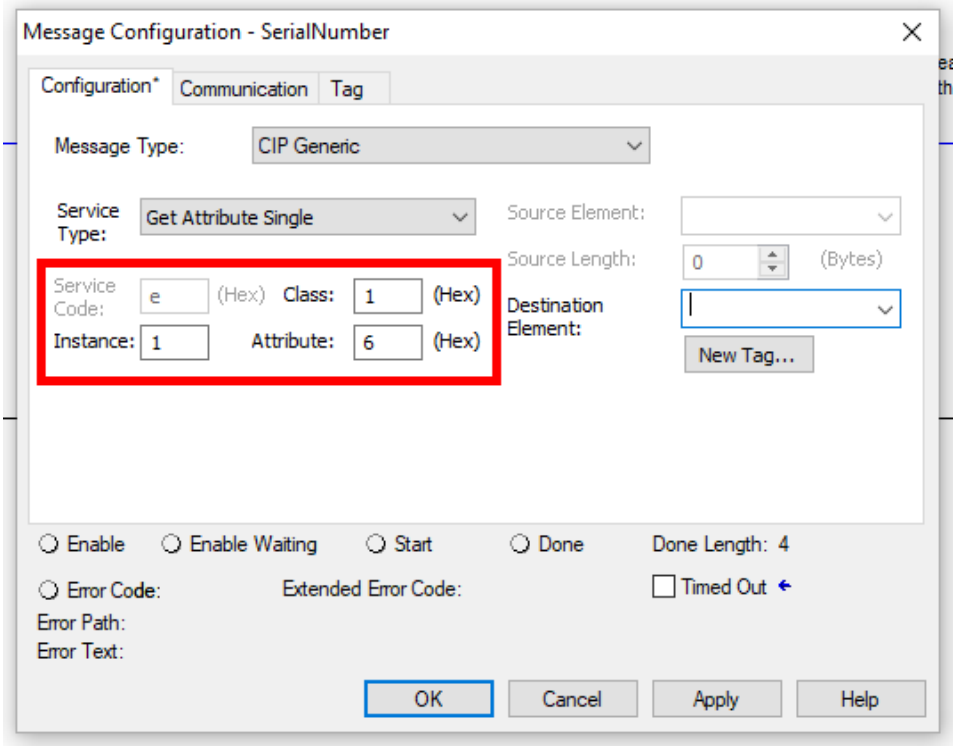
10. Choose the **Get Attribute Single** function from the **Service Type** drop-down.

    This will auto-populate the Service Code hex character.

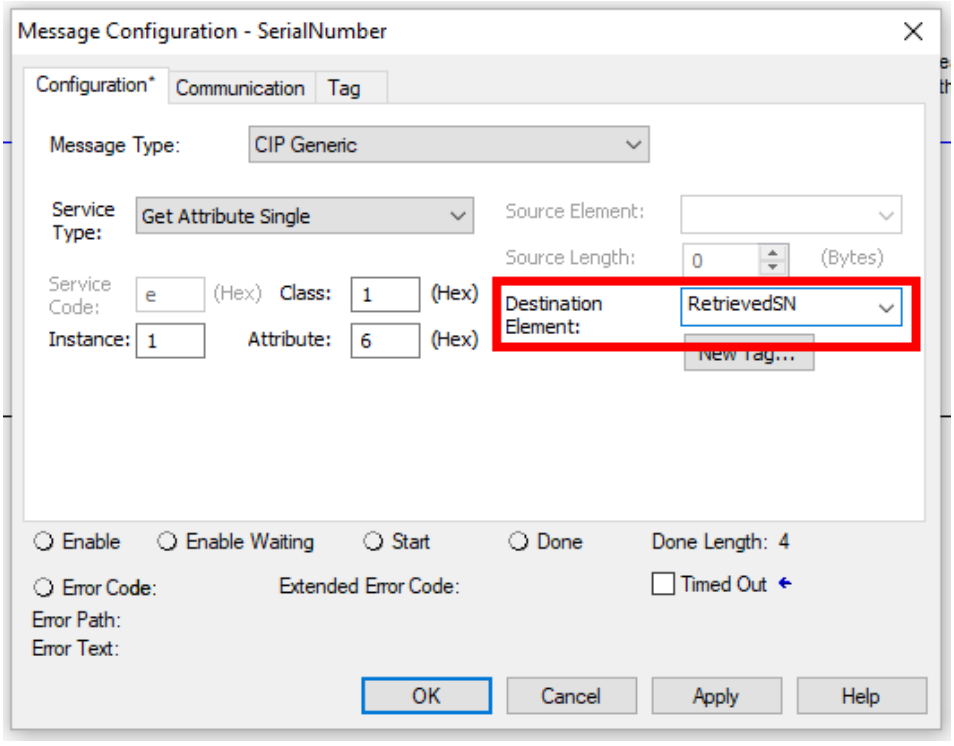


11. Type 1 in **Class**, 1 in **Instance**, and 6 in **Attribute**.

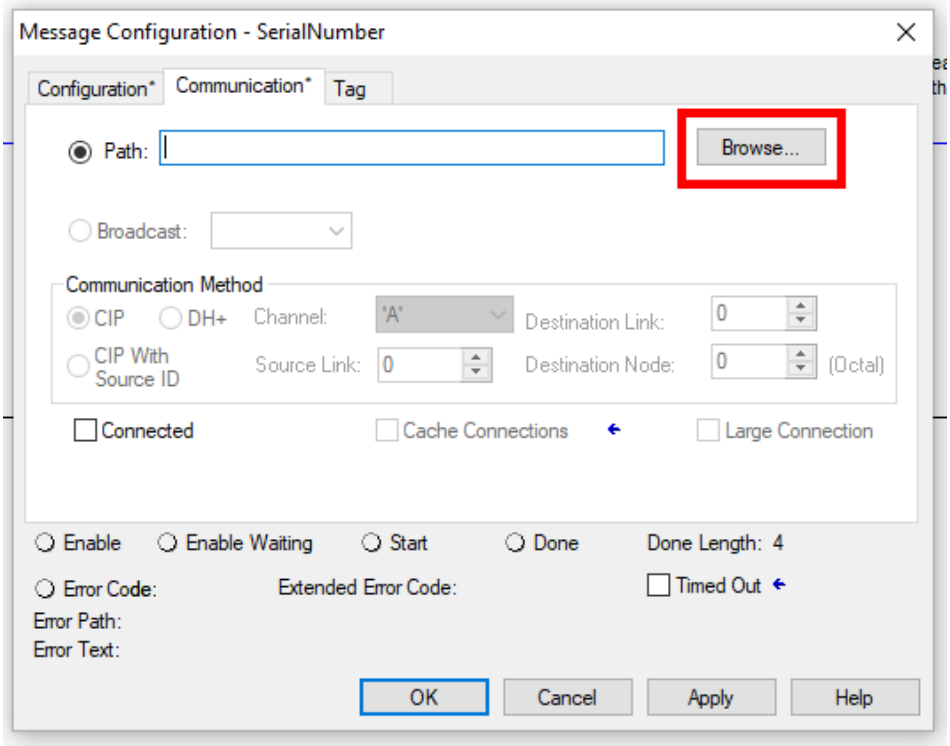
    These settings indicate that the sensor's serial number will be retrieved.



12. Choose the DINT tag you created to store the serial number from **Destination Element**.

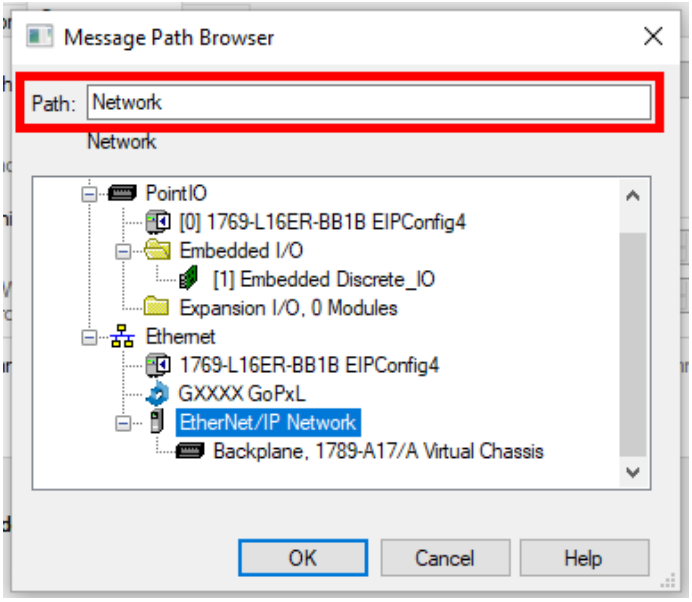


13. On the **Communication** tab, click **Browse**.



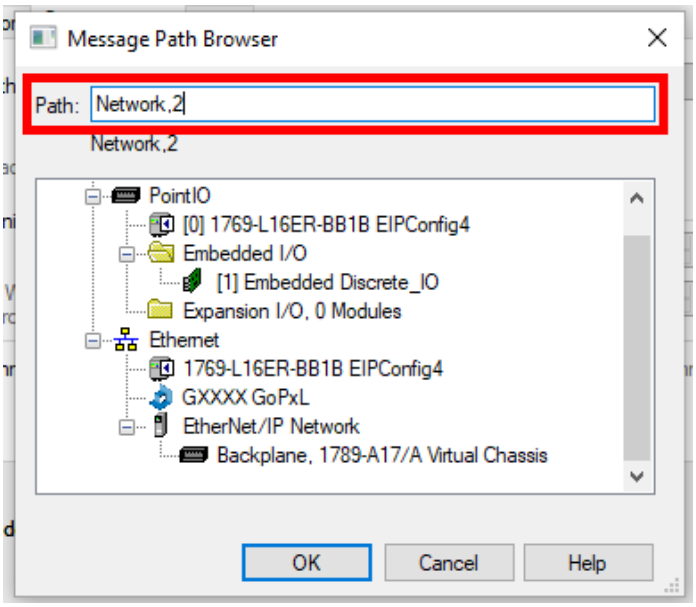
14. In the Message Path Browser dialog, click the **EtherNet/IP Network** node.

    This will route communication messages to the EtherNet/IP network.



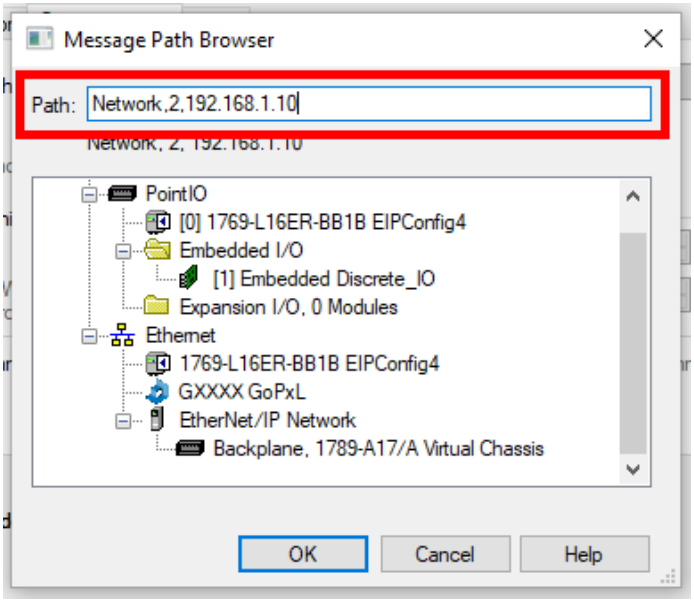
15. In **Path**, type the Ethernet port on the PLC that is physically connected to the GoPxL device, after the name in the field.

    Here, the port “2” is added.



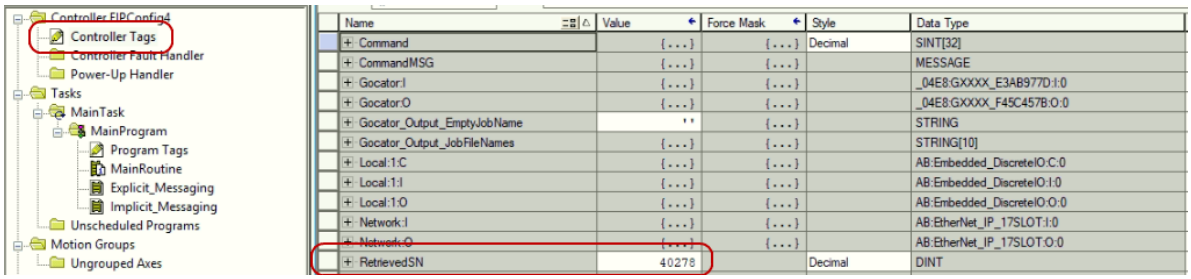
16. In **Path**, type the IP address of the GoPxL device to complete the path.

    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 *Explicit Messaging Sample State* on page 1053. 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.

Explicit Messaging - Set Single Attribute to GoPxL on PLC

You use the Command assembly to do the following:

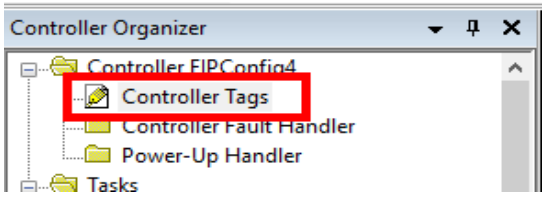
- Start a sensor
- Stop a sensor
- Align a sensor
- Clear sensor alignment
- Set a sensor's runtime variables
- Load a job on a sensor
- Trigger a sensor

To see the information needed to properly configure the control byte, see *Explicit Messaging Command Input* on page 1049.

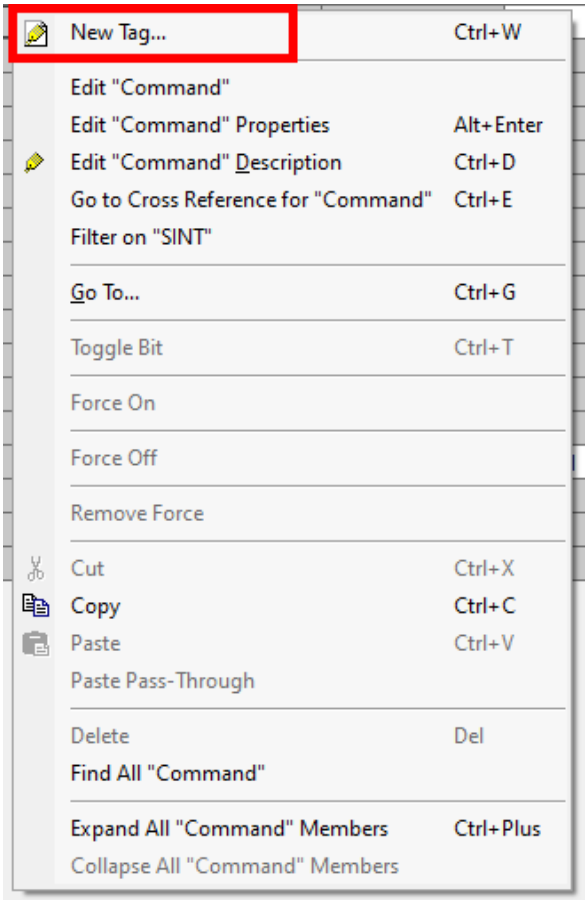
> LMI recommends following the steps in *To read the sensor's serial number:* on page 1085 to verify the communication path and message block *before* attempting to control a sensor.

To set a single attribute to the sensor on the PLC:

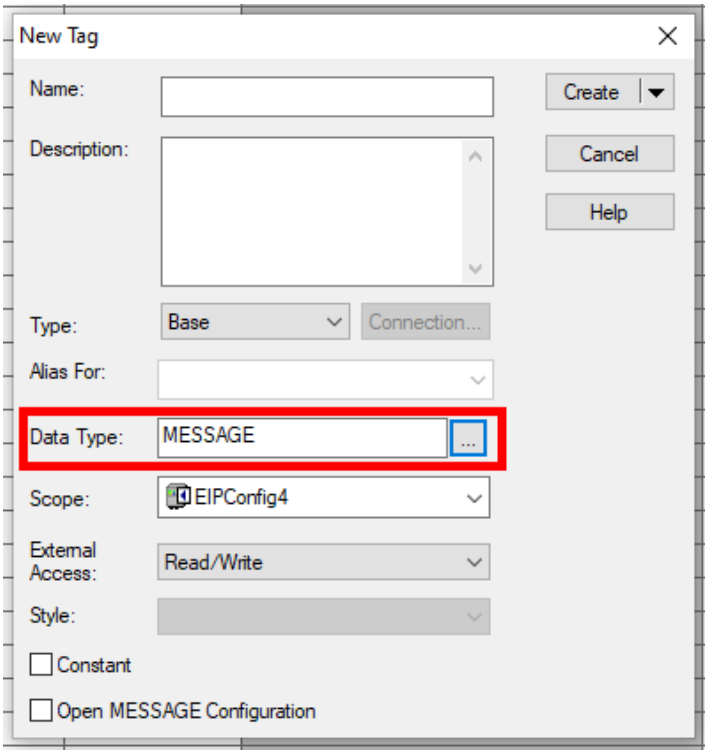
1. In Studio 5000, in the Controller Organizer, expand Controller Tags by double-clicking it.



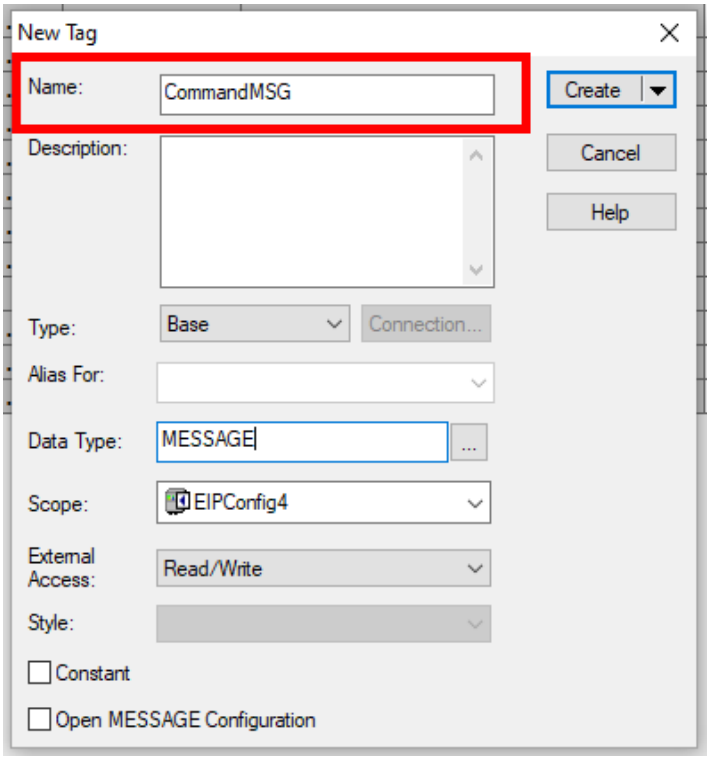
2. Right-click in the middle of the screen and select **New Tag** from the context menu.



3. Change the data type to MESSAGE.

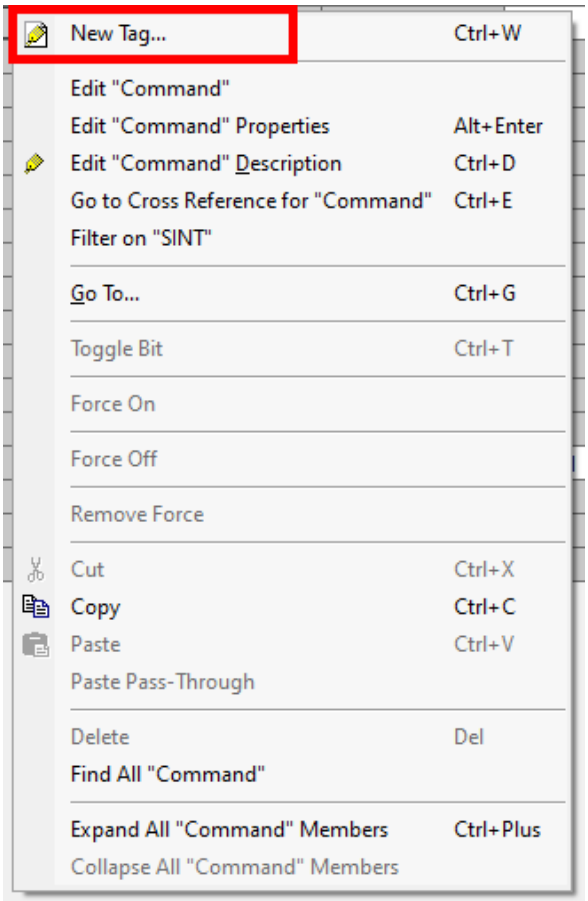


4. Name the tag and click **Create**.

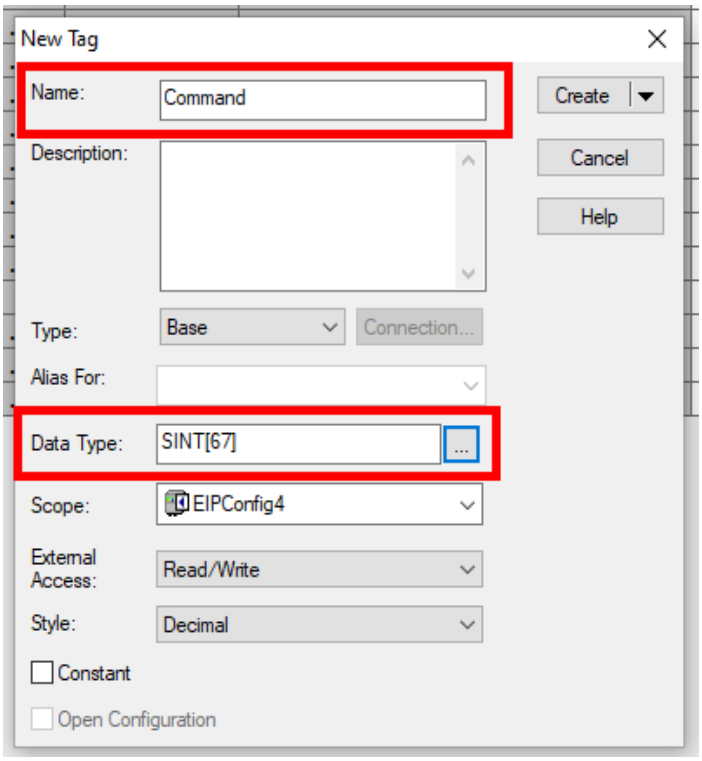


This creates a block to store parameters for sending data to the GoPxL device.

5. Right-click in the middle of the screen again and choose **New Tag** from the context menu.

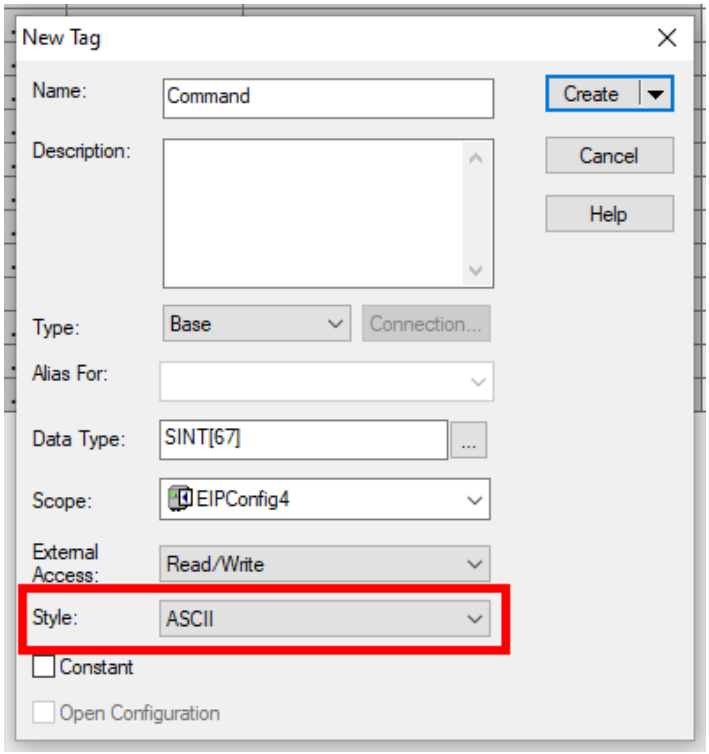


6. Change the data type to SINT[32] and name the tag.



7. Set Style to one of the following:

   If you will be loading job files in GoPxL over the protocol, change **Style** from the default to **ASCII**. This will make editing the command assembly easier later.



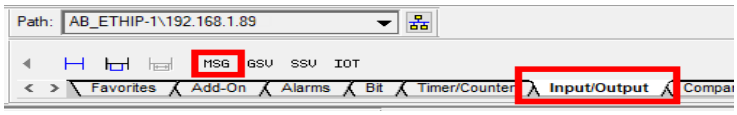
   If you will only be starting or stopping the sensor, leave **Style** at the default setting of **Decimal**.

8. Click **Create**.

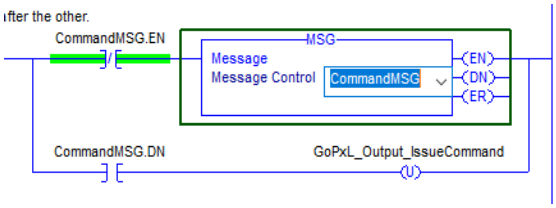
   This creates a tag to store the command data before sending it.

9. In the ladder, go to the Input/Output function blocks and click MSG to add a Message function block.

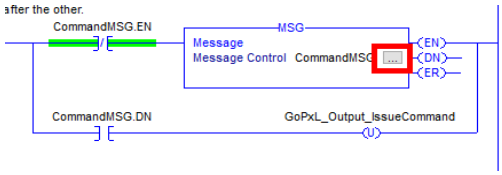
   You may need to add a new rung to allow this.



10. Once the new MSG function block has been added, click the tag dropdown and select the MSG tag you created earlier.

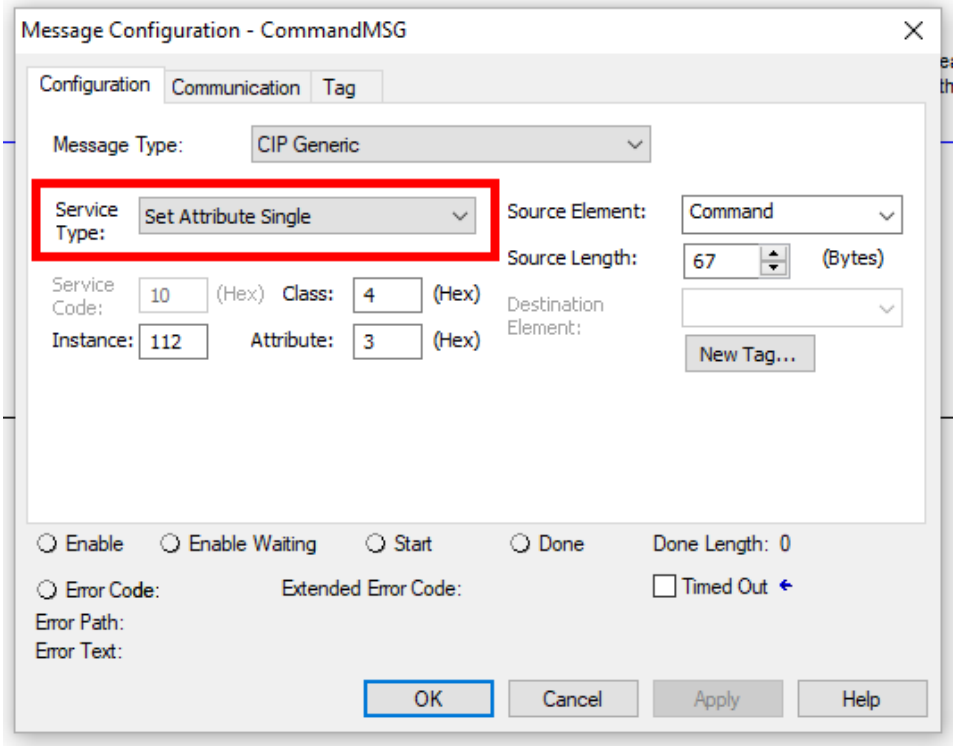


11. Click the grey box to open the Configuration Dialog box

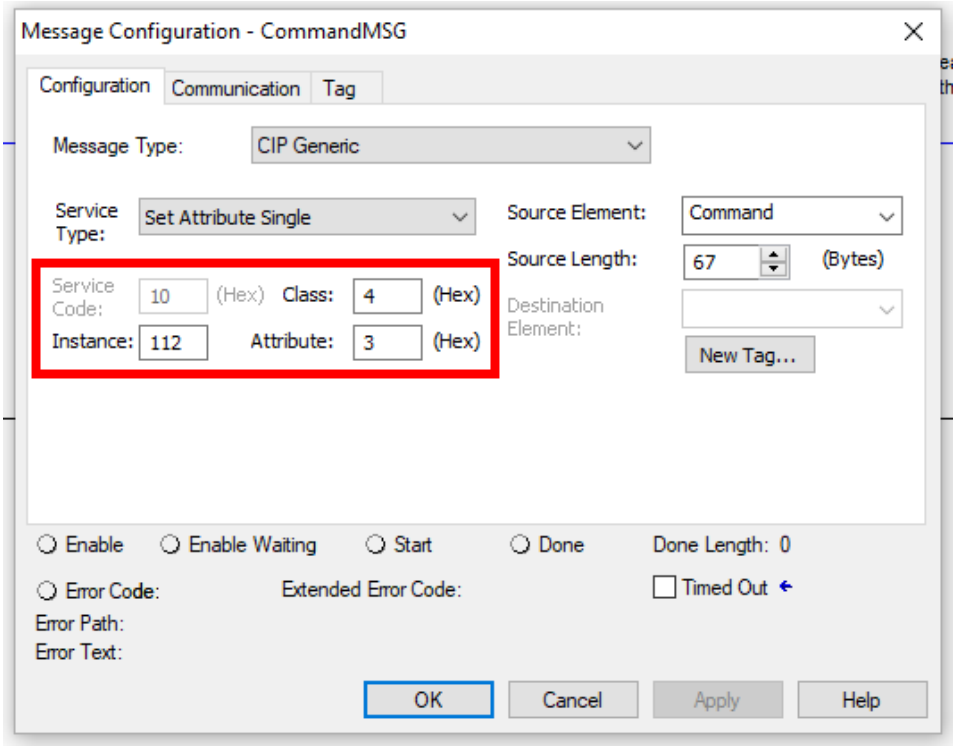


12. In the Message Configuration dialog, choose the **Set Attribute Single** function from the **Service Type** drop-down.

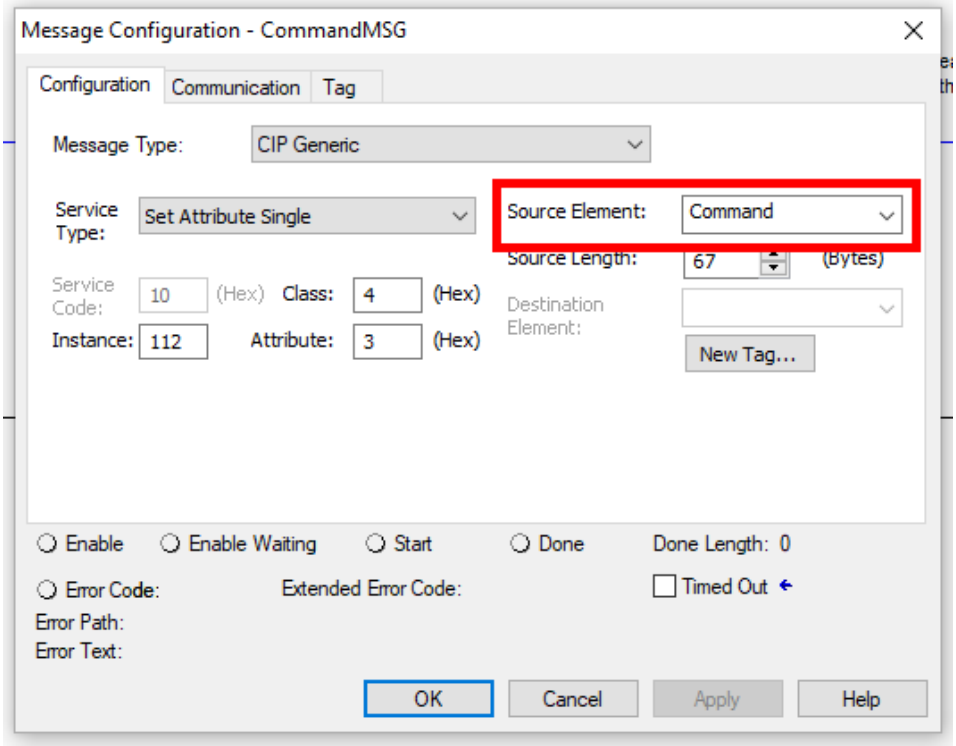
    This will auto populate the Service Code hex character.



13. Enter 4 for Class, 112 for Instance, and 3 for Attribute to set the sensor's command assembly.

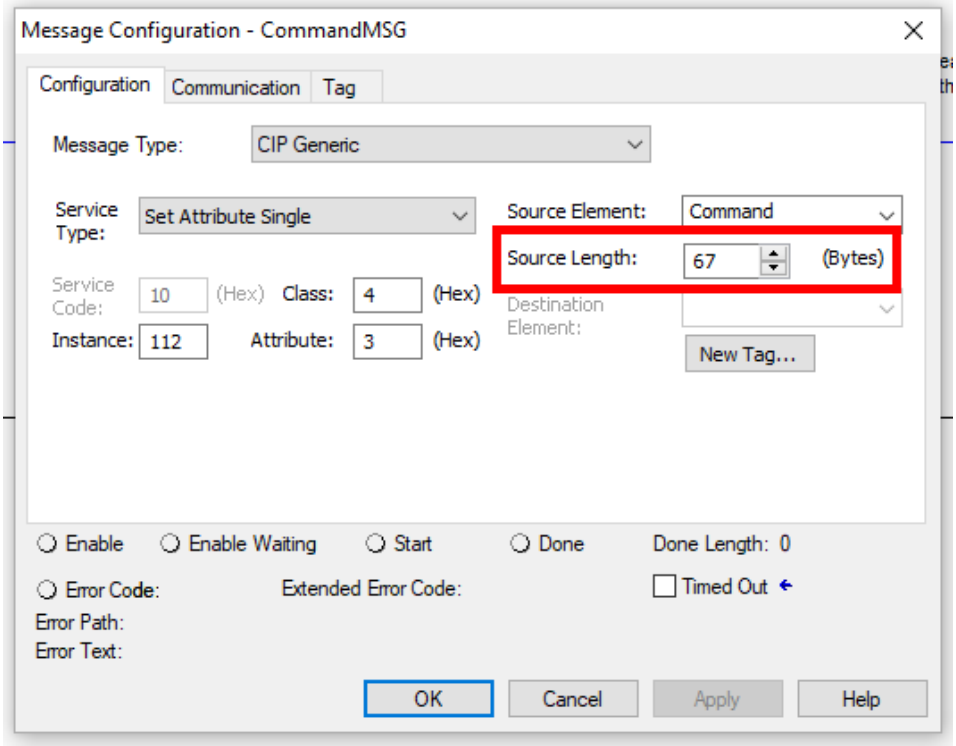


14. Select the SINT[32] tag you created to store the command assembly from **Destination Element**.

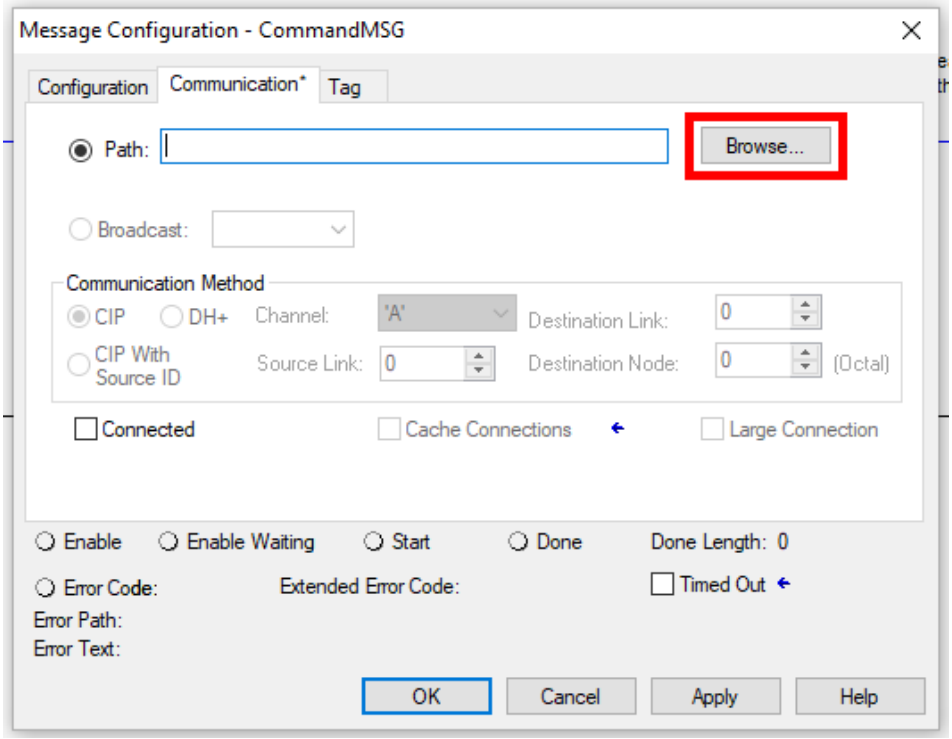


15. Make sure that the length is set to 67 bytes so that the entire command assembly is transmitted.

    A partial transmission may result in an unexecuted command.

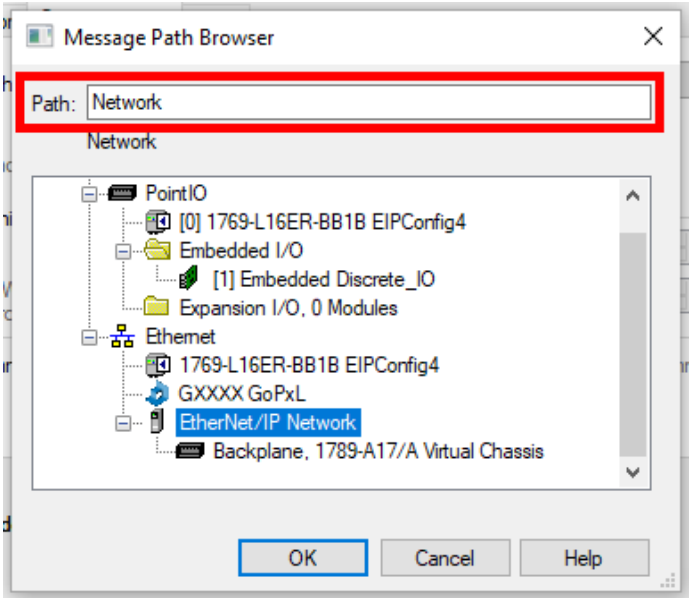


16. On the **Communication** tab, click **Browse**



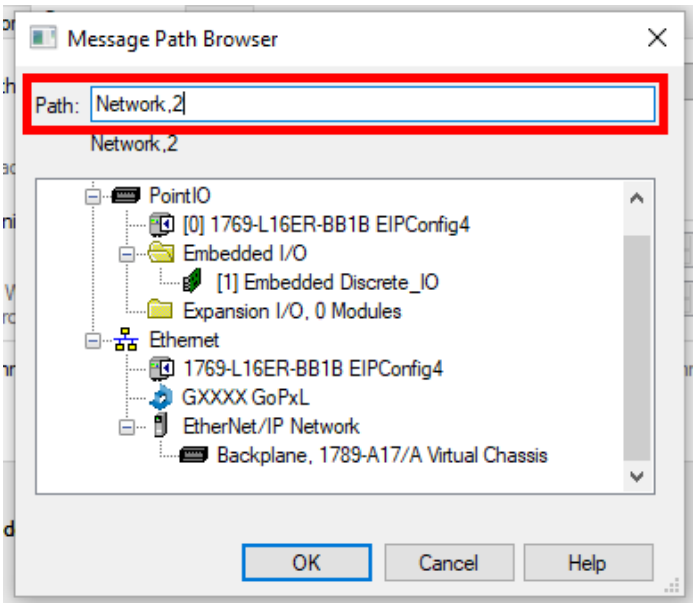
17. Click the **EtherNet/IP Network** node.

    This will route communication messages to the EtherNet/IP network.



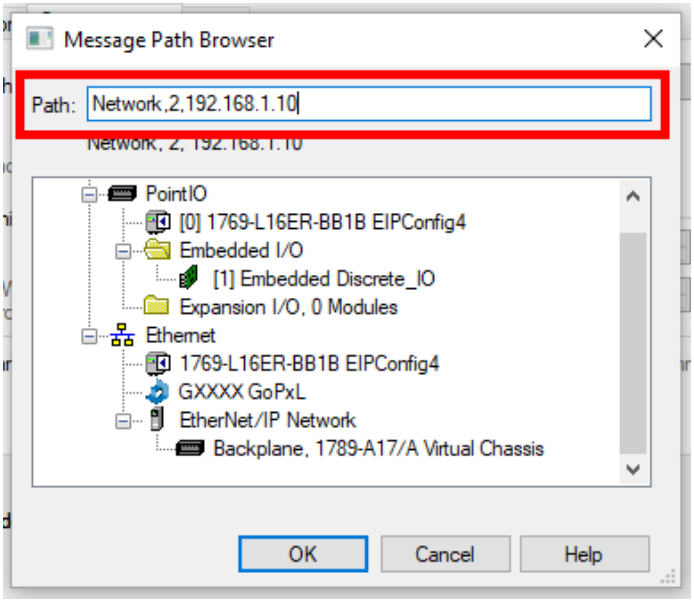
18. Add the Ethernet port that is physically connected to the GoPxL device.

    This will add the specific port address to your communication path.

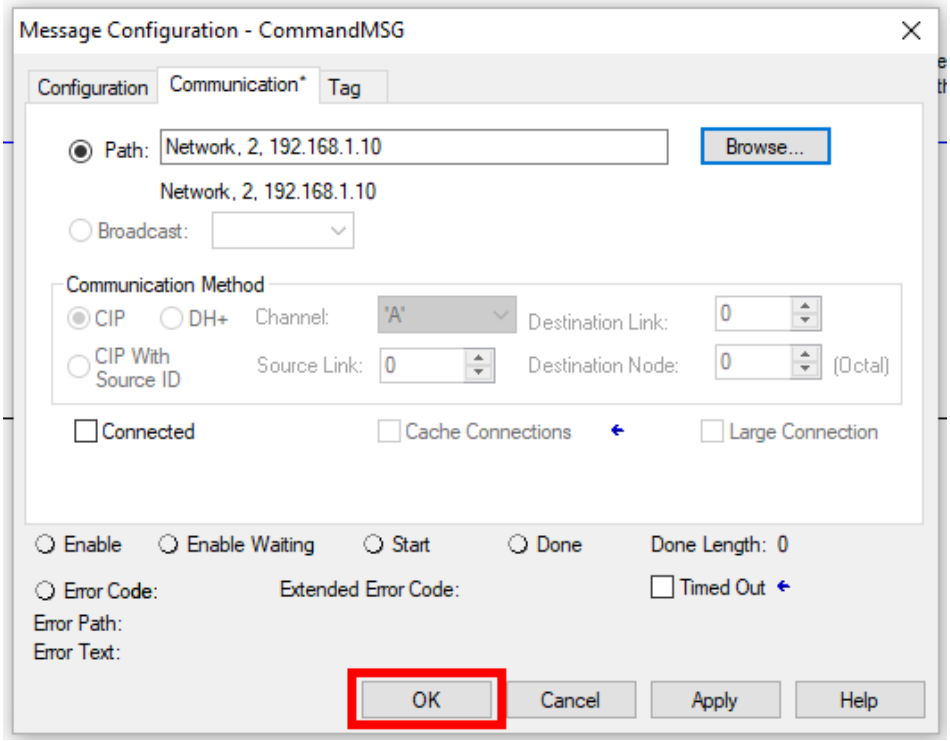


19. Type the IP address of the Gocator to complete the path.

    It is important to double-check that the network, port, and IP address are separated by commas in the form "networkname,port,IPaddress".



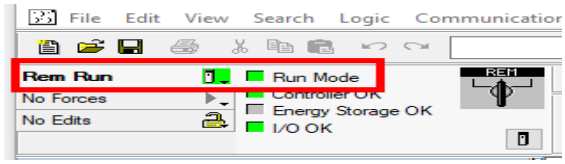
20. Click **OK**.
21. Click **OK** to exit the Message Configuration dialog.



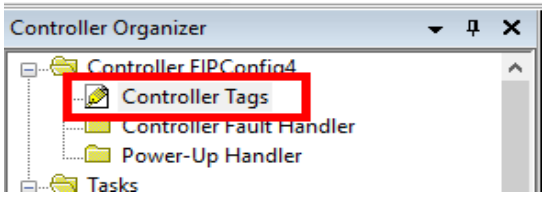
To start a sensor over explicit messaging, the Command assembly must be correctly modified for the integer-based command byte.

Do the following:

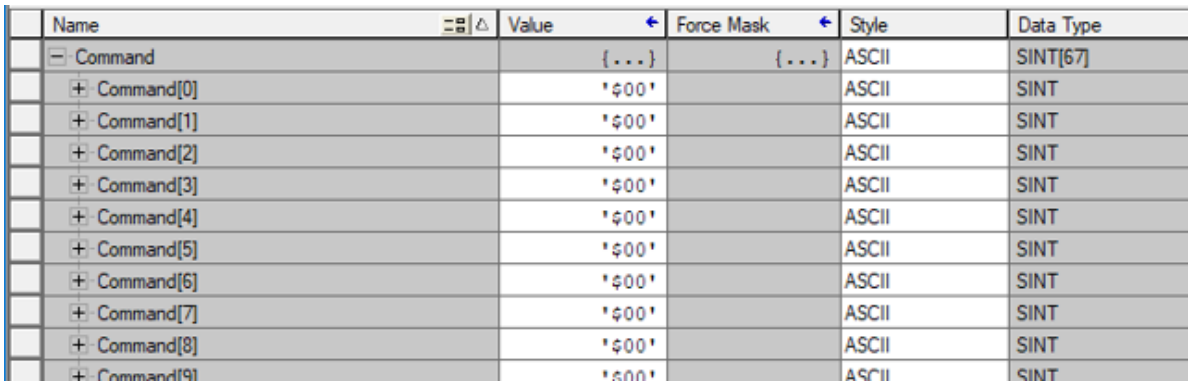
1. Make sure that you have downloaded your PLC program to the controller and that your controller is in Run Mode.



2. Expand Controller Tags by double-clicking it.

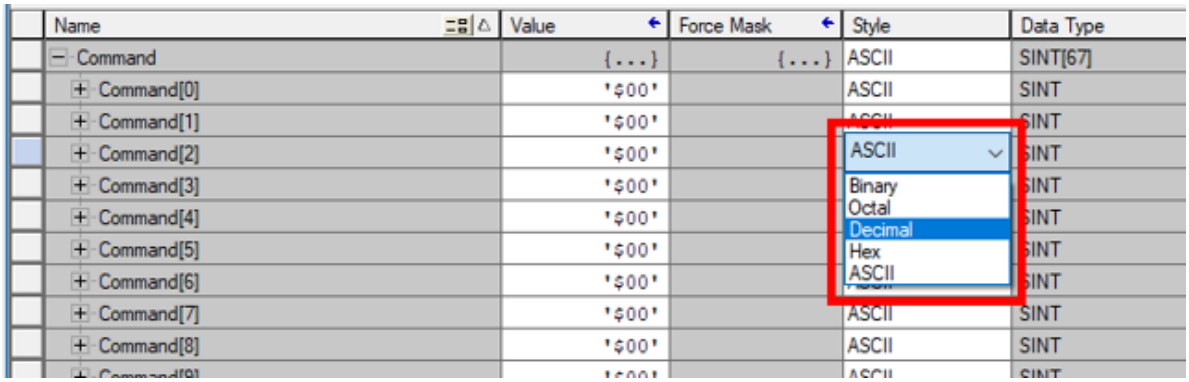


3. Expand the Command assembly tag.

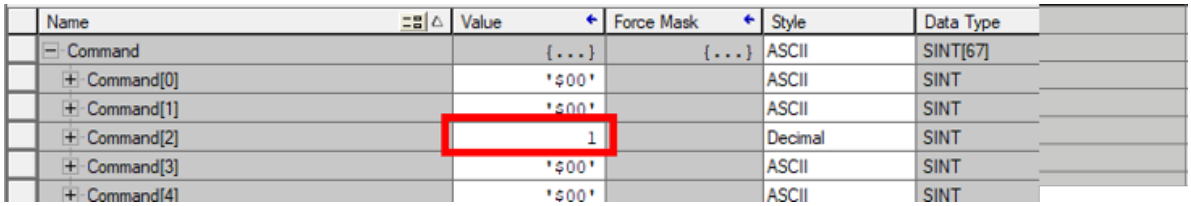


Name	Value	Force Mask	Style	Data Type
Command	{...}	{...}	ASCII	SINT[67]
Command[0]	'$00'		ASCII	SINT
Command[1]	'$00'		ASCII	SINT
Command[2]	'$00'		ASCII	SINT
Command[3]	'$00'		ASCII	SINT
Command[4]	'$00'		ASCII	SINT
Command[5]	'$00'		ASCII	SINT
Command[6]	'$00'		ASCII	SINT
Command[7]	'$00'		ASCII	SINT
Command[8]	'$00'		ASCII	SINT

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 identifiers are sent as integer-based values.



4. Type the number 1 into the value field of Command[2].



Name	Value	Force Mask	Style	Data Type
Command	{...}	{...}	ASCII	SINT[67]
Command[0]	'$00'		ASCII	SINT
Command[1]	'$00'		ASCII	SINT
Command[2]	1		Decimal	SINT
Command[3]	'$00'		ASCII	SINT
Command[4]	'$00'		ASCII	SINT

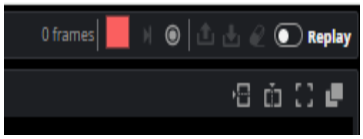
5. After executing the MSG instruction that we set up earlier, in a web browser, type in the sensor IP address to the URL bar.



This should load the web GUI.

6. Verify that the sensor started.

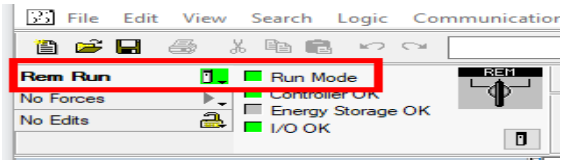
If the Run button is a red square, then the sensor was successfully started.



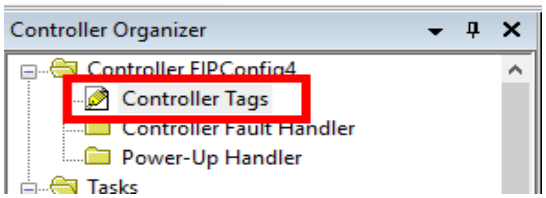
Your ladder logic should only be able to edit the Command assembly 1 time. Since Explicit Message Buffering is checked from the Gocator setup, multiple message transfers from improper ladder logic will end up buffering on the Gocator side of the network. The only way to easily clear the messaging buffer is to power cycle the sensor.

LOADING A SENSOR JOB FILE

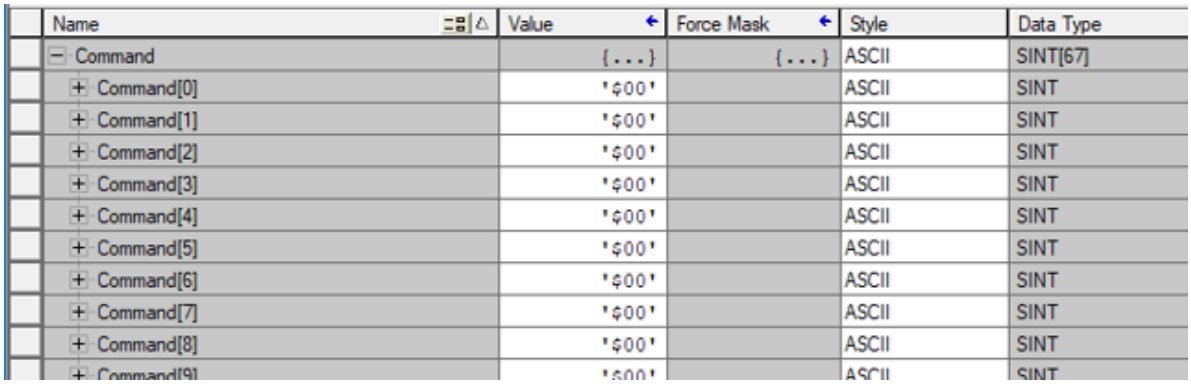
1. Make sure that you have downloaded your PLC program to the controller and that your controller is in Run Mode



2. Expand **Controller Tags** by double-clicking it

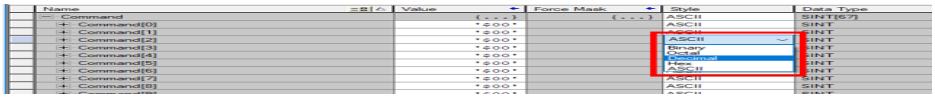


3. Expand the Command assembly tag



Name	Value	Force Mask	Style	Data Type
Command	{...}	{...}	ASCII	SINT[67]
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

If you changed the formatting of the Command tag array to ASCII, then change the display of only the first byte, Command[2], back to Decimal as the control command are sent as integer-based values.



4. If `job1.gpjob` is the job file to be loaded on the sensor and it is not currently running, type each of the four characters making up the filename, excluding the extension (`.gpjob`) into Command[3] through Command [6] of the Command assembly.

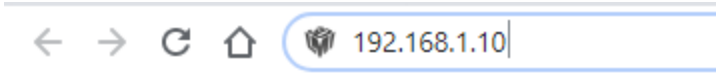
   The ASCII character inputs here are case sensitive. All non-jobname characters must be null or empty values. If the style was changed to ASCII as the default during the tag creation, this will be done already, and the alphanumeric characters can be directly typed into the value column of the bytes.

Name	Value	Force Mask	Style	Data Type
Command	{...}	{...}	ASCII	SINT[67]
Command[0]	'$00'		ASCII	SINT
Command[1]	'$00'		ASCII	SINT
Command[2]	0		Decimal	SINT
Command[3]	'j'		ASCII	SINT
Command[4]	'o'		ASCII	SINT
Command[5]	'b'		ASCII	SINT
Command[6]	'1'		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

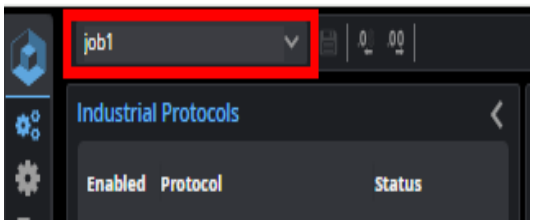
5. Type the integer value 5 into the Command byte to transmit the job name for loading.

Name	Value	Force Mask	Style	Data Type
Command	{...}	{...}	ASCII	SINT[67]
Command[0]	'$00'		ASCII	SINT
Command[1]	'$00'		ASCII	SINT
Command[2]	5		Decimal	SINT
Command[3]	'j'		ASCII	SINT
Command[4]	'o'		ASCII	SINT
Command[5]	'b'		ASCII	SINT
Command[6]	'1'		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

6. In a web browser, the GoPxL device IP address to the URL field.



7. Once the web GUI loads, verify that the job was loaded on the GoPxL device by looking at the job name box.



Ethernet ASCII Protocol

You can use a Programmable Logic Controller (PLC) that supports the Ethernet ASCII protocol to operate a sensor and receive values, such as measurements, stamps, or strings, and decisions (together called results).

You can send data from a PLC to GoPxL after adding and configuring an instance of the User Data Input tool; for more information on the tool, see *User Data Input Tool* on page 1015.

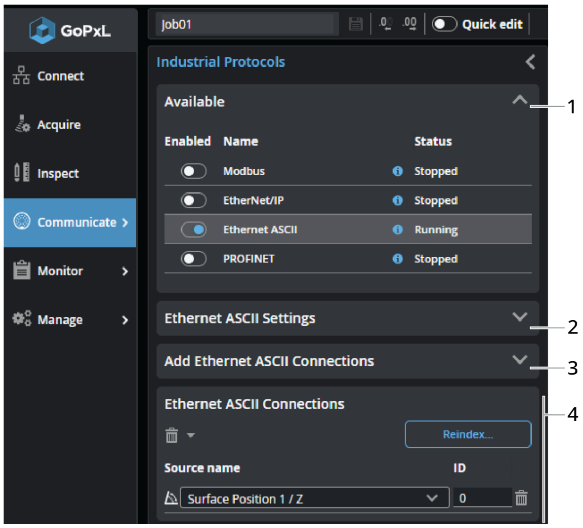
To send data over Ethernet ASCII to the User Data Input tool, set `setdata` or `setdatastring` to update the shared User Data Input buffer and confirm with OK/ERROR. For more information, see *Commands and Formats* on page 1113.

Two modes are available: asynchronous and polling.

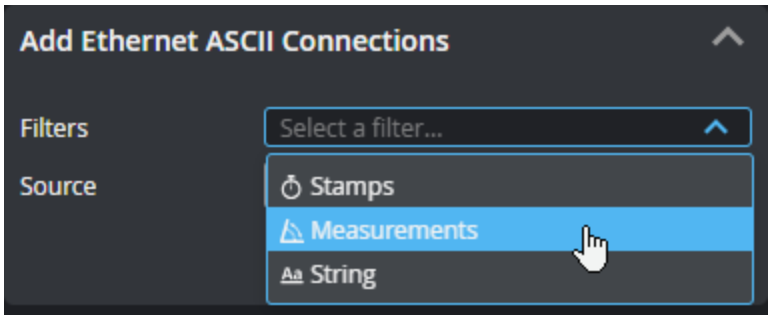
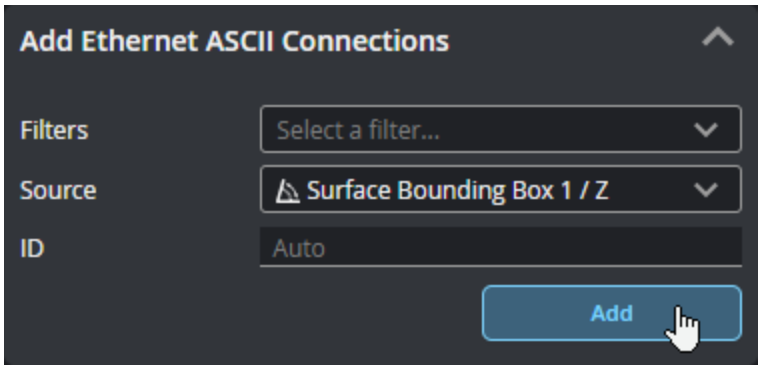
When the sensor is in asynchronous mode, results are sent automatically, as soon as they are available. Data is still available synchronously (that is polled) using the supported polling commands.

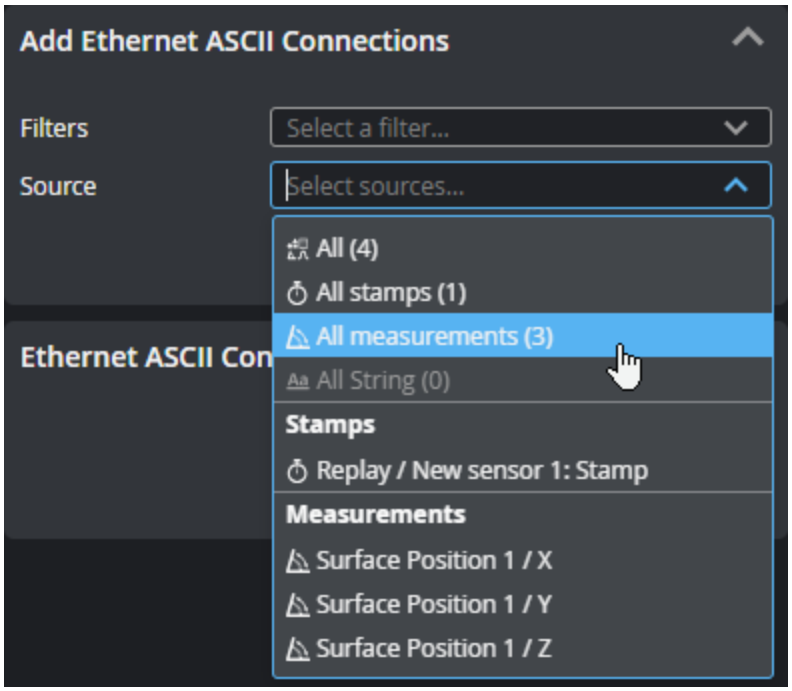
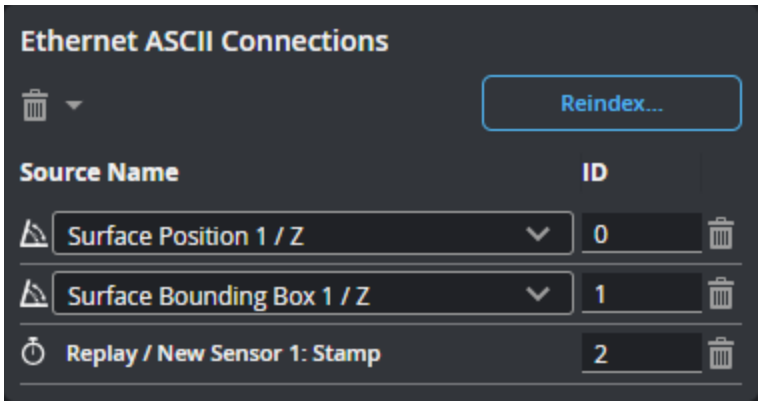
When the sensor is in polling mode (asynchronous mode is disabled), you must use commands on the PLC to operate the sensor and to retrieve results. Ethernet ASCII protocol in polling mode only supports a subset of the tasks that can be accomplished in the web interface, and only measurement results can be transmitted to the PLC (scan data is not sent).

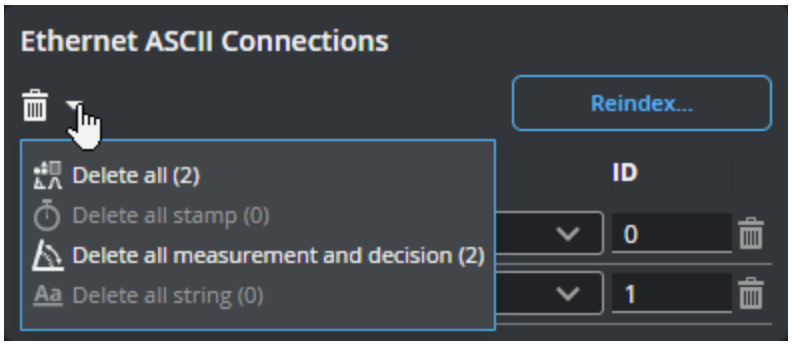
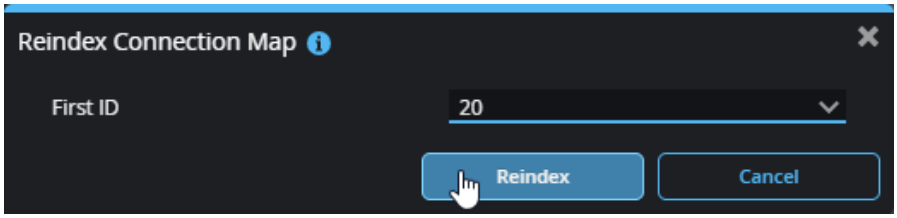
You configure the protocol on the **Communicate** > **Industrial** page. After configuring the ASCII protocol parameters in GoPxL (see below), see *Ethernet ASCII Protocol Format - PLC Information* on page 1112 for information on the available polling commands.



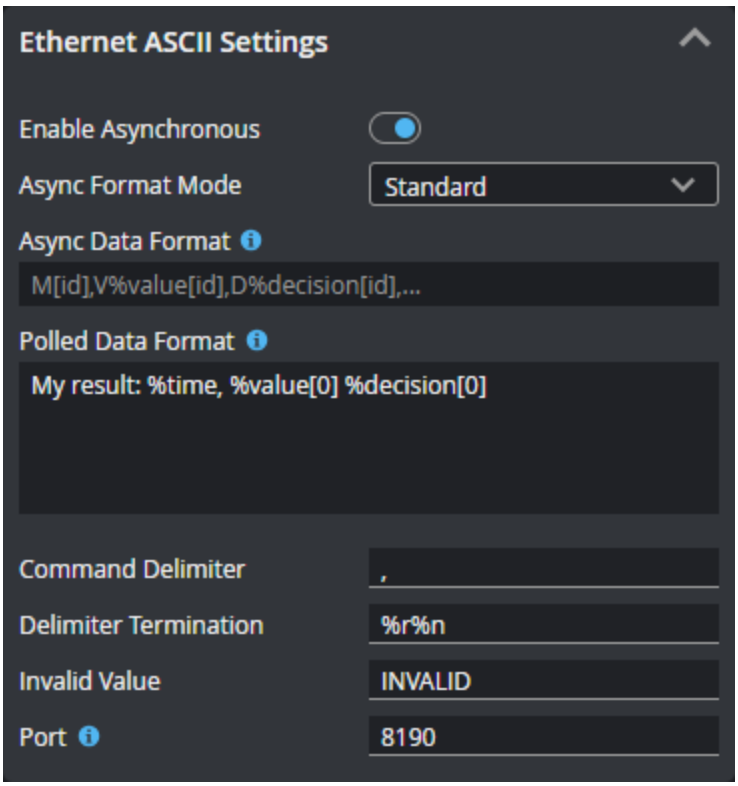
	Element	Description
1	Available protocols	Click a row in the table to choose which protocol to configure. Enable a protocol by toggling the slider to the right.
2	Ethernet ASCII Settings	Lets you configure the Ethernet ASCII protocol. For descriptions of the settings, see *Ethernet ASCII settings* on page 1111.

	Element	Description
3	Add Ethernet ASCII Connections	Lets you add sources (such as stamps or measurements) in the connection map. Added connections appear in the **Ethernet ASCII Connections** at the bottom of the panel. If you have many sources, you can filter them by type to make it easier to find the source you want in the **Source** drop-down. You can select more than one filter at once.  Choose the source in the **Source** drop-down. If you leave **Address** as "Auto", GoPxL automatically increments the value; otherwise, you can set it yourself. By default, connections are added directly after the last source in the connections map. You can optionally set the address at which a source is added using the **Address** parameter.  You can also add all sources, or all sources of a given type.

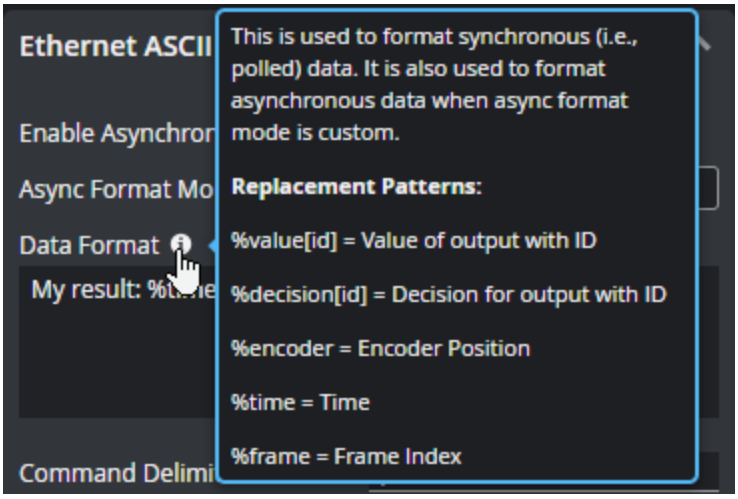
	Element	Description
		 Click **Add** to add the source to the EtherNet/IP Connections table below.
4	Ethernet ASCII Connections	The connections listed here will be sent by the protocol to a compatible PLC. You can delete a connection using the trashcan icon in its row, or change its ID.  You can also use shortcut commands to delete all of the connections, or specific categories of connections.

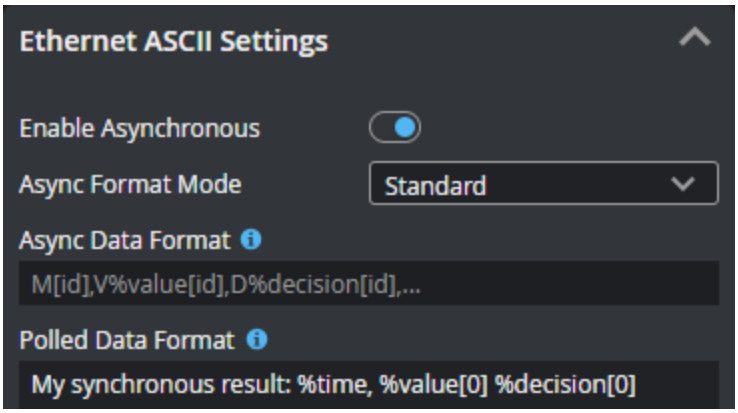
Element	Description
	 Use **Reindex...** to remove gaps of unused addresses between connections. You can also use **Reindex...** to set the value of the first index. 

The Ethernet ASCII protocol provides several settings you must configure. The settings that are displayed change depending on the operation mode (polling vs. asynchronous) and the format mode.



Ethernet ASCII settings

Parameter	Description
Enable Asynchronous	Toggles asynchronous operation. In asynchronous mode, the data results are transmitted when they are available. Choose the **Async Format Mode** and, if you chose Custom in **Async Data Format**, set the format. (These parameters are only displayed when asynchronous mode is enabled.) Note that when asynchronous mode is enabled, data is still available using polling commands. When this option is disabled, the protocol only runs in polling (or synchronous) mode, and you must send commands on the data channel to request the latest result.
Async Format Mode	When you enable asynchronous operation, you must choose the format mode. The format mode is one of the following: Standard, Standard with Stamp, and Custom. For more information on the formats, see below.
Async Data Format and Polled Data Format Data Format	These fields contain the formats the protocol uses for the data it sends. GoPxL displays different "Format" fields depending on whether you enable asynchronous operation and what you set **Async Format Mode** to. For details on the supported replacement patterns you can use in the formats, click on the info icon next to the field's name:  When asynchronous operation is disabled, you configure the format in **Data Format**. When asynchronous operation is enabled and you choose either of the "Standard" modes, the format for asynchronous data is displayed in the read-only **Async Data Format** field. For polled data (which is always available), you can set a different format under **Polled Data Format**.

Parameter	Description
	 In asynchronous mode, when you set the mode to Custom, you set the format for both asynchronous and polling data in the **Data Format** field. 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.
Command Delimiter	Sets the command delimiter character.
Delimiter Termination	Sets the command termination character.
Invalid Value	Sets the invalid value characters.
Control Port	Sets the port. This is the port to which you PLC must connect.

Ethernet ASCII Protocol Format - PLC Information

You can use a programmable logic controller (PLC) that supports the Ethernet ASCII protocol to operate a sensor and receive values, such as measurements, stamps, or strings, and decisions. To use the ASCII protocol, it must be enabled and configured in the active job. By default, the control port used to connect to the sensor is 8190. For information on enabling Ethernet ASCII, as well as changing the control port and other protocol configurations in GoPxL, see *Ethernet ASCII Protocol* on page 1107.

Two modes are available: asynchronous and polling.

When the sensor is in asynchronous mode, results are sent automatically, as soon as they are available. No commands are required, but the client can still send a command to the sensor to receive output data or to control the device.

When the sensor is in polling mode, you use commands to operate the sensor and to retrieve results. Ethernet ASCII protocol in polling mode only supports a subset of the tasks that can be accomplished in the web interface, and only measurement results can be transmitted to the PLC (scan data is not sent).

This section describes the following aspects of the Ethernet ASCII protocol:

- the polling operation commands you can use to control a sensor and retrieve results from it
- result formats

The protocol communicates using ASCII strings. The output result format from the sensor is user-configurable.

For information on the Ethernet ASCII protocol format, see *Commands and Formats* below.

Commands and Formats

Commands are sent from the client to the sensor. The commands are not case sensitive. However, the `readprop` command's metrics path argument *is* 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.

You can change the delimiter and termination characters in the **Special Character** settings in the GoPxL interface; for more information, see *Ethernet ASCII Protocol* on page 1107.



Measurement IDs come from the **ID** field in the connection map. Be sure to add measurements to the connection map and make note of the IDs of the measurements you need to add to the data format.

Client Commands

For the commands that write to the shared user data input buffer, see *Set Data Commands* on page 1116.

Ethernet ASCII client commands

Command	Arguments	Description
start	-	Starts the sensor.
stop	-	Stops the sensor.
trigger	-	Software Trigger.
loadjob	Job name	Load the specified job file, up to 63 character long, including the last null terminating character.
result	-	Return the measurement or string data specified in the custom

Command	Arguments	Description
		format string. For a description of the measurement value (prefixed with the letter "V") and decision (prefixed with the letter "D") format, see *Measurement Output Format* on page 1125 and *String Output Format* on page 1125.
	One or more measurement or string IDs, separated by command delimiter.	Return the measurement or string data for each measurement or string ID in the following format string: M[ID],V[VALUE],D[DECISION],... Note that string data will not include a decision. Example: result,0,1 Output: OK,M0,V-5000,D0,S1,VsampleString
measurement	One or more measurement IDs, separated by command delimiter.	Return the measurement data for each measurement ID in the following format string: M[ID],V[VALUE],D[DECISION],... For a description of the measurement value (prefixed with the letter "V") and decision (prefixed with the letter "D") format, see *Measurement Output Format* on page 1125. Example: measurement,0,1 Output: OK,M0,V-5000,D0,M1,V5000,D1
string	One or more string IDs, separated by command delimiter.	Return the string data for each string id in the following format string: S[ID],V[VALUE],,... For a description of the string value (prefixed with the letter "V"), see *String Output Format* on page 1125. Example: string,0,1 Output: OK,S0,VsampleString,S1,V2ndString
value	One or more measurement or string IDs, separated by command delimiter.	Return the measurement or string value for each measurement or string ID in the following format string: M/S[ID],V[VALUE],... For a description of the measurement/string value (prefixed with the letter "V"), see *Measurement Output Format* on page 1125 and *String Output Format* on page 1125. Example: value,0,1 Output: OK,M0,V-5000,S1,VsampleString
decision	One or more measurement IDs, separated by command delimiter.	Return the measurement decision for each measurement ID in the following format string: M[ID],D[DECISION],... For a description of the measurement decision (prefixed with the letter "D") format, see *Measurement Output Format* on page 1125. Example: decision,0,1 Output: OK,M0,D0,M1,D1
stamp	-	Return the time, encoder and frame index stamp values for the first stamp ID found.

Command	Arguments	Description
For information on the interpretation of time, see *Timetick Interpretation* on page 1119.		Order of values returned is: Time, Encoder, Frame. . Example: stamp Output: OK,Time,381497381349,Encoder,0,Frame,18,
	One or more of these arguments in any order, separated by the command delimiter: "time" "encoder" "frame"	Return the specified stamp value(s) in the order of the provided arguments. Example: stamp,time,frame Output: OK,381497381349,18
	A stamp ID.	Return the time, encoder, and frame index stamp values for the passed stamp ID. Example: stamp,1
time	One or more stamp IDs, separated by command delimiter.	Return the stamp's time value for each stamp ID. The timetick of the last frame. For information on the interpretation of time, see *Timetick Interpretation* on page 1119. Example: time,1 Output: OK,381497381349
encoder	One or more stamp IDs, separated by command delimiter.	Return the stamp's encoder value for each stamp ID. Example: encoder,1 Output: OK,0
frame	One or more stamp IDs, separated by command delimiter.	Return the stamp's frame index stamp value for each stamp ID. Example: frame,1 Output: OK,18
align	-	Performs the alignment operation.
clearalign	-	Clears sensor alignment.
readprop	One or more strings representing a resource whose property to return, separated by command delimiter.	Returns the resource JSON object specified by the resource path. It is possible to use JSON pointers to retrieve a specific value within the JSON document. Example #1: readprop,/system Output: OK,{"_links":{"self":{"href":"/system"}},"_schema":{"properties":{"autoLiveSimEnabled":{"sortOrder":200,"title":"autoLiveSimEnabled","type":"boolean"},"autoStart":{"sortOrder":100,"title":"autoStart","type":"boolean"},"runState":{"enum":[0,1,2],"enumText":

Command	Arguments	Description
		["Ready","Running","Conflict"],"readOnly":true,"sortOrder":0,"title":"runState","type":"integer"}},"sortOrder":0,"title":"","type":"object"},"autoLiveSimEnabled":true,"autoStart":true,"runState":1}, Example #2: readprop,/system#/runState,/system#/autostart Output: OK,1, true, Example #3: readprop,/jobs#/loadedJob Output: OK,"Job02", Example #4: readprop,/jobs#/loadedJobDirty Output: OK,false,

Set Data Commands

The Ethernet ASCII protocol provides "setdata" commands for writing various sizes of data to the shared user data input buffer at a specified offset.

Two types of commands are available: commands that write signed or unsigned integer or float values of various sizes and types, and a single command that writes an ASCII byte array to the buffer.

Numerical values

To use these commands (for example, `setdatau8`, `setdatai8`, `setdataf32`, and so on), you must specify an offset in the buffer, followed by the value.

The maximum offset is determined by the size of the value type.

The values can only be a base-10 number. Other formats such as scientific or hex notations result in undefined behavior. For integer values, any decimal portion of the input is ignored; for floating-point values, the decimal portion is optional.

See the table below for the type and maximum offset for each command.

String value

The `setdatastring` writes an ASCII byte array to the buffer. For more information, see *setdatastring* on page 1119.

In the table below, "UDI buffer" refers to the "shared user data input buffer."

Ethernet ASCII set data client commands

Command	Arguments	Description
setdatau8	2 arguments in this order: "offset" "value" separated by command delimiter.	Writes a 1-byte unsigned integer value into the UDI buffer at the specified offset. The offset argument can be any integer between 0 and 999, inclusive. The value must be a base-10 1-byte unsigned integer. Example: setdatau8,10,100 Output: OK
setdatai8		Writes a 1-byte signed integer value into the UDI buffer at the specified offset. The offset can be any integer between 0 and 999, inclusive. The value must be a base-10 1-byte signed integer. Example: setdatai8,10,-100 Output: OK
setdatau16		Writes a 2-byte unsigned integer value into the UDI buffer at the specified offset. The offset can be any integer between 0 and 998, inclusive. The value must be a base-10 2-byte unsigned integer. Example: setdatau16,10,10000 Output: OK
setdatai16		Writes a 2-byte signed integer value into the UDI buffer at the specified offset. The offset can be any integer between 0 and 998, inclusive. The value must be a base-10 2-byte signed integer. Example: setdatai16,10,-10000 Output: OK
setdatau32		Writes a 4-byte unsigned integer value into the UDI buffer at the specified offset.

Command	Arguments	Description
		The offset can be any integer between 0 and 996, inclusive. The value must be a base-10 4-byte unsigned integer. Example: setdatau32,10,1000000 Output: OK
setdatai32		Writes a 4-byte signed integer value into the UDI buffer at the specified offset. The offset can be any integer between 0 and 996, inclusive. The value must be a base-10 4-byte signed integer. Example: setdatai32,10,-1000000 Output: OK
setdatau64		Writes an 8-byte unsigned integer value into the UDI buffer at the specified offset. The offset can be any integer between 0 and 992, inclusive. The value must be a base-10 8-byte unsigned integer. Example: setdatau64,10,100000000 Output: OK
setdatai64		Writes an 8-byte signed integer value into the UDI buffer at the specified offset. The offset can be any integer between 0 and 992, inclusive. The value must be a base-10 8-byte signed integer. Example: setdatai64,10,-100000000 Output: OK
setdataf32		Writes a 4-byte single-precision floating-point value into the UDI buffer at the specified offset. The offset can be any integer between 0 and 996, inclusive. The value must be a base-10 floating point number. Decimal portion of the

Command	Arguments	Description
		input is optional. Example: setdataf32,10,1000.1234 Output: OK
setdataf64		Writes an 8-byte double-precision floating-point value into the UDI buffer at the specified offset. The offset can be any integer between 0 and 992, inclusive. The value must be a base-10 floating point number.Decimal portion of the input is optional. Example: setdataf64,10,1000.123456 Output: OK
setdatastring	3 arguments in this order: "offset" "max length" "string" separated by command delimiter.	Writes an ASCII byte array into the UDI buffer. The sum of offset + max length cannot exceed 1000. The string must not be null terminated. If the string is longer than the max length value, it will be truncated to max length. If the string is shorter than max length value, remaining bytes up to the max length will be padded with 0x00. Example: setdatastring,10,5,ABCDE Output: OK

TIMETICK INTERPRETATION

Timetick is a 64-bit counter that increases with time, so each scan from the same device has a unique timetick value.

On Gocator 3D sensors, to convert this value to microseconds, use the following:

```
time (µs) = timetick / 1.024.
```

ERRORS FOR EACH COMMAND

The following table lists the possible errors for each command.

Ethernet ASCII error handling

Command	Arguments	Description
start	ERROR,Could not start the sensor	
stop	ERROR,Could not stop the sensor	
trigger	ERROR,Could not trigger	
loadjob	ERROR,Job name required.	Job name has to be provided as a command argument.
	ERROR,Failed to load job <job name>.gpjob	
result	ERROR,There is no measurement data to output. Please confirm that the sensor is running	Could not find any measurements. Make sure protocol's connection map is not empty and the sensor is running.
	ERROR,Invalid parameter. Please verify your input	Provided command argument was invalid. It must be a number.
	ERROR,Specified measurement ID not found. Please verify your input	Could not find a measurement data for specified ID.
	ERROR,The custom format string is not valid. Please verify your input	Custom format string is invalid.
measurement	ERROR,One or more measurement ID must be provided	Measurement ID has to be provided as an argument.
	ERROR,There is no measurement data to output. Please confirm that the sensor is running	Could not find any measurements. Make sure protocol's connection map is not empty and the sensor is running.
	ERROR,Invalid	Provided command argument was invalid. It must be a number.

Command	Arguments	Description
	parameter. Please verify your input	
	ERROR,Specified measurement ID not found. Please verify your input	Could not find a measurement data for specified ID.
string	ERROR,One or more string ids must be provided.	String ID has to be provided as an argument.
	ERROR,There is no data to output. Please confirm that the sensor is running.	Could not find any data. Make sure protocol's connection map is not empty and the sensor is running.
	ERROR,Invalid parameter. Please verify your input.	Provided command argument was invalid. It has to be a number.
	ERROR,Specified id not found. Please verify your input.	Could not find an output for specified ID.
value	ERROR,One or more measurement/string ids must be provided.	Measurement/string ID has to be provided as an argument.
	ERROR,There is no data to output. Please confirm that the sensor is running.	Could not find any data. Make sure protocol's connection map is not empty and the sensor is running.
	ERROR,Invalid parameter. Please verify your input.	Provided command argument was invalid. It has to be a number.
	ERROR,Connection with id not found.	Could not find data for specified ID.
decision	ERROR,One or more measurement ids must be provided.	Measurement ID has to be provided as an argument.
	ERROR,There is no data to output. Please confirm that	Could not find any data. Make sure protocol's connection map is not empty and the sensor is running.

Command	Arguments	Description
	the sensor is running.	
	ERROR,Invalid parameter. Please verify your input.	Provided command argument was invalid. It has to be a number.
	ERROR,Connection with id not found.	Could not find data for specified ID.
	ERROR,String connection with id does not contain a decision.	A string id was included in the decision command.
	ERROR,Connection with id %u is not a measurement or string.	An id mapping to neither measurement nor string was included in the decision command.
stamp	ERROR,Invalid stamp command format.	Invalid command arguments.
	ERROR,One or more stamp ids must be provided.	Stamp ID has to be provided as an argument.
	ERROR,Invalid parameter. Please verify your input.	Provided stamp ID was invalid. It has to be a number.
	ERROR,Stamp with id not found.	Could not find stamp data for specified ID.
	ERROR,Connection id is not a stamp.	Specified id was not a stamp.
time	ERROR,One or more stamp ids must be provided.	Stamp ID has to be provided as an argument.
	ERROR,Invalid parameter. Please verify your input.	Provided stamp ID was invalid. It has to be a number.
	ERROR,Stamp with id not found.	Could not find stamp data for specified ID.
	ERROR,Connection id is not a stamp.	Specified id was not a stamp.

Command	Arguments	Description
encoder	ERROR,One or more stamp ids must be provided.	Stamp ID has to be provided as an argument.
	ERROR,Invalid parameter. Please verify your input.	Provided stamp ID was invalid. It has to be a number.
	ERROR,Stamp with id not found.	Could not find stamp data for specified ID.
	ERROR,Connection id is not a stamp.	Specified id was not a stamp.
frame	ERROR,One or more stamp ids must be provided.	Stamp ID has to be provided as an argument.
	ERROR,Invalid parameter. Please verify your input.	Provided stamp ID was invalid. It has to be a number.
	ERROR,Stamp with id not found.	Could not find stamp data for specified ID.
	ERROR,Connection id is not a stamp.	Specified id was not a stamp.
align	ERROR,Could not align	
clearalign	ERROR,Could not clear alignment	
readprop	ERROR,String representing a resource must be provided	Resource path must be provided as an argument.
	ERROR,Could not read property	Resource may not exist.
setdatau8 setdatau16 setdatau32 setdatau64 setdatai8 setdatai16 setdatai32	ERROR,Invalid command. Please verify that the command entered is correct and any arguments are separated using the configured delimiter.	Command is missing one or more of the arguments.

Command	Arguments	Description
setdatai64		
setdataf32	ERROR,Invalid parameter. Please verify your input.	One or more of the arguments contains invalid parameter values, specific to the command.
setdataf64		
setdatastring		

Data Output Format Mode

POLLING MODE

Polled data output is displayed using the custom format. You can modify this format in the GoPxL interface. For more information, see *Ethernet ASCII Protocol* on page 1107. Because polling mode is always available, even when asynchronous output is enabled, you can always modify the custom format.

Format Mode	Description
Custom	This format can be changed/edited by the user. Default formatting is: %time, %value[0] %decision[0] It is possible to format the specified measurement or string value according to the format string. This pattern does not apply to measurement decisions. Example: %sprintf[%9.3f, %value[0]] to format the measurement value of measurement id 0 as a floating-point number with 9 digits and 3 decimal places.

ASYNCHRONOUS MODE

When asynchronous output mode is enabled, the data output can be in one of three formats:

Format Mode	Description
Standard	This is a pre-defined format that cannot be changed. The format is available as Read-Only information for reference. Formatting is: M[id],V%value[id],D%decision[id],... Note that string ids will not include decisions.
Standard with Stamp	The format is the Standard format pre-pended with the Time and Encoder information. This is a pre-defined format that cannot be changed. The format is available as Read-Only information for reference. Formatting is: T%time,E%encoder,M[id],V%value[id],D%decision[id],.. Note that string ids will not include decisions.
Custom	This format is the same as the format used by Poll mode and can be changed/edited by the user. Default formatting is: %time, %value[0] %decision[0]

Format Mode	Description
	It is possible to format the specified measurement value according to the format string. This pattern does not apply to measurement decisions. Strings can also be formatted using sprintf. Example: %sprintf[%9.3f, %value[0]] to format the measurement value of measurement id 0 as a floating-point number with 9 digits and 3 decimal places.

MEASUREMENT OUTPUT FORMAT

A measurement contains the fields described below.

Field	Type	Address offset	Description
Value	32s	0	Measurement value. This value is the actual measurement value multiplied by 1000. If the measurement value is invalid, the string "INVALID" is the output for the measurement value.
Decision	8u	4	Measurement decision. Bit 0: 0 = Fail 1 = Pass Bit 1: 0 = Decision value is valid 1 = Decision value is invalid

STRING OUTPUT FORMAT

A string contains the fields described below.

Field	Type	Address offset	Description
Value	string	0	String value. If the string value is invalid, the invalid value parameter is output.

PROFINET Protocol

You can use a PROFINET Programmable Logic Controller (PLC) over Ethernet to operate a sensor and receive values, such as measurements, stamps, or strings. PROFINET only supports a subset of the tasks that can be accomplished in the web interface (for example, starting, stopping, and aligning sensors, and switching Jobs). Scan data can't be sent to the PLC.

You can send data from a PLC to GoPxL after adding and configuring an instance of the User Data Input tool; for more information on the tool, see *User Data Input Tool* on page 1015.

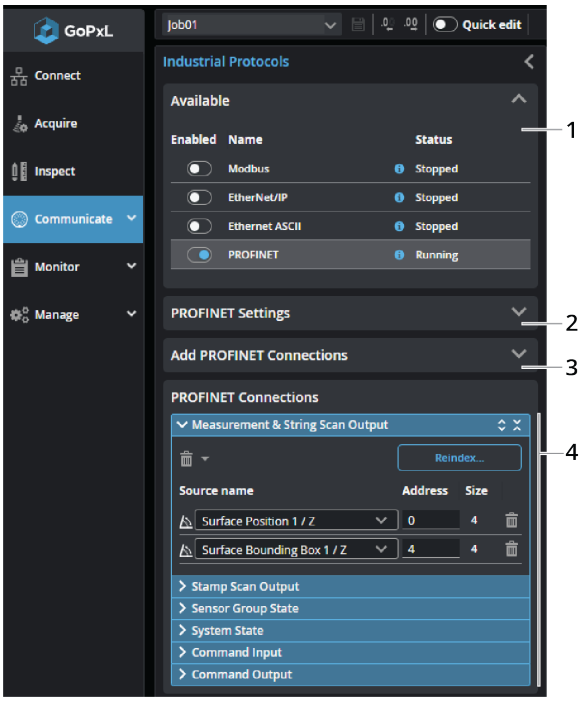
To send data over PROFINET to the User Data Input tool, you must use a special Implicit messaging command (command 9) on the PLC; for more information, see *Connections Map* on page 1130.

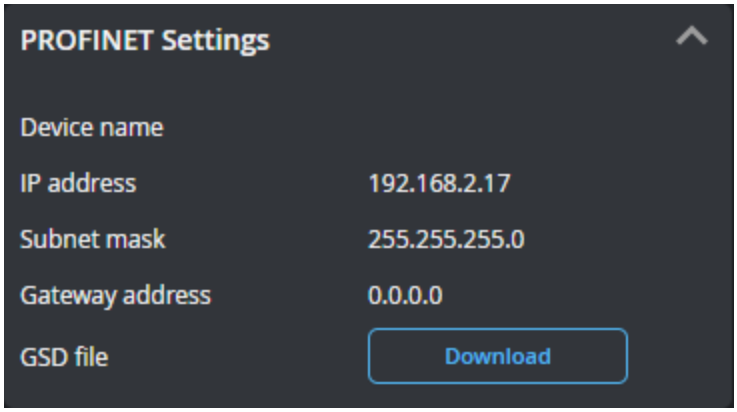
> GoPxL supports PROFINET on sensors, on GoMax ORIN / ORIN+ / NX, and in PC instances of GoPxL.

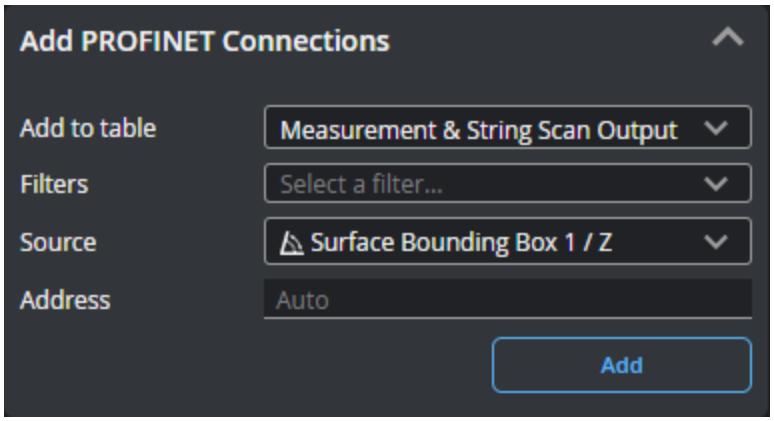
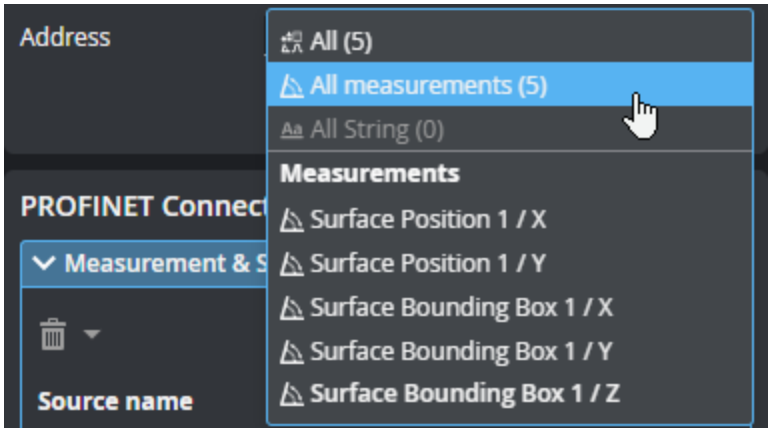
> If you are using a GoMax Smart Vision Accelerator, only connect Ethernet-based PLCs or PCs to the ETH1 port on the GoMax device (whether via a switch or not).

> Due to library limitations, PROFINET cannot be stopped manually after it has started. To stop it, disable the service (an error will be displayed in the log), save your job file, and restart the sensor.

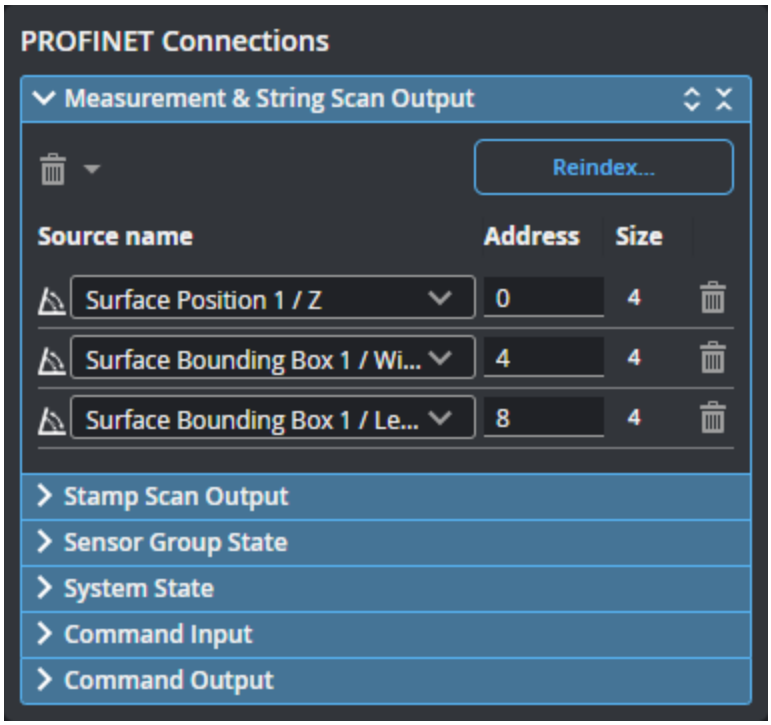
You configure the protocol on the **Communicate** > **Industrial** page. After configuring the PROFINET protocol in GoPxL (see below), use the information in *Connections Map* on page 1130 to configure your PLC.



	Element	Description
1	Available protocols	Click a row in the table to choose which protocol to configure. Enable a protocol by toggling the slider to the right.
2	PROFINET Settings	 **GSD File** Click **Download** to save a GSD file to the client computer for later installation on the PLC. This section also provides network information.
3	Add PROFINET Connections	Lets you add sources (such as stamps, measurements, or strings) to send over the protocol. Added connections appear in the **PROFINET Connections** section at the bottom of the panel. When GoPxL is running on a standalone sensor (one not controlling others), **Add to Table** only contains two options (Measurement Scan Output and Stamp Scan Output), which represent register assemblies. When GoPxL is running on a sensor controlling other sensors, or on a PC or GoMax, **Add to Table** contains an additional entry (Sensor Group State). Use this parameter to choose which register a source is added to. Choose the source in the **Source** drop-down. If you leave **Address** as "Auto", GoPxL automatically increments the value; otherwise, you can set it yourself. By default, connections are added directly after the last source in the connections map. You can optionally set the address at which a source is added using the **Address** parameter.

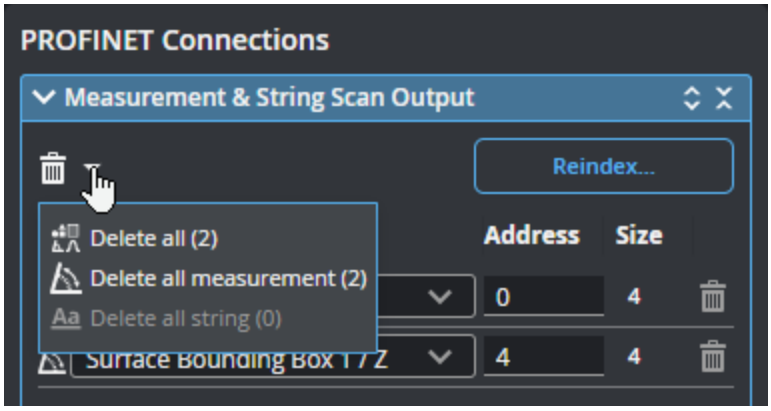
	Element	Description
		 You can also add all sources, or all sources of a given type.  Click **Add** to add the source to the PROFINET Connections table below.
4	PROFINET Connections	The connections listed here are the ones that will be sent by the protocol to a compatible PLC. You can delete a connection using the trashcan icon in its row, or change its ID. When changing an ID, make sure to use the size of the source to correctly calculate the minimum offset.

Element	Description

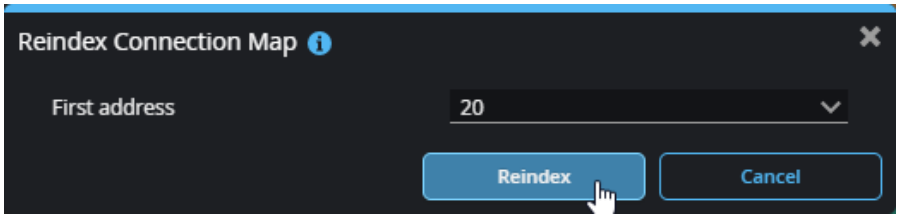


For string sources, you must specify a maximum expected length in the source's **Size** field.

You can also use shortcut commands to delete all of the connections, or specific categories of connections.



Use **Reindex...** to remove gaps of unused addresses between connections. You can also set the value of the first index.



Using PROFINET on a PC

If you are using PROFINET on a PC (using the GoPxL Manager or software developed using the GoPxL SDK), be aware of the following:

- The PC instance must either run in Administrator or be installed as a service. Otherwise, the TIA portal will not be able to discover the instance.
- The IP address of the PC instance must be a specific hardware address (such as the PC's network adapter), and not the "Any IP (0.0.0.0)" or "Localhost (127.0.0.1)".
- You must install a pcap-API compatible driver such as as Npcap or WinPcap. This lets the PC capture and send the raw network traffic necessary for the PC to interact directly with the PROFINET protocol.
- Because Windows is not a realtime OS, you must fully test a PC-based sensor system and limit the use of unnecessary software. Otherwise, you might encounter timing issues, and PLC requirements may not be met or connections may get dropped. The PC should have a CPU able of handling occasional processing spikes from either Windows or other software.

For information on running GoPxL on a PC, see *Running GoPxL on a Windows PC* on page 1188.

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.

GoPxL supports PROFINET communication to provide system state, sensor group state, stamp, and measurement output data to a PLC, as well as accept incoming commands. PROFINET defines the entire data exchange between controllers (called "IO-Controllers") and the devices (called "IO-Devices"). GoPxL operates as a PROFINET IO-Device, and a PLC that connects to it acts as the PROFINET IO-Controller. All communication is initiated by the PLC, with GoPxL responding to the PROFINET requests.

> GoPxL supports PROFINET on sensors, on GoMax ORIN / ORIN+ / NX, and in PC instances of GoPxL.

If you are using PROFINET on a PC, see *Using PROFINET on a PC* above for important information.

While the PROFINET protocol includes several types of requests, the two basic operations are reading and writing PROFINET objects. Commands are invoked by the PLC sending *set* requests. Data is requested by the PLC by sending *get* requests for PROFINET objects.

Connections Map

The PROFINET protocol supports the following objects.

Assembly Objects

A register assembly is a container that contains register blocks. This is a one-to-one mapping, which means each PROFINET module will have a corresponding control framework register assembly. These modules are added to the GoPxL irrespective of the selection of register assembly, but the data is added to this PROFINET module based on the selection of register assembly.

The "Measurement & String" module is now available in different sizes (16 to 800 bytes) to reduce PLC memory use. Select the desired module in TIA Portal.

Assembly Objects

PROFINET Module Name	Register assembly	Supported Register Block Types	Service	Module ID	Slot Number	Max size of Register Assembly
Command input module	Control Input register assembly	"ControlInput"	Write	0x30	1	67
Command output module	Control Output register assembly	"ControlOutput"	Read	0x31	2	3
System State module	System State register assembly	"System"	Read	0x32	3	76
Stamp module	Stamp Scan Output register assembly	"Stamp"	Read	0x33	4	36
Measurement & String (16)	Measurement & String Scan Output	"MeasurementValue", "String"	Read	0x34	5	16
Measurement & String (32)			Read			32
Measurement & String (64)			Read			64
Measurement & String (128)			Read			128
Measurement & String (256)			Read			256
Measurement & String (512)			Read			512
Measurement & String (800)			Read			800
Sensor Group State module	Sensor Group State register assembly	"Scanner"	Read	0x35	6	18

The PROFINET protocol supports various block types. Each block type can be either an input block or an output block. Input blocks contain write-only objects and are used by the PROFINET IO-Controller to send commands such as starting or stopping a scan. Output blocks contain read-only objects and are used to store various types information, such as control, system, and sensor group state data as well as scan output data (stamp and measurements).

Supported Block Types

The block types supported by the PROFINET protocol are listed in the table below. The size field indicates the number of registers that each block type requires.

Supported Block Types

Block Type	Read or Write only?	Size (number of PROFINET registers)	Description
"ControlInput"	WO	67	Command message set by the PROFINET IO-Controller.
"ControlOutput"	RO	3	Result data corresponding to a command that was written to the Control Input block.
"MeasurementValue"	RO	4	Measurement data of a scan output measurement data message.
"Scanner"	RO	18	Information about a specific sensor group.
"Stamp"	RO	36	Stamp message information.
"String"	RO	8 (by default)	String data of a scan output string data message. Size can be configured through the GoPxL web interface.
"System"	RO	76	Information about the LMI device.

Control Input

The input of the Control Input Block contains the fields described below:

Control Input Fields

Field	Type	Address offset	Description
Command Sequence Number	16u	0	The command sequence number is set by the external IO-Controller to uniquely identify a command request. This number is sent back in the Control Output. This is an optional field but is recommended that PROFINET IO-Controllers use this to help the IO-Controllers correlate the Command Status in the Control Output with a specific command request in Control Input.
Command Identifier	8u	2	Identifies the requested action from the external client. Supported actions: 0 = No Command 1 = Start Scanning 2 = Stop Scanning 3 = Align Sensor Group 4 = Clear Alignment 5 = Load Job File 6 = Software Trigger 9 = Set User Data Input. For use with the User Data Input tool; for more information see *User Data Input Tool* on page 1015.

Field	Type	Address offset	Description
Unused	UINT8	3	Reserved.
Command Arguments	Array[8u]	4	Optional. **Load Job File** The argument must be a null-terminated string containing the job filename. Do not include the file extension, as the control service server appends it automatically. Each register holds a single character, with a limit of 63 characters including the null terminator. **Set User Data Input** Use up to the full 63 registers for data input.

Control Output

The output of the Control Output Block contains the fields described below:

Control Output Fields

Field	Type	Address offset	Description
Command Sequence Number	16u	0	The command sequence number received from the IO-Controller in the Control Input.
Command Status	8u	2	Status of the command, referenced by the Command Sequence Number. 0 = No Command Received 1 = In Progress 2 = Success 3 = Fail

System State

The output of the System State Block contains the fields described below:

System State Fields

Field	Type	Address offset	Description
System Uptime	64u	0	Amount of time the application has been running, in seconds.
System State	8u	8	Whether system is running or not. 0 = System is stopped 1 = System is running
Current Job File Name Length	8u	9	Number of characters in the current job file name.
Current Job File Name	Array[8u]	10	Name of the job file currently loaded into the system,

Field	Type	Address offset	Description
			without the extension. Maximum length of job file name is 64 characters. Each register contains a single character. Job file name is NOT null terminated.
Buffer Count	8u	74	The current amount of buffered scan output data when buffering is enabled.
Buffer Overflow	8u	75	A value of 0 indicates that no overflow has occurred. A value of 1 indicates that overflow occurred and data is being lost.

Sensor Group State

The output of the Sensor Group Block contains the fields described below:

Sensor Group State Fields

Field	Type	Address offset	Description
Current Encoder Position	64s	0	The current encoder position.
Current Sync Time	64u	8	Current time of the clock used to stamp all scan output messages, in microseconds.
Alignment State	8u	16	Indicates whether all devices under the sensor group are aligned (that is, all devices in the sensor group have a non-identity transform matrix). 0 = Not Aligned 1 = Aligned 2 = Aligning
Laser Enabled	8u	17	Whether one or more devices under a sensor group has its laser enabled to be able to turn on. 0 = All devices' lasers are disabled and cannot turn on 1 = One or more devices' lasers can be turned on

Stamp

The output of the Stamp Block contains the fields described below:

Stamp Fields

Field	Type	Address offset	Description
Z Position	64s	0	Encoder position at time of last index pulse of last frame.
Encoder Position	64s	8	Encoder position of last frame when the image data was scanned/taken.

Field	Type	Address offset	Description
Time	64u	16	The timetick of the last frame. Timetick is a 64-bit counter that increases with time, so each scan from the same device has a unique timetick value. On Gocator 3D sensors, to convert this value to microseconds, use the following: `time (µs) = timetick / 1.024`.
Frame Index	64u	24	The frame number of the last frame.
Digital Input States	8u	32	Digital input states of last frame. Bit 0: State of digital input 1 on master or sensor Bit 1: State of digital input 2 on master or sensor
Buffer	8u	33	Unused
Buffer	8u	34	Unused
Buffer	8u	35	Unused

Measurement

The output of the Measurement Block contains the fields described below:

Measurement Fields

Field	Type	Address offset	Description
Value	32s	0	Measurement value. This value is the actual measurement value multiplied by 1000. If the measurement value is invalid, the 32-bit value is set to 0x80000000.

String

The output of the String Block contains the fields described below:

Measurement Fields

Field	Type	Address offset	Description
String	string	0	String value. Unused space will be set to null (0x0). The string is otherwise not null terminated. If invalid, the block's buffer is entirely set to null, equivalent to an empty string. The string will be truncated if the block's size is smaller than the length of the string.

Siemens PLC Instructions

This section explains how to set up and verify PROFINET communication between a Siemens S7-1200 PLC and GoPxL using Siemens TIA Portal. It walks through the key steps required to bring the system

online, including project creation, PLC setup, GSD installation, sensor configuration, network setup, and basic functional checks. The goal is to establish a stable and predictable communication link that supports reliable operation and repeatable system commissioning.

The guide also includes practical configuration recommendations and troubleshooting steps based on real integration and testing experience. These notes help avoid common issues such as device conflicts on the network, incorrect IP configuration, and project corruption. Following these steps helps ensure consistent PLC-to-sensor communication, simplifies commissioning and debugging, and reduces integration risk in both development and production environments.

Creating and Configuring New Projects for Siemens PLCs

Before continuing, make sure of the following:

- A Siemens S7-1200 PLC is powered on and reachable on the network.
- TIA Portal is installed and the PLC firmware is supported by the selected TIA version.
- The host PC and PLC connect to the same Ethernet subnet.
- GoPxL is installed and the sensor is reachable via the Connect page.
- PROFINET is enabled on the sensor and the sensor has been restarted after enabling it.

CREATING A NEW PROJECT

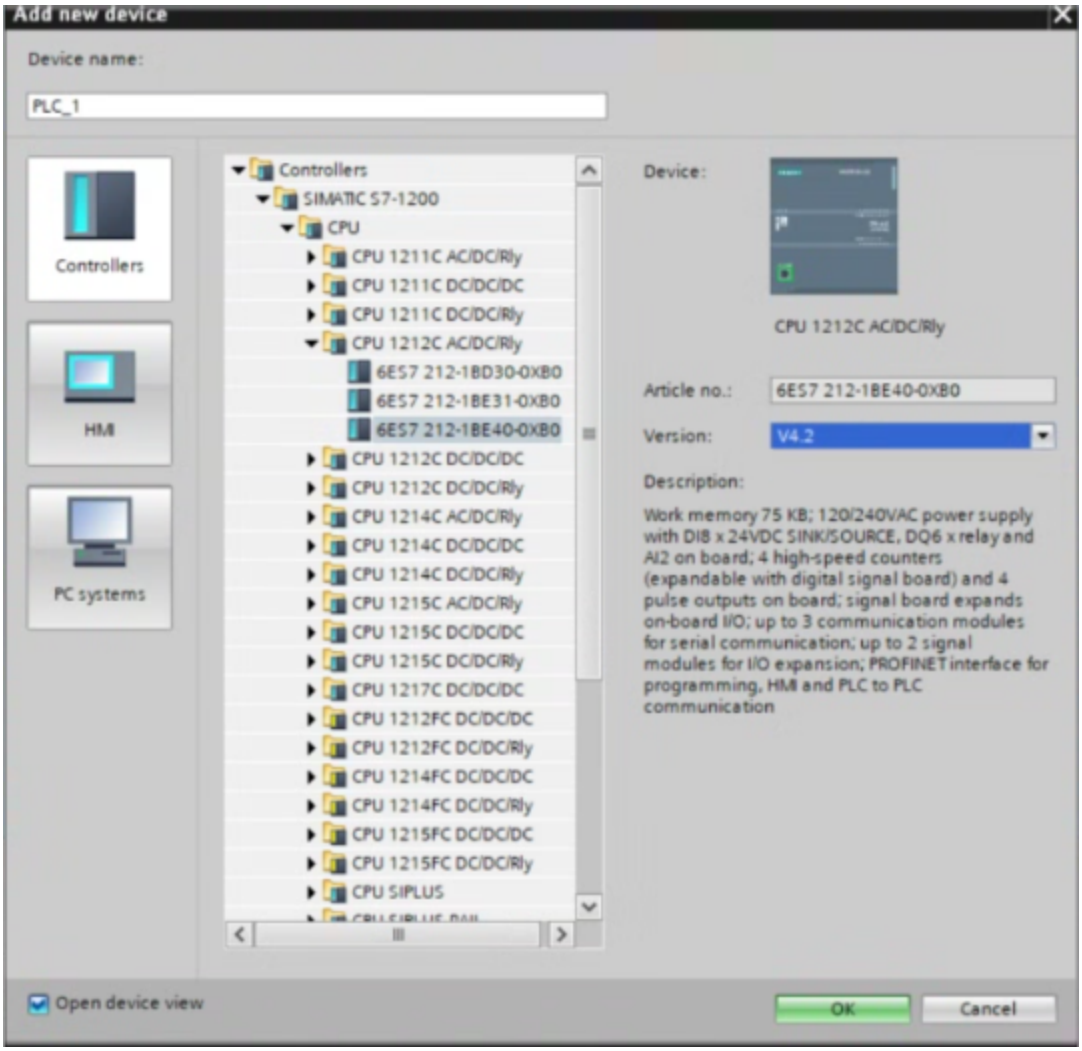
1. Launch TIA Portal.
2. Click **Create new project**.
3. Enter the required project details.
4. Click **Create**.
5. Click **Project view**.

ADDING AND CONFIGURING THE SIEMENS S7-1200 PLC

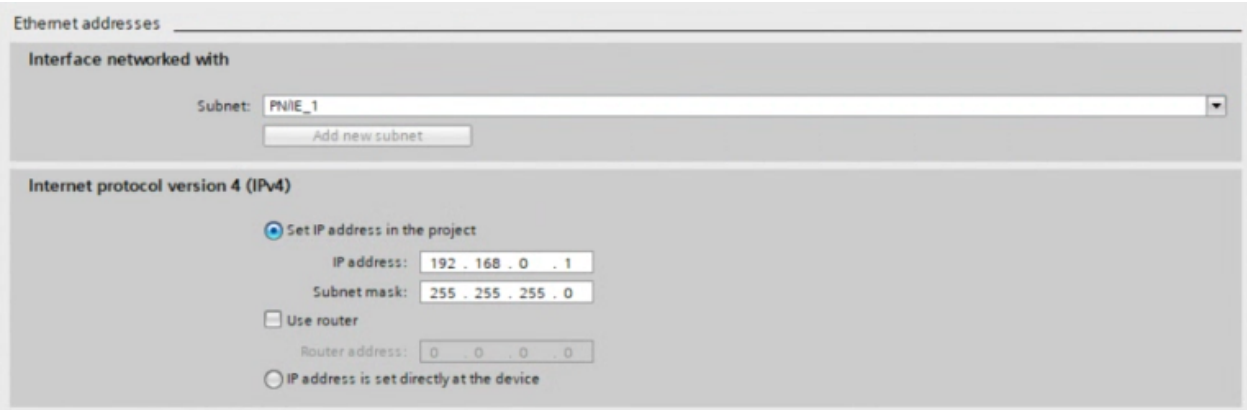
1. In the Project tree, open **Devices**.
2. Double-click **Add new device**.
3. In the **Add new device** dialog, do the following:
   a. Click **Controllers**.
   b. Find the correct PLC model in list of models, for example:

      `SIMATIC S7-1200 → CPU → CPU 1212C AC/DC/Rly → 6ES7-212-1BE40-0XB0 → Firmware V4.x`

      Make sure the selected CPU order number and firmware version match the physical PLC. The correct model and firmware version appear on the PLC device label and in the TIA Portal online diagnostics view.
4. Click OK.



CONFIGURING PLC NETWORK SETTINGS

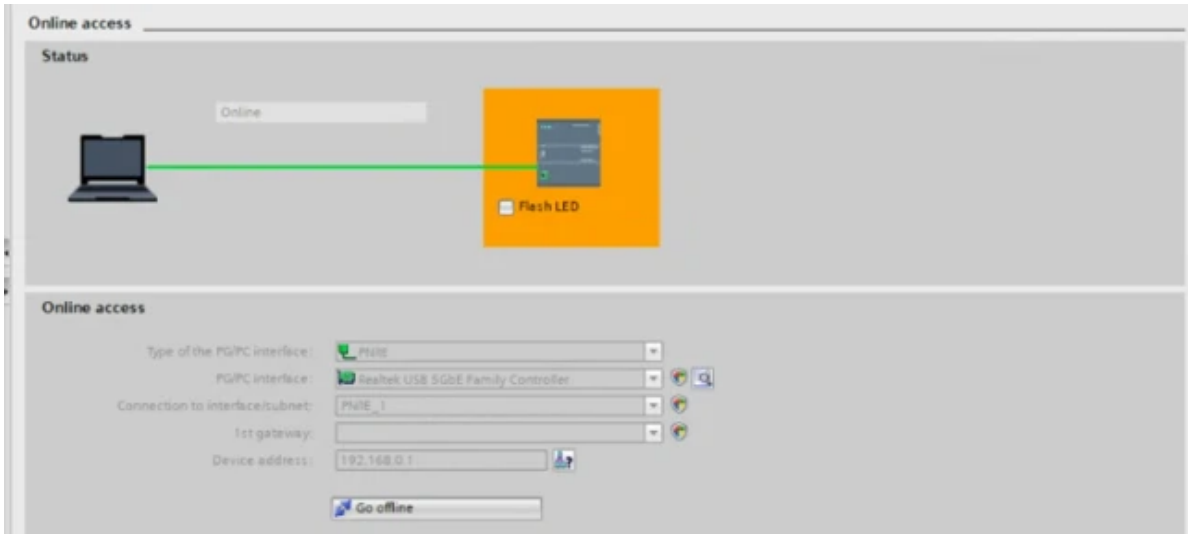


1. In the **Devices** panel, right-click the PLC and choose **Properties**.
2. Go to **PROFINET interface [X1] → Ethernet addresses** and do the following:
3. Click **Add new subnet**.

4. Under **Internet protocol version 4 (IPv4)**, select "Set IP address in the project".
5. Enter the PLC's IP address and subnet mask.
6. Click **OK** and then **Save project**.

VERIFYING PLC CONNECTIVITY

1. In the **Devices** panel, expand the PLC node.
2. Double-click Online & diagnostics.
3. Open Online access and set the following:
   - **Type of PG/PC interface**: PN/IE
   - **PG/PC interface**: The Ethernet adapter
   - **Connection to subnet**: PN/IE_1
4. Click **Go online**.
5. If connected:
   a. Select **Flash LED** to confirm communication.
   b. Verify the LEDs are flashing; if so, disable flashing.
6. From the CPU pane:
   a. Switch to RUN and confirm green LED.
   b. Switch to STOP and confirm amber LED.
7. Click **Go offline**.



Compilation is not required for simple online monitoring if the PLC already runs a valid program. Also note the following:

- Compile validates hardware configuration and PLC program consistency.
- Download transfers the compiled configuration to the PLC.

- Compilation is required before download when:
  - Hardware configuration changes
  - GSD modules are added or changed
  - I/O layouts change

Configuring PROFINET Connection between a Siemens S7-1200 PLC and GoPxL

Before continuing, make sure of the following:

- You have configured the S7-1200 PLC; for more information, see *Creating and Configuring New Projects for Siemens PLCs* on page 1136.
- A GoPxL-enabled sensor in GoPxL on PC Instance with PROFINET support is available, powered on, and connected to the same Ethernet network.
- A single PLC project may include multiple PROFINET-enabled sensors, provided each sensor has PROFINET enabled and a unique device name and IP address.
- Only one active GoPxL instance per PC is supported. Multiple GoPxL instances on the same host system are not supported.

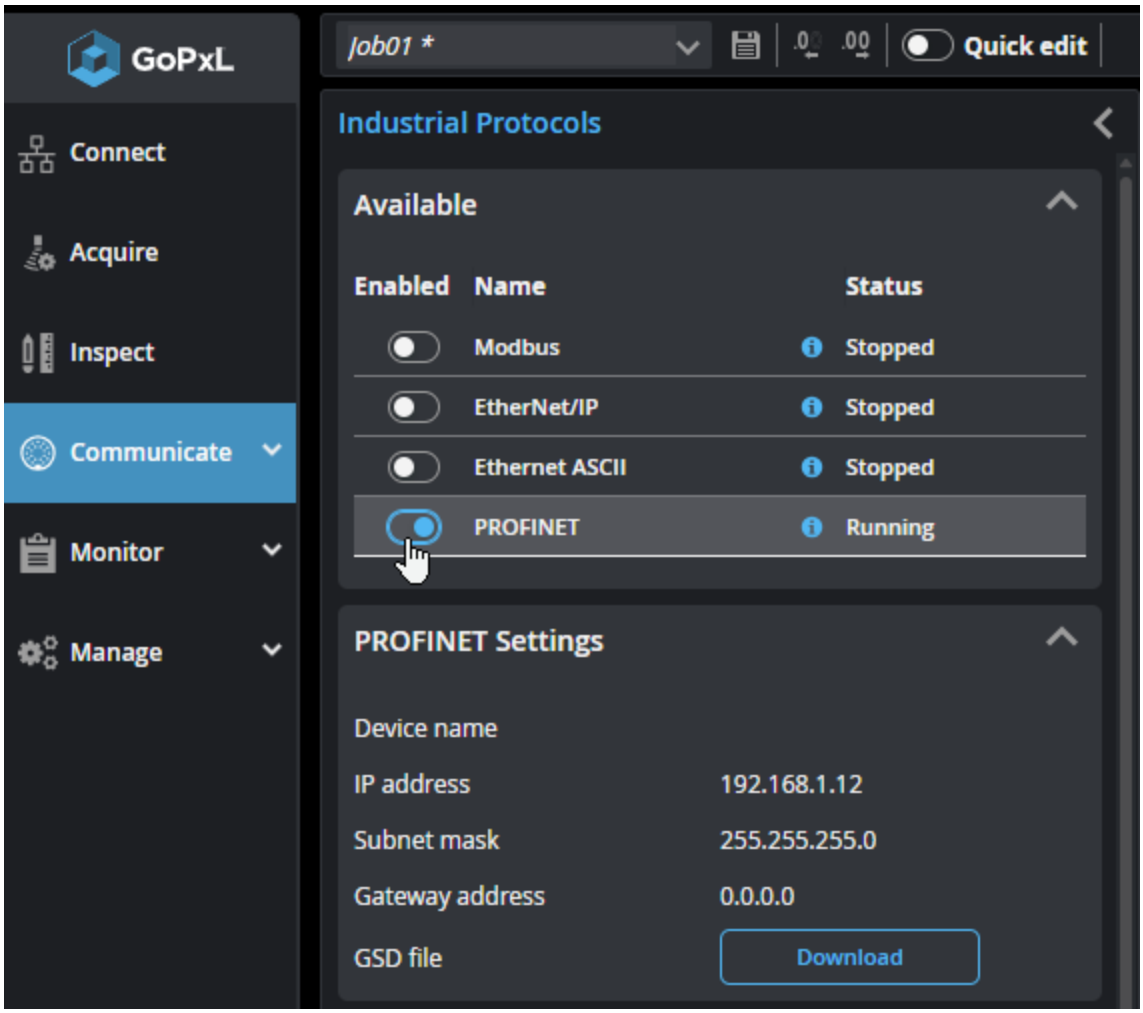
LMI recommends the following network constraints:

- Avoid multiple PROFINET devices on the same subnet.
- Maintain one PLC project per sensor to avoid device conflicts.

INSTALLING GSD FILES

To use a GoPxL device (a sensor, or a GoMax or PC instance of GoPxL accelerating a sensor) with a PROFINET PLC, you must download a GSD file from GoPxL and install it in TIA Portal. The GSD file describes the capabilities of the GoPxL device, such as the data stream, I/O modules, and connection information.

1. In the GoPxL web interface, go to **Communicate** > **Industrial Protocols**, and enable PROFINET.



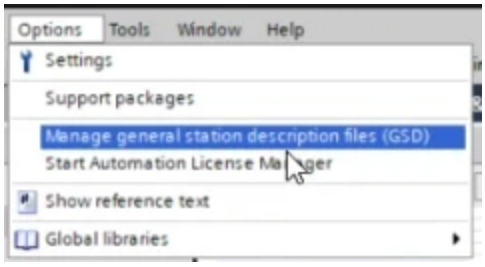
2. In **PROFINET Settings**, click **Download**.

   Make note of where the file is saved.

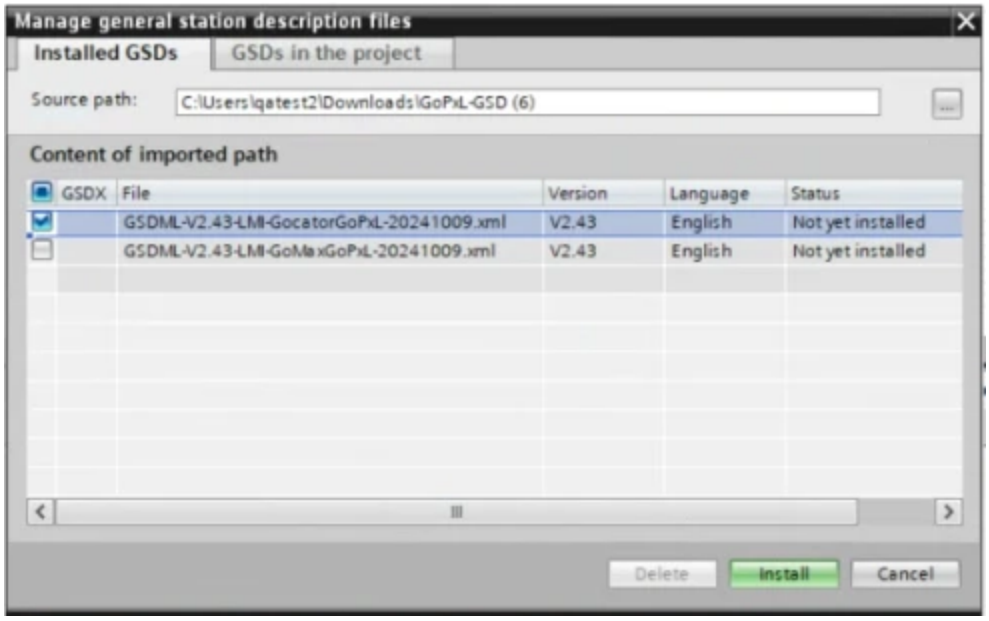
   You may need to tell your browser to accept the download.

3. In TIA Portal, load the PLC project in Project view.

4. From the main menu, choose **Options > Manage general station description files (GSD)**.



5. Browse to the folder containing the GSD file in Source path and select the file.



6. Click **Install**.

   Verify that the GSD file is correctly installed.

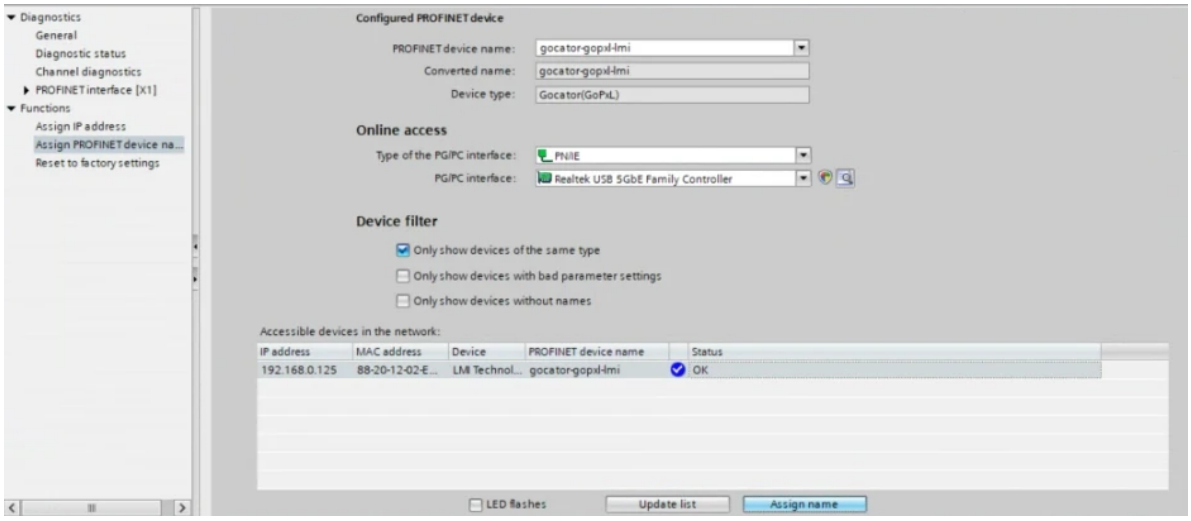
ADDING AND CONFIGURING GOPXL SENSOR

1. From the left panel, open **Devices & networks**.
2. In **Catalog** panel, go to **Other field devices > PROFINET IO > Sensors > LMI Technologies > LMI Sensors > Gocator (GoPxL)**, or **GoMax (GoPxL)** if you are configuring this for a GoMax.
3. Drag the device icon into the **Devices & networks** workspace.
4. Assign the PROFINET interface by doing the following:
   a. Select the GoPxL device.
   b. Click the **Not assigned** link.
   c. Select the appropriate PLC interface.
5. Confirm the device is successfully assigned.
6. Right-click the GoPxL device icon and select **Properties**.
7. Go to **Ethernet address** and assign the IP address and subnet mask of the sensor.
8. Go to **Real-time settings** and configure the parameters as required.
9. Click **Save project**.

Verifying Configuration and Connection

1. From the **Devices** panel, go to **Ungrouped devices > gocator-gopxl-lmi**.
2. Double-click **Online & diagnostics**.
3. Go to **Functions > Assign PROFINET device name**.
4. Confirm the correct interface is selected.

5. Click **Update list**.
6. Verify that the sensor is correctly detected.



If the sensor does not appear, do the following:

- Confirm the sensor is powered up.
- Confirm PROFINET is enabled in GoPxL.
- Confirm IP and subnet configuration.

Note the following:

- Each PROFINET device on the network must have a unique device name.
- Device name assignment must complete before cyclic data exchange can begin.

To rename a device name, do the following:

1. Open **Devices & networks**.
2. Right-click the GoPxL device
3. Select **Properties**.
4. Go to Ethernet address > PROFINET device name, and enter the new name, and save the project.
   After renaming, follow the steps above to apply the new name to the physical sensor.

COMPILING AND DOWNLOADING CONFIGURATION TO PLC

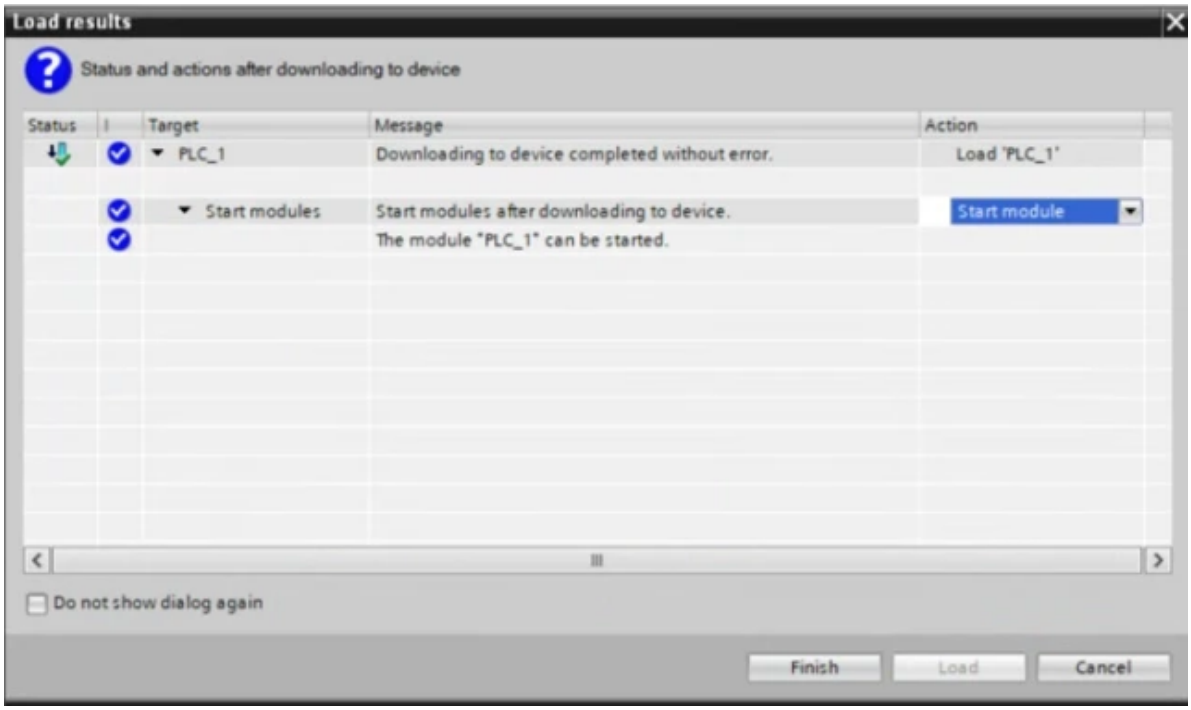
You must compile and download the configuring to the PLC whenever you do any of the following:

- Add or remove devices
- Modify the PROFINET configuration
- Install or update GSD files

- Change the module layout or I/O size
- Update device names or network parameters

To do compile and download the configuration

1. From the main toolbar, select **Edit > Compile**.
2. In the **Compile** dialog, select **Hardware and software** (or **Hardware only** if you only changed device/network settings).
3. Confirm the configuration compiles with no errors.
4. From the main toolbar, select **Online > Download to device**.
5. Select the correct PG/PC interface and Ethernet adapter, and click **Load**.
6. Review the Load preview and confirm the target device matches the intended PLC.
7. In the **Load results** dialog, confirm successful completion with no errors.
8. Under Action, select **Start module** and click **Finish**.



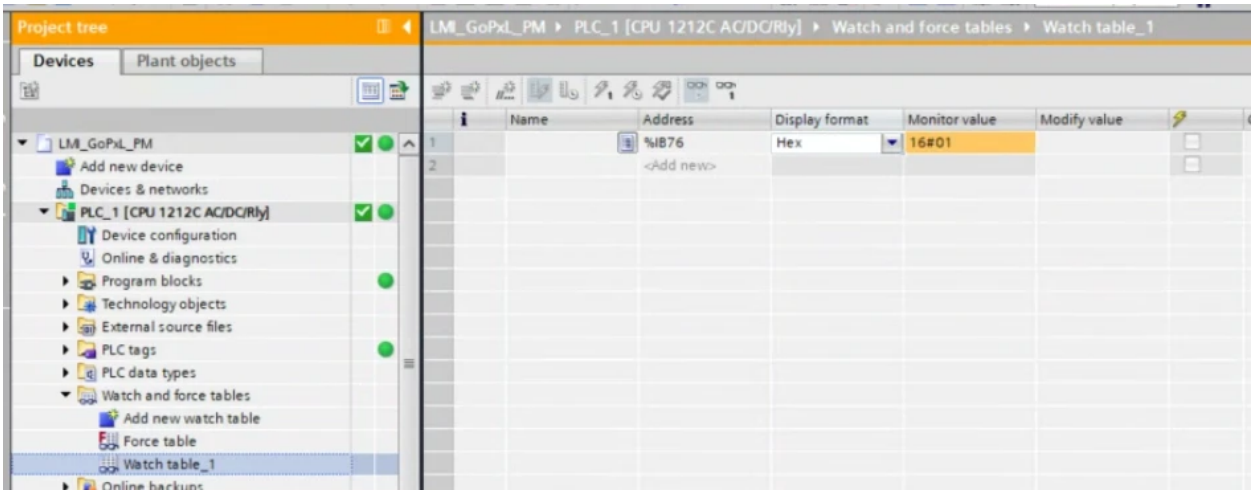
9. Verify the CPU enters RUN state (green status LED).

MONITORING SENSOR STATE

To monitor the sensor state:

1. Select **Go online** from the main toolbar.
2. From **Devices > PLC_1 > Watch and force tables**, select **Add new watch table**.
3. In the **Address** column, enter `%IB76` (`16#01` hex - this value references Input Byte 76 which maps to the sensor status).

4. Right-click **Monitor value** and select **Monitor all**.
5. In the GoPxL web interface, start the sensor and do the following:
    a. Confirm value = 1 (`16#01` hex).
    b. Select **Stop Sensor**.
    c. Confirm value = 0 (`16#00` hex).
6. Select **Go offline** to disconnect.



For information on the PROFINET format, see *Connections Map* on page 1130.

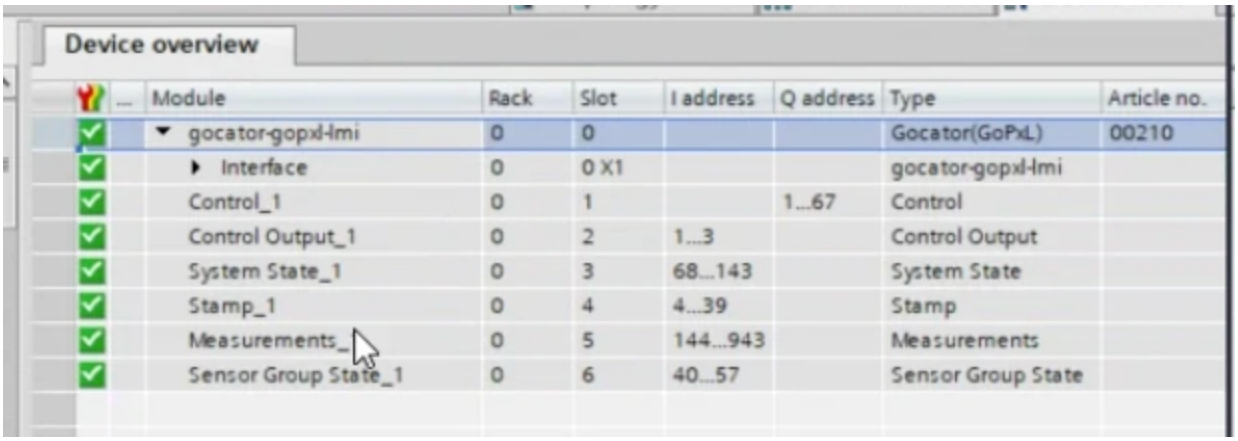
DEVICE OVERVIEW: MODULE MAPPING AND USAGE

GoPxL exposes its control and status data to PLCs through PROFINET I/O modules, which are defined in the GSD file and instantiated inside the PLC project. Each PROFINET module maps directly to a corresponding internal GoPxL register assembly, which defines the structure, size, and meaning of the exchanged data. This establishes a one-to-one mapping between:

- PROFINET modules (visible and selectable in TIA Portal);
- and GoPxL register assemblies (internal data blocks used by the control framework).

When a module is enabled in the PLC project, GoPxL automatically detects the selected module set during connection and activates the corresponding register assemblies internally. The cyclic I/O data exchanged with the PLC reflects only the enabled modules.

The **Device overview** tab in TIA Portal shows the PROFINET device structure and cyclic I/O layout for the device. Each row represents one PROFINET module that GoPxL exposes to the PLC. Together, these modules define how control commands, system status, measurement data, and diagnostic information are exchanged between the PLC and GoPxL.



Interface: Defines the PROFINET network interface and real-time communication parameters. This module enables cyclic communication between the PLC and GoPxL.

Control_1 (Output): Carries command data from the PLC to GoPxL. Typical usage includes start, stop, trigger, alignment, and job-load commands.

Control Output_1 (Input): Returns command status and acknowledgments from GoPxL back to the PLC. This module supports handshake logic and command validation.

System State_1 (Input): Provides system-level status bits, such as sensor readiness, fault state, and operational health.

Stamp_1 (Input): Delivers timing and frame metadata, including timestamp, encoder value, and frame index. This data supports latency measurement, synchronization, and traceability.

Measurements_1 (Input): Contains cyclic measurement data. This module delivers GoPxL tool measurement values based on the active job configuration.

Sensor Group State_1 (Input): Reports the operational state and diagnostic information of sensor groups.

Note the following:

- The I/O address ranges for GoPxL PROFINET modules may appear differently across projects and PLCs, even when the same GSD file and module set is in use. This behavior is expected and reflects how Siemens TIA Portal allocates memory based on project context, hardware configuration, and address optimization rules. Although the absolute I/O addresses differ, the internal structure, byte layout, and semantic meaning of each module remain follow the GoPxL PROFINET Byte Map definition; for more information, see the table in *Assembly Objects* on page 1130.
- Each PROFINET module corresponds to exactly one internal register assembly.
- Module slot numbers and sizes are fixed by the GSD definition.
- GoPxL detects the configured module set at runtime and transmits only the enabled data blocks.

- PLC memory addresses (I/O offsets) are assigned dynamically by TIA Portal and therefore vary by project, but the internal byte layout of each module remains constant.
- The following is a summary of the data flow:

  PLC → GoPxL:

  Control commands transmit through the Command Input module.

  GoPxL → PLC:

  Status, results, timing, and state data transmit through the remaining read-only modules.

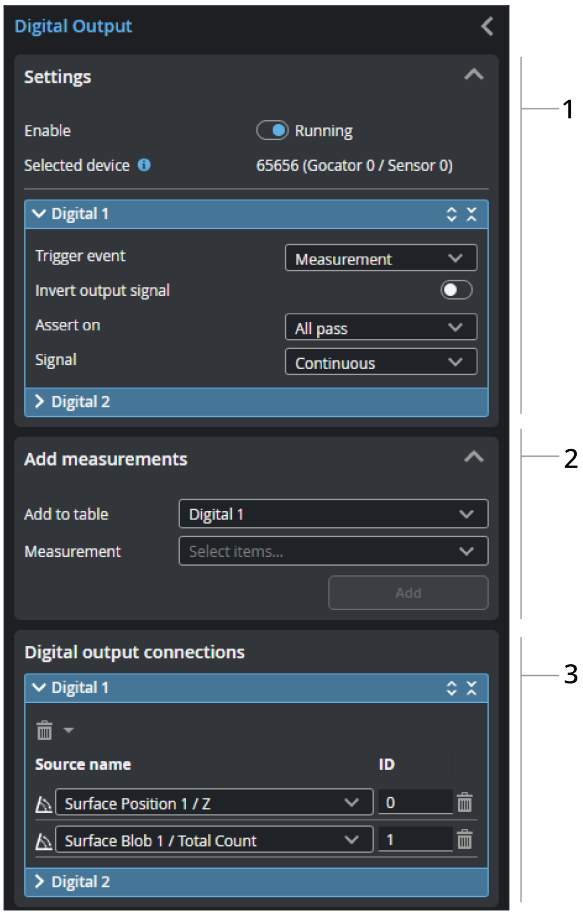
Digital Output

GoPxL can send digital signals over a sensor's digital outputs based on measurement decisions, software commands, and other events. These signals can then be used to control a PLC or other external device, such as indicator lights or air ejectors. The following trigger events are available:

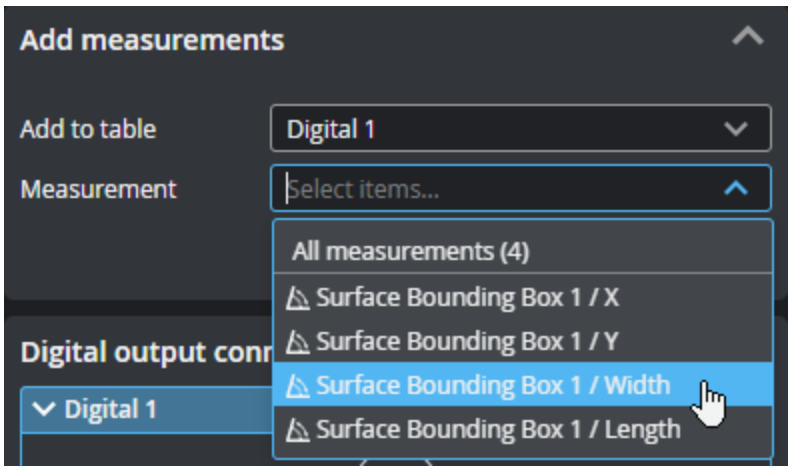
- Measurements
- Software (that is, a command sent from GoPxL SDK applications)
- Alignment state
- Exposure begin or end
- System state (that is, when the system is ready to accept digital input triggers)

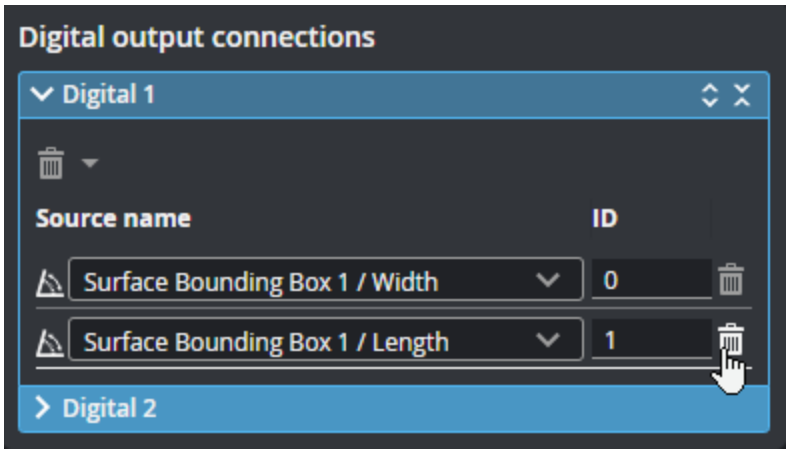
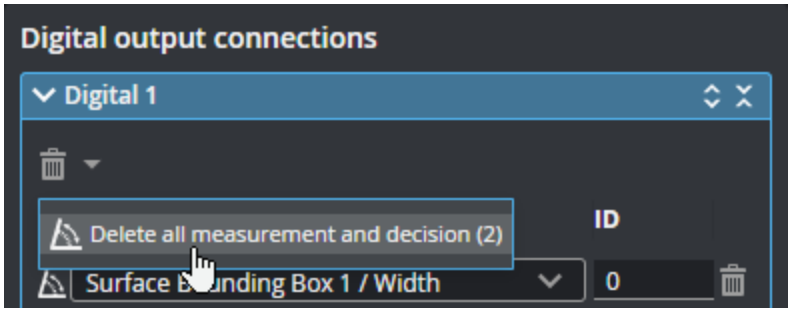
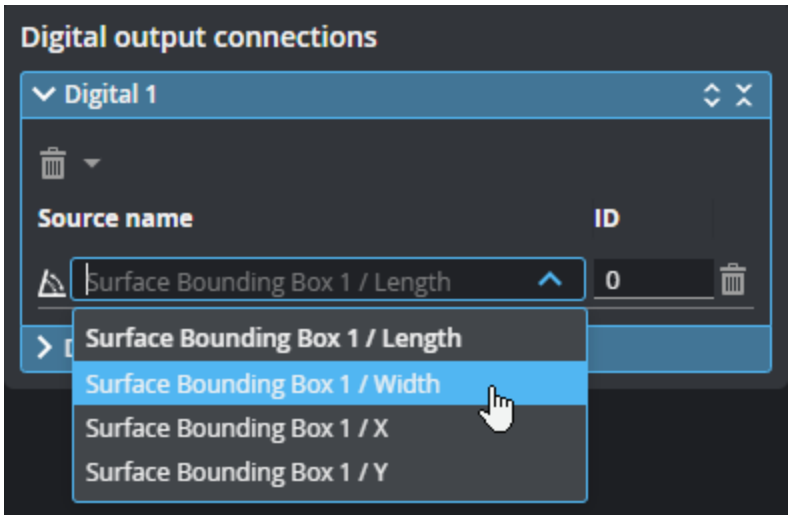
GoPxL provides two digital output channels. For information on wiring digital outputs to external devices, see the following: *Digital Outputs* on page 1349.

Counts of low and high digital signals, as well as digital output drops, are available on the **Monitor** > **Health** page, in the expandable **Devices** section. For more information, see *Health* on page 1168.



	Element	Description
1	Settings	**Enable** lets you enable digital output.

	Element	Description
		Active Device shows which sensor's digital output port GoPxL will send digital signals. Each expandable "Digital {n}" section lets you choose the type of trigger event and configure settings related to the signal. When you set **Trigger event** to Measurements, you use the **Add Measurements** and **Digital Output Connections** sections to add and manage the measurements that GoPxL sends as digital outputs. Otherwise, a digital signal is sent when the selected trigger event occurs. For information on the trigger settings, see *Digital Output Settings* on the next page.
2	Add measurements	Lets you choose a measurement and which digital output the signal is sent over. This section is only displayed if at least one digital output's trigger is set to Measurements. To set a measurement for digital output, choose the output in **Add to table**, choose the measurement in **Measurement**, and click **Add**. You can also add all measurements at once by selecting "All Measurements".  After adding a measurement, it appears in the **Digital output connections** section of the digital output you chose.
3	Digital output connections	Lets you view, delete, and change the measurements that trigger a digital output signal. This section is only displayed if at least one digital output's trigger is set to Measurements. To delete a measurement, click the trashcan icon next to it.

Element	Description
	 You can also use shortcut commands to delete all of the connections.  To change a measurement, click an output's drop-down and choose a different measurement. 

Digital Output Settings

Parameter	Description
Trigger event	GoPxL can send a digital output signal when one of several trigger events occurs.

Parameter	Description
	Depending on the event you choose, other parameters are displayed that you may need to configure. For more information, see the other parameter descriptions below. **Measurement** GoPxL sends a digital signal based on the decisions, for each frame, returned by the measurements added to the digital output. Set **Signal** to choose between a pulsed or continuous signal. Optionally, set a pulsed signal to be scheduled with **Scheduled**. You must set **Assert on** to configure what types of measurement decisions result in a digital signal. **Software** When GoPxL receives a software command from a GoPxL SDK application, it sends a digital signal over the configured digital output. Use **Signal** to choose between a pulsed or continuous signal. Optionally, set a pulsed signal to be scheduled with **Scheduled**. For pulsed or scheduled pulse, the sensor must be scanning before sending the command. For continuous signal, the sensor does not have to be scanning **Alignment** When a sensor is in an aligned state, GoPxL sends a continuous digital signal. Only alignment performed using the **Alignment** routine on the **Acquire** page results in a trigger event (for more information, see *Aligning Sensors with up to 5 Degrees of Freedom* on page 123): alignment performed using the Surface Align Wide or Surface Align Ring tools does not. Whether the sensor is running does not affect the output. **Exposure begin** GoPxL sends a pulsed digital signal with the provided pulse width when exposure begins. **Exposure end** GoPxL sends a pulsed digital signal with the provided pulse width when exposure ends. **System state** The System state output signal goes high after the sensor has started and is ready to receive digital input to start triggering for scanning, that is, "External Input" triggering is enabled in **Source** in the **Trigger** panel. For more information on triggers, see *Triggers* on page 141.

Parameter	Description
	External input GoPxL sends a continuous digital output signal when it receives a digital input trigger. You can this option to troubleshoot external input signals. The sensor must be running for an external input to trigger a digital output.
Invert output signal	Inverts the output signal. Normally, GoPxL will send a high digital signal when an event occurs. Toggling this setting causes GoPxL to send a low signal when the event occurs.
Assert on	When **Trigger event** is set to Measurement, use this setting to determine which type of decision in the measurements added to the output in **Digital output connections** cause GoPxL to send a digital signal. GoPxL checks decisions for each frame. One of the following: **All pass**: If all added measurements return a pass decision, GoPxL sends a digital signal. **Any fail or invalid**: If any added measurement returns a fail decision or Invalid, GoPxL sends a digital signal. **Always**: GoPxL sends a digital signal, for each frame, after the tools producing the selected decisions have finished processing, regardless of the result (pass, fail, or invalid). The output activates only once for each frame even if multiple decision sources are selected. LMI recommends using pulsed digital signals with this mode, as otherwise the signal will remain high indefinitely.
Signal	Measurement and Software trigger events let you choose the signal type (pulsed or continuous). **Pulsed** GoPxL sends a pulsed signal over the digital output, using the value you set in **Pulse width**. **Continuous** GoPxL sends a continuous signal over the digital output. When the trigger event is Measurement and **Assert on** is set to always, LMI recommends using pulsed digital signals instead of continuous.
Pulse width	The pulse width used when the signal type is pulsed. Note that Measurement and Software trigger events let you choose between pulsed and continuous, whereas the exposure types are always pulsed.
Scheduled	When this setting is enabled, you can set a delay using **Delay** and **Delay Domain**. Only Measure and Software trigger events let you schedule output.

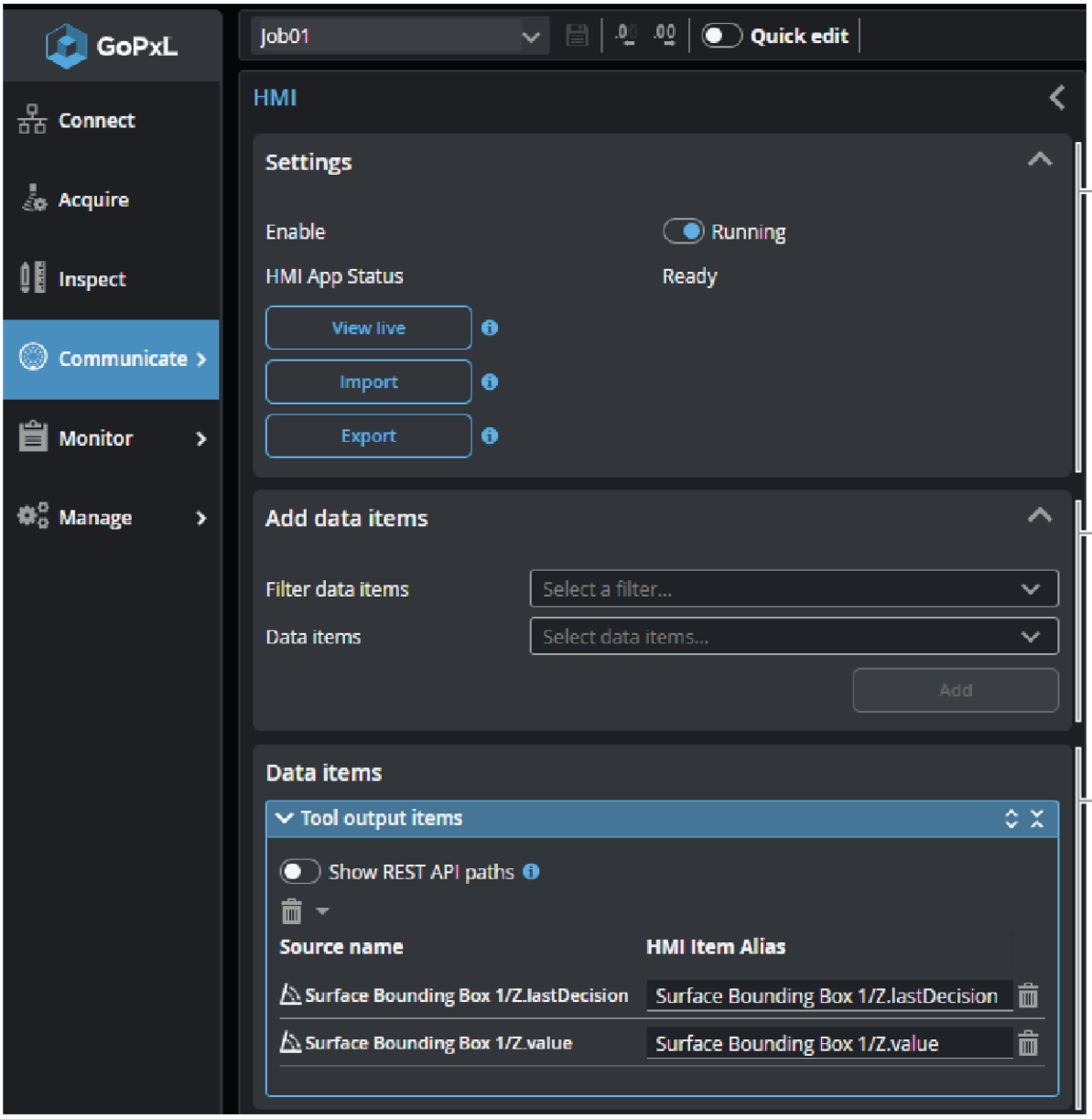
Parameter	Description
Delay Domain	The delay and domain GoPxL uses when **Scheduled** is enabled. The delay domain is either time-based or based on encoder distance.
Delay	The unit depends on whether the domain is time (microseconds) or encoder (millimeters).

HMI

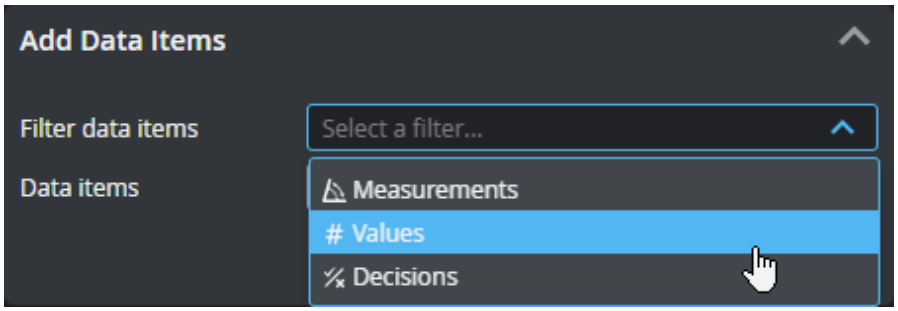
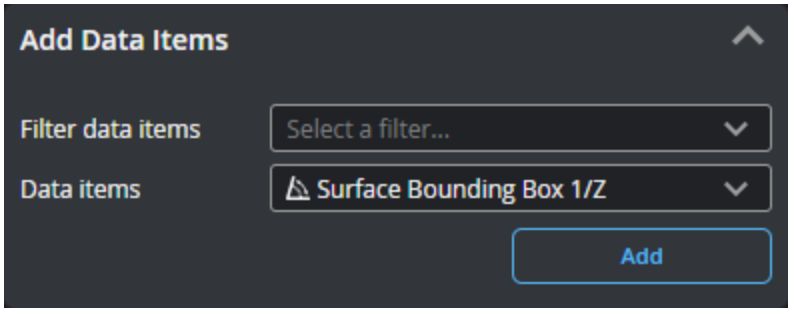
The HMI panel lets you enable GoHMI, choose which outputs you want to send to an HMI application, and launch with the HMI. You can also import and export an HMI.

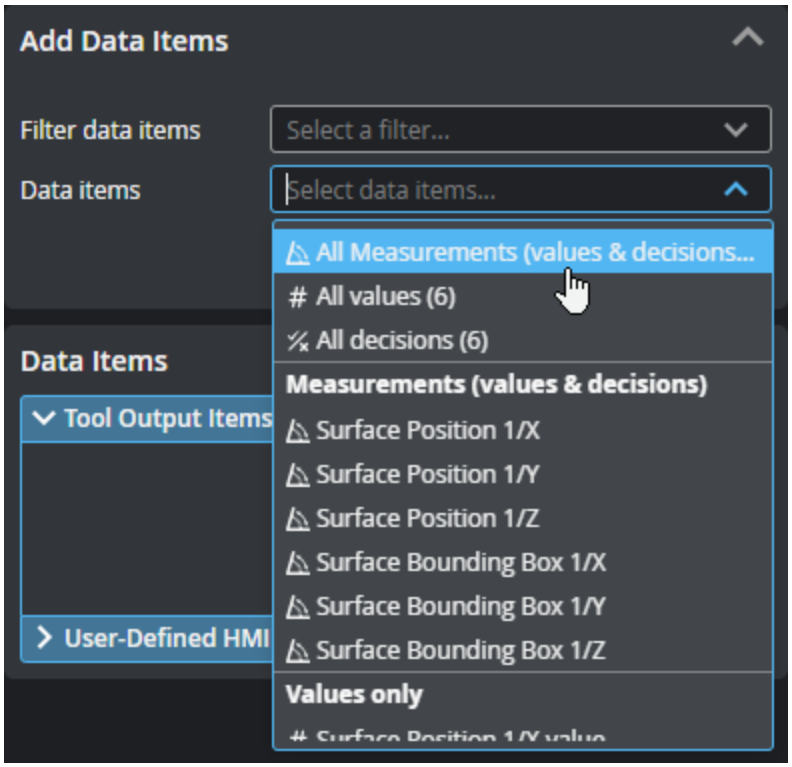
You can send measurement data items (which contain both values and decisions), or values or decisions individually.

You configure the HMI on the **Communicate** > **HMI** page. For information on how to design an HMI app, see *GoHMI and GoHMI Designer* on page 1210.

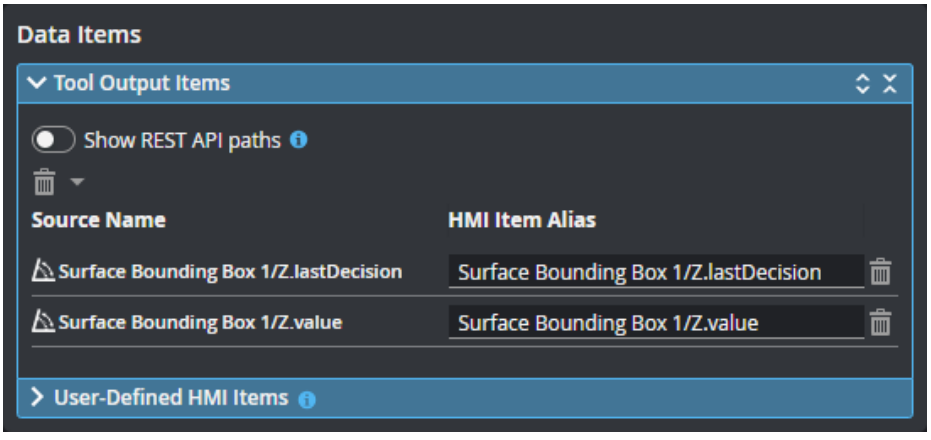


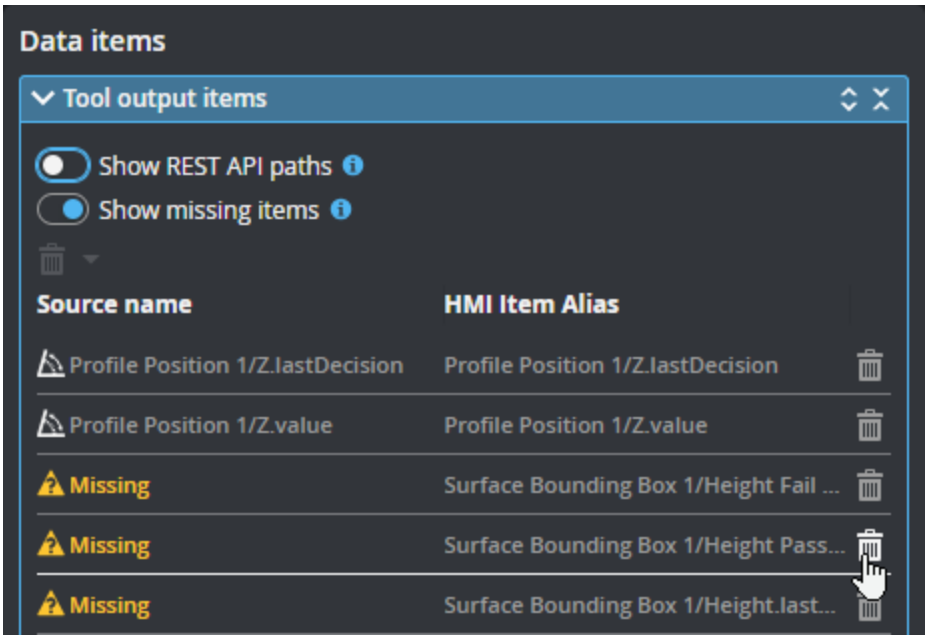
When you start HMI on a PC instance of GoPxL, a helper application (WebIQ Server) launches.

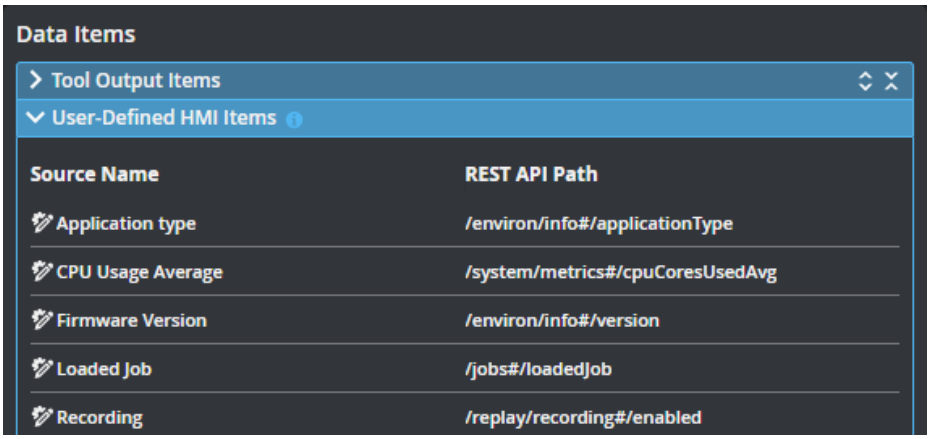
	Element	Description
1	Settings	After you have enabled the service, the following functions are available: **View live** Opens the HMI in a browser tab. **Import / Export** Lets you import and export HMIs so that you can transfer them to a different platform, for example, going from a PC to a GoMax device, or back up an HMI. **HMI App Status** Shows the status of the app.
2	Add Data Items	Lets you add sources (such as stamps or measurements) that GoPxL will send to the running HMI. Added connections appear in the **Data Items** section at the bottom of the panel. If you have many measurements, you can filter them by type to make it easier to find the item you want in the **Data Item** drop-down. You can enable more than one filter.  Choose an item in **Data items**. You can also  You can also add all items, or all items of a given type.

Element		Description
		
3	Data Items	The data listed here will be sent to the running HMI app. For more information, see *Data Items* below.

Data Items

Type	Description
Tool Output Items	This section contains the measurement decisions and values you added, which will be sent to the HMI.  You can delete an item using the trashcan icon in its row. You can also use shortcut

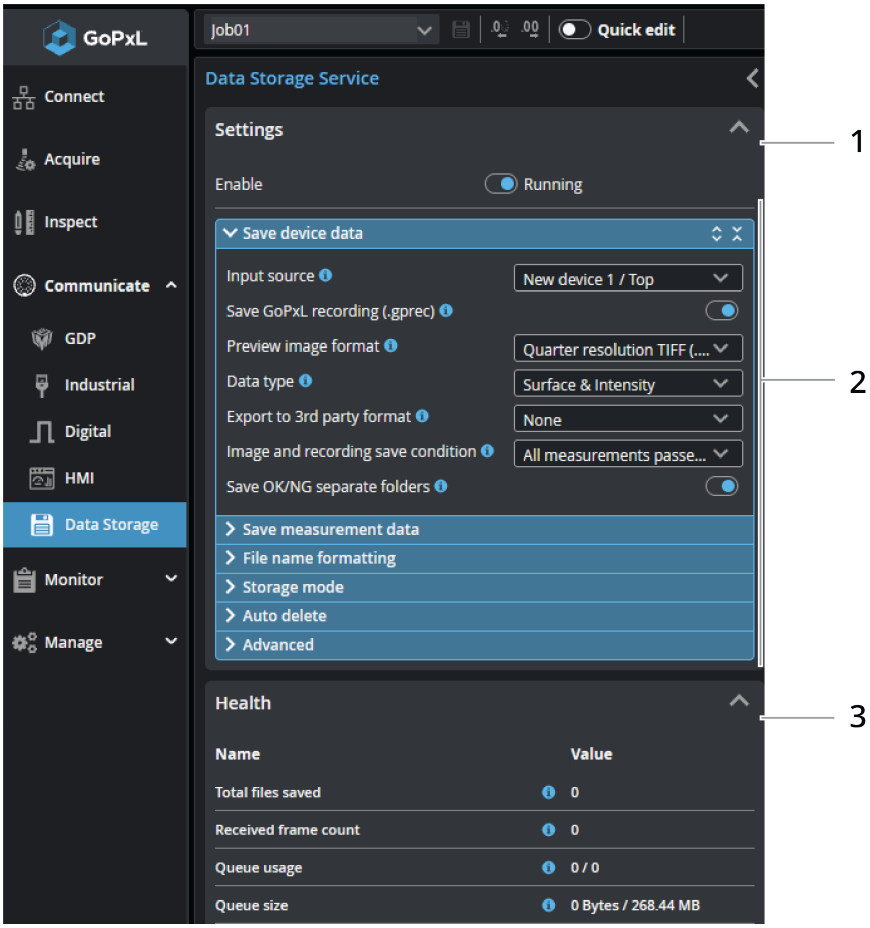
Type	Description
	commands to delete all of the items, or specific categories of items. **Show REST API Paths** Toggles between showing the full internal path to an item, and an alias, which is a simplified representation of the item. In GoHMI Designer, you will see the REST API path for the items you add. Enabling this setting in GoPxL can help make it easier to find which item is which when developing an HMI. **Show missing items** If the source for an item is missing (that is, the tool providing the source has been removed in the job, or an output has been disabled), this option is displayed. Enabling it lists connections whose source can't be found. You can delete the missing connection, enable the source measurement in the tool if it has just been disabled. If the tool itself is missing and you add it again, you may still need to delete missing items on the HMI panel, and add them back again, to make sure you add the correct ones. . 
User-Defined HMI Items	This section contains GoPxL REST API resources that are not tool outputs. They are defined and managed from within GoHMI Designer.

Type	Description
	

Data Storage Service

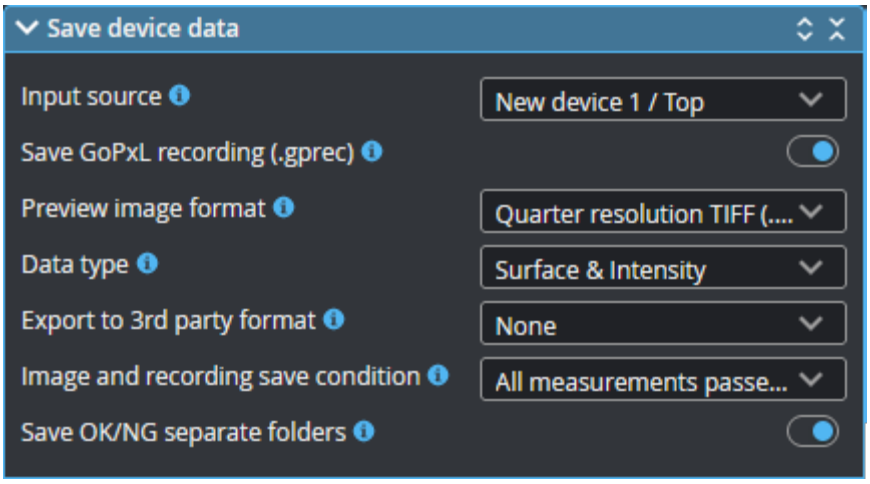
The data storage service uploads data acquired on sensors and smart cameras to a remote FTP server. On GoMax Smart Vision Accelerators, smart cameras, and GoPxL for PC, data can be stored to a local folder. Auxiliary data, such as preview images and logs, can also be stored.

> On GoMax devices and 2D smart cameras, when the service is configured to save files on the device's internal FTP server (**Storage mode** > **Save location** is set to Local), the optional timestamp used in file names is based on the internal time of the device, and the time is reset when the device power cycles.



	Element	Description
1	Enable/Disable	Enables or disables the data storage service and displays its status.
2	Data Storage settings	Lets you configure the data storage service. For more information, see *Settings* on the next page.
3	Health pane	Provides information related to the data storage service. For more information, see *Health* on page 1166.

You configure the source, save condition, and types of data the service saves to the FTP server in the expandable **Save device data** section.



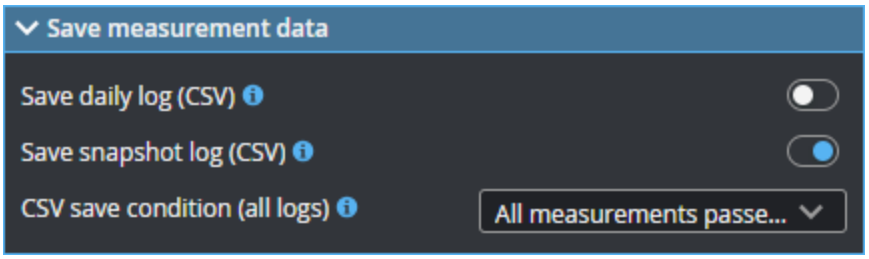
Settings

Save device data

Setting	Description
Input source	The source from which the tool takes data. The service does not support measurement inputs.
Save GoPxL recording (.gprec)	Saves the current frame to a GoPxL recording file.
Preview image format	The format the tool uses when saving a preview image of the current frame. All formats are compressed; some are lossy, others lossless. One of the following: • None • JPG (lossy, 95% quality) • PNG • TIFF uncompressed (16-bit lossless) • TIFF compressed (16-bit lossless). Note that with this option, saving frames will take considerably longer and may require lowering the frame rate. • Quarter resolution TIFF (8-bit lossless)
Data type	The data types in which a preview image is stored.
Export to 3rd party format	Exports the current frame's Surface data to another format. Note that on sensors, only PCL, Mountainsmap, and TIFF are available; to save to the other formats, you must run GoPxL on a PC or use a GoMax device. One of the following: • None • Wavefront OBJ (.obj) • ODSCAD OMC (.omc)

Setting	Description
	• PCL PCD (.pcd) • STL ASCII (.stl) • STL Binary (.stl) • MountainsMap SUR (.sur) • TIFF uncompressed (.tif)
Image and recording save condition	The condition that triggers saving frame data. Frames are saved in either an "OK" or "NG" folder in the folder defined in the **Storage mode** section. For more information, see *Storage mode* on page 1162. One of the following: **All measurements passed or failed (everything)** Always saves a frame, whether measurements pass or fail, but also if you haven't added any measurements. Frames will be saved to the "NG" folder if any enabled measurement returns a fail; frames are only saved to the "OK" folder if all enabled measurements pass, or if no measurement are enabled. **All measurements passed** **All measurements failed** Only saves a frame if all enabled measurements pass or fail, respectively. **Any measurements failed** Saves a frame if any of the enabled measures fail. **Specific measurement failed** Saves a frame when a selected measurement fails. Select the measurement in the **Measurement** drop-down. **Specific measurement range** Saves a frame only if the value of the measurement selected in **Measurement** is in the range defined by **Minimum** and **Maximum**; the range is inclusive.
Save OK/NG separate folders	Whether to save frame data in separate folders ("OK" and "NG"), or instead to the root folder.

You enable and configure CSV logs in the **Save measurement data** section.

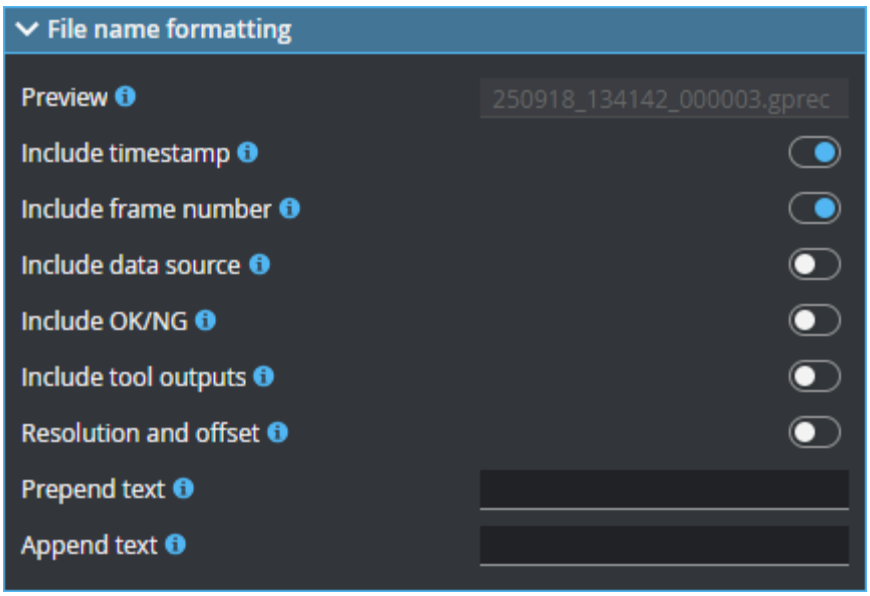


When either the daily or snapshot log is enabled, you can optionally configure the CSV file's export format by clicking the **Configure CSV** button; note that this feature is currently in beta.

Save measurement data

Setting	Description
Save daily log (CSV)	Saves a CSV log every 24 hours, from the time of the initial creation of the daily log. Each time a frame is acquired, GoPxL adds information relative to that frame to this log. For information on the log file format, see *Log Format* on page 1165.
Daily log period	The number of days the tool saves a log.
Save snapshot log (CSV)	Saves a CSV log for each new frame. For information on the log file format, see *Log Format* on page 1165.
All log (CSV) save condition	The condition used to save CSV logs (both daily and snapshot). For the available conditions, see *Image and recording save condition* on the previous page.

You configure the format of the file names in the expandable **File name formatting** section.



File name formatting

Setting	Description
Preview	An example file name resulting from the configured formatting, such as the following: `prependedText_250918_134142_000003_0.00_1.00_appendedText.gprec`

Include timestamp	Includes a date and time stamp in the filename using the following format:
	`ddmmyyy_hhmmss`
	For example:
	`250918_134142.gprec`

Include frame number	The six-digit number of the frame. This resets each time acquisition starts. For example: `000003.gprec`

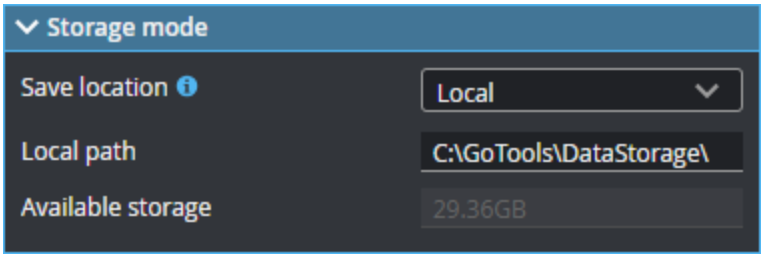
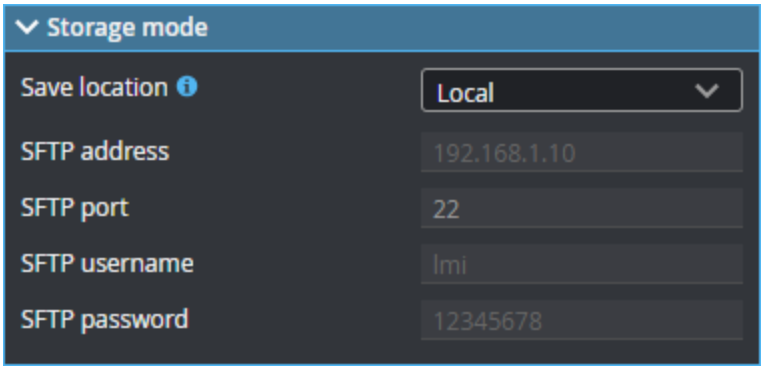
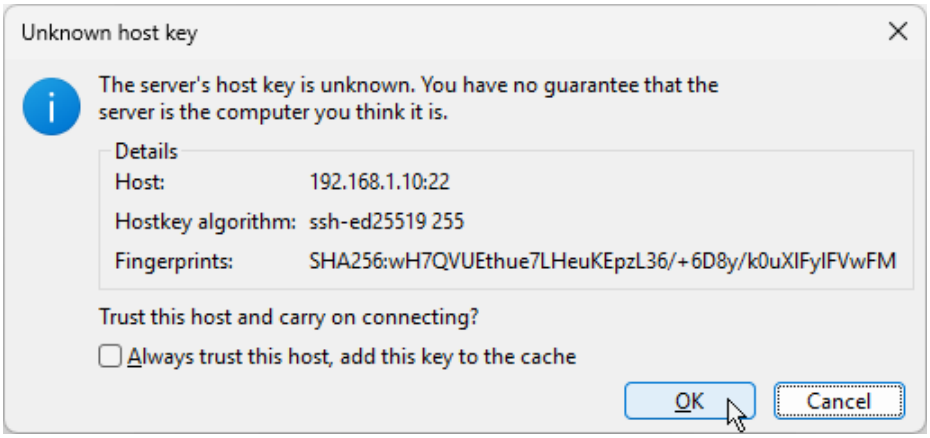
Include data source	The source selected in **Save device data** > **Input source**, using the following format:
	`{SOURCE_NAME} - {OUTPUT}`
	For example:
	`Gocator 0 - Top`
	`Surface Transform 1 - Transformed Surface`
	`Image Transform 1 - Transformed Image`

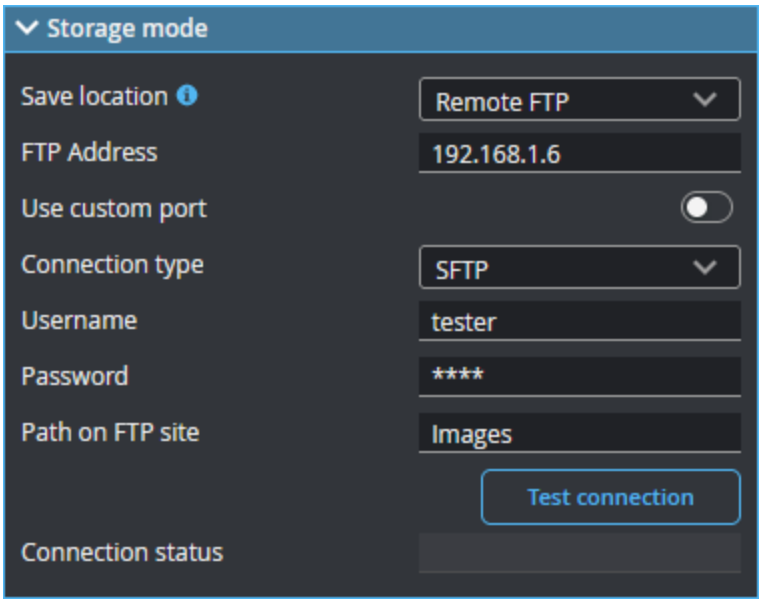
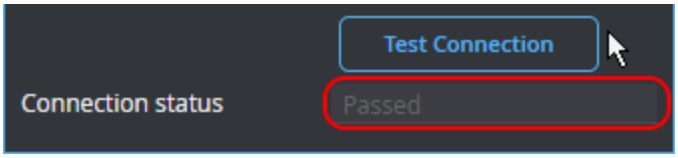
Include OK/NG	Adds "_OK" or "_NG" to the filename. Note that "_NG" will be added to the filename if any enabled measurement returns a fail. "_OK" will only be added to the filename if all enabled measurements pass, or if there are no measurements.
Include tool outputs	When this setting is enabled, the measurement value of the measurements you select in **Tool outputs selection** > **Measurement {n}** is added to the file names. Adjust **Tool outputs count** to include the measurement values of more than one measurement. Values are separated by an underscore character.
Resolution and offset	Adds the resolution and offset to the filename.
Prepend text Append text	Text added to the beginning of the file name, or at the end of the file name before the extension. Useful for adding information such as part numbers, production run, operator initials, and so on.

You configure the settings of the remote FTP/SFTP server in the expandable **Storage mode** section. If you are running GoPxL on a PC, you can also select and configure local storage.

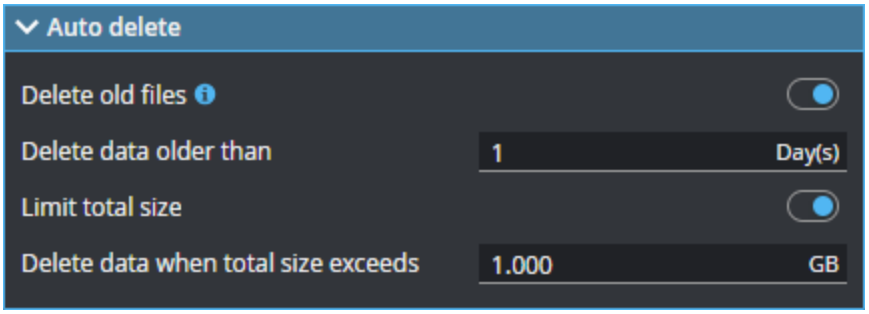
Storage mode

Setting	Description
Save location	Where the files are saved. The locations that are available depend on the device you are using or whether you are using GoPxL for PC. **Local**

Setting	Description
	Only available on GoMax devices, GoPxL for PC, and Gocator 2D Smart Cameras. When you select Local, on GoPxL for PC, files are saved to the local file system, using the path in **Local path**.  On GoMax and Gocator 2D Smart Cameras, files are saved using an SFTP server running on the device.  To access files on a GoMax or Gocator 2D Smart Camera, copy the SFTP server information (IP address of the device, port, and credentials) and connect to Gocator device's SFTP server using an FTP client application such as FileZilla or WinSCP. Note that the client application will probably display a message saying that the SFTP server's host key is unknown. Click OK to accept the connection. Typically you can also choose to always trust the host. 

Setting	Description
	Remote FTP When you select Remote FTP, the data storage service saves frames to a remote FTP server. You must install and then configure the FTP server on the PC connecting to the device, and fill in the fields in the **Storage mode** section using the IP address and optional port of the server, the connection type, and the user name and password, so that GoPxL can connect to that FTP server.  You should test the credentials and connection information by clicking **Test Connection**. If the connection succeeds, GoPxL displays "Passed" in **Connection status**. 

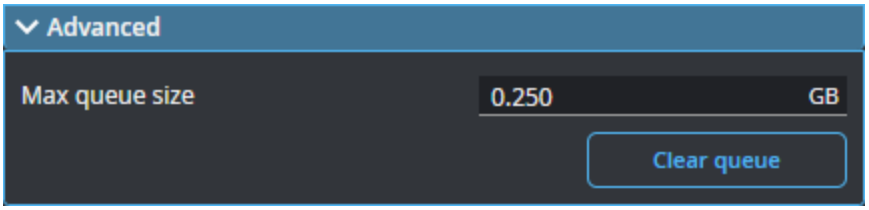
To ensure that the device receiving the frames doesn't run out of space, you can set limits in the expandable **Auto delete** section.



Auto Delete

Setting	Description
Delete old files	When this setting is enabled, the service deletes files older than the number of days you specify in **Delete data older than**.
Limit total size	When this setting is enabled, the service starts deleting files to keep the amount of data stored below the value you set in **Delete data when total size exceeds**. Before sending data, the service compares the space used with this limit, and deletes older files until the amount of stored data is below the limit. For this reason, the amount of data stored may exceed the limit until the next frame save.

You set advanced settings in the expandable **Advanced** section.



Advanced

Setting	Description
Max queue size	The maximum queue size for frames. Set this to a higher value if you need to run the device at a high frame rate to help ensure all frames are saved. Note however that the higher the value, the longer frames will continue to be saved after acquisition stops. Therefore, to avoid losing frames, you must keep the remote FTP server running, and the device saving frames powered up, until the frames in the queue have all been saved. You can clear the queue manually by clicking **Clear queue**.

Log Format

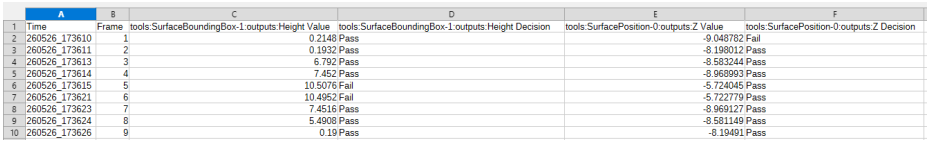
Logs are saved in the CSV (comma-separated values) format. The logs can be opened in spreadsheet applications or in text editors. When importing a log into a spreadsheet, make sure to set the separator to "comma."

For each frame, a row is output to the log file (daily log or "snapshot" log), containing the following columns:

Log format

Name	Description
Time	A timestamp representing the date and time of the log entry.
Frame	The frame number of the frame.
tools	One or more column pairs for the value and decision of an enabled measurement. For example, in the following, columns C and D contain the value and decision,

Name	Description
	respectively, of the Height measurement of a Surface Bounding Box tool.



	A	B	C	D	E	F
1	Time	Frame	tools:SurfaceBoundingBox-1:outputs:Height Value	tools:SurfaceBoundingBox-1:outputs:Height Decision	tools:SurfacePosition-0:outputs:Z Value	tools:SurfacePosition-0:outputs:Z Decision
2	260526_173610	1	0.2148	Pass	-9.048782	Fail
3	260526_173611	2	0.1932	Pass	-8.198012	Pass
4	260526_173613	3	6.792	Pass	-8.583244	Pass
5	260526_173614	4	7.452	Pass	-8.968993	Pass
6	260526_173615	5	10.5076	Fail	-5.724045	Pass
7	260526_173621	6	10.4952	Fail	-5.722779	Pass
8	260526_173623	7	7.4516	Pass	-8.969127	Pass
9	260526_173624	8	5.4908	Pass	-8.581149	Pass
10	260526_173626	9	0.19	Pass	-8.19491	Pass

You can configure the CSV output by clicking the **Configure CSV** button. Note that this feature is currently in beta.

Health

The Health pane provides statistics related file saving, queue, and memory.

Health

Name	Description
Total files saved	The number of files saved to the configured storage.
Received frame count	The number of frames the Data Storage service has received.
Queue usage	How much of a queue's available capacity is currently being used. Useful for knowing how many more frames can be processed before the system is out of storage or memory.
Queue size	The amount of memory currently used by queue, in bytes.
Files auto deleted	The number of files deleted by auto delete; for more information, see *Auto Delete* on the previous page.
Out of storage or memory drops	The number of frames dropped due to memory or storage limits.
Frames failed to save	The number of frames that were not saved. This can happen, for example, when there are connectivity issues to a remote SFTP server, or hardware issues when saving locally.

Monitor - Measurement Stats, Health & Performance

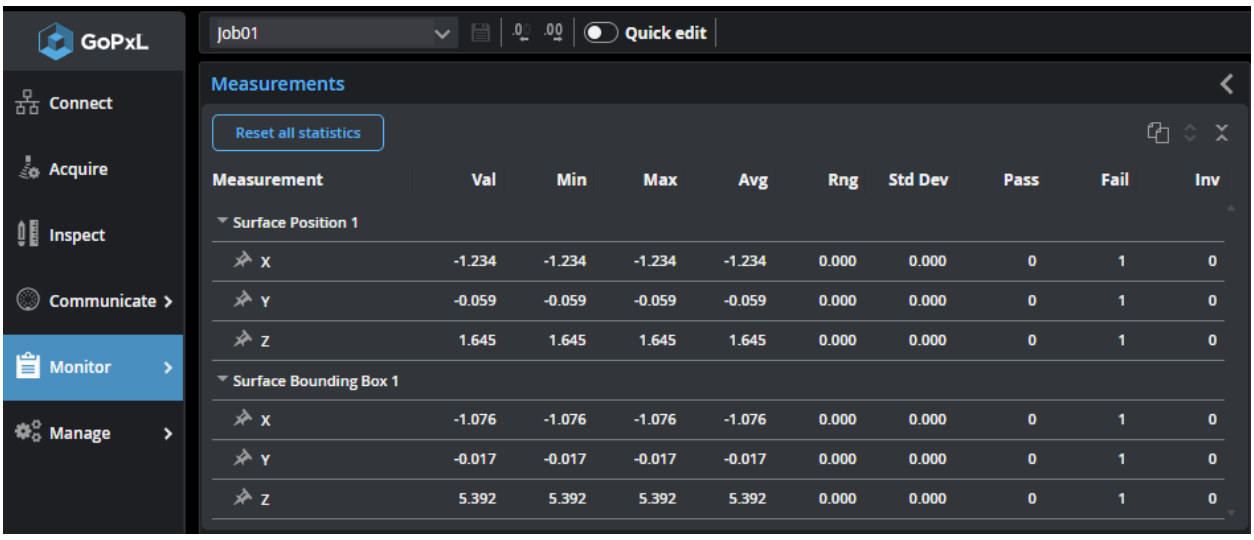
The **Monitor** category summarizes sensor and system health information and provides tool and measurement statistics. It also provides tool performance statistics, and audit logging configurations.

Available report pages

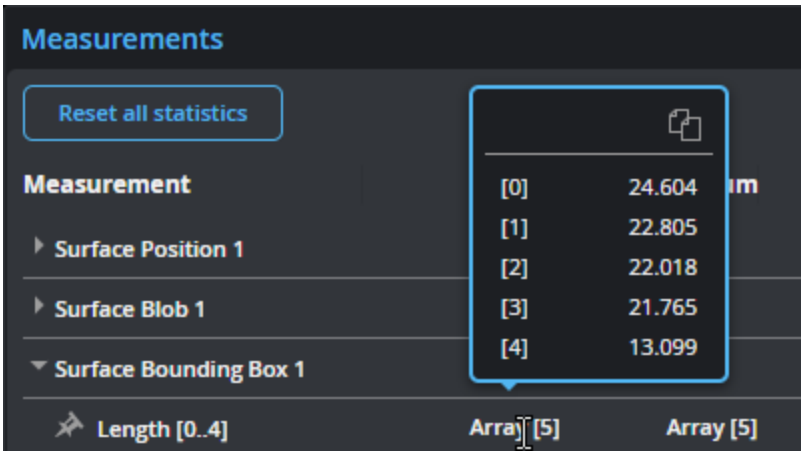
Report	Description
Measurements	The **Measurements** page provides statistics for each enabled measurement. For more information, see *Measurements* below.
Health	The **Health** page lets you monitor system and sensor health indicators, as well as various counters, such as processing drops. For more information, see *Health* on the next page.
Performance	The **Performance** page provides performance statistics. For more information, see *Performance* on page 1171.
Audit Log	The **Audit Log** page lets you configure remote logging. For more information, see *Audit Log* on page 1172.

Measurements

The **Monitor** > **Measurements** page displays statistics for each measurement enabled in the **Inspect** page, grouped by the tool that contains the measurement. Values are reset when the sensor is started. To reset statistics manually, click the **Reset All Statistics** button.



For arrayed measurements, hover over "Array [n]" to see values in the array. Click the Copy icon to copy the values to the clipboard.



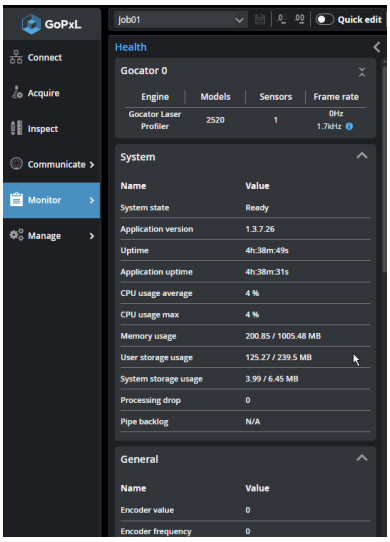
For each measurement, the following information is available:

Measurement statistics

Name	Description
Value	The most recent measurement value.
Minimum	The minimum measurement value that has been returned.
Maximum	The maximum measurement value that has been returned.
Average	The average of all measurement values collected since the sensor was started.
Range	The difference between Maximum and Minimum.
Std dev	The standard deviation of all measurement values collected since the sensor was started.
Pass count	The number of pass decisions the measurement has generated.
Fail count	The number of fail decisions the measurement has generated.
Invalid count	The number of frames that returned no valid measurement value.

Health

In the **Health** panel, you can monitor system, sensor group, and sensor health indicators, as well as various counters, such as processing drops.



The following tables list the information available in the Health panel.

> When a sensor is being accelerated (by a PC instance of GoPxL or by GoMax), the value reported in the System section comes from the accelerator. However, only the following metrics are appreciably different from the physical device when accelerated: Uptime, CPU Usage, Memory Usage, and System/User Storage Usage.

System

Name	Description
System state	Current system state indicating whether the device is scanning or not.
Application version	System software version.
Uptime	Time since the system was power-cycled or reset. (If the sensor is accelerated by a PC or a GoMax accelerator, this is the uptime of the PC or GoMax.)
Application uptime	Time since GoPxL started. (On a PC, indicates how long the PC has been running.)
CPU usage average	Average system CPU usage for the system.
CPU usage max	Maximum system CPU usage for the system.
Memory usage	Sensor memory (MB used / MB total available). When a sensor is accelerated, this displays N/A.
User storage usage	Sensor flash storage (MB used / MB total available). When a sensor is accelerated, this displays N/A.
System storage usage	Non-volatile storage available (MB used / MB total available).
Processing drop	The sum of various indicators related to processing drops including drops due to insufficient CPU and buffer overflows.
Pipe backlog	Number of bytes queued for processing.

General

Name	Description
Encoder value	Current encoder value (ticks).
Encoder frequency	Current encoder frequency (Hz).
Processing latency	Last delay from camera exposure start to when the results are ready for output.
Processing latency (Max)	Peak delay from camera exposure start to when the results are ready for output.
Scan count	Number of scans performed since system state last changed to Running.
Scanner drops	Number of sensor group (scanner) drops.
Current speed	Current scanning frame rate.
Trigger drops	Count of camera frames dropped due to excessive trigger speed.
CUDA in use	With G3 sensors accelerated by PC or GoMax, indicates whether GoPxL is using CUDA.

For each sensor in a system, GoPxL displays a table in the expandable **Sensors** section listing the health indicators for that sensor.

Sensor {n} health

Name	Description
Master connected	Whether the sensor is connected to a Master sensor networking hub.
Laser safety	Whether Laser Safety is enabled. With snapshot and confocal line sensors, enabling safety is not required in order to scan.
Sensor uptime	Time since the sensor was power-cycled or reset.
CPU usage average	Average system CPU usage for the sensor.
CPU usage max	Maximum system CPU usage for the sensor.
Ethernet receive rate	Rate at which data is being received by the sensor over Ethernet (Mbps).
Ethernet send rate	Rate at which data is being sent to the sensor over Ethernet (Mbps).
Ethernet link speed	Speed of the Ethernet link (Mbps).
Memory usage	Sensor memory usage (MB used / MB total available).
User storage usage	Sensor flash storage usage (MB used / MB total available).
System storage usage	Non-volatile storage available (MB used / MB total available).
Max spot count	Maximum available spots, across all cameras. (G2)
Total exposure time	Total amount of time the laser light (G2), projector (G3), or light source (G5) has been turned on to make scans.
Internal temperature	Internal sensor temperature (degrees Celsius).
Laser driver temperature	Laser driver control temperature (degrees Celsius). (G2)
Light driver temperature	Light driver control temperature (degrees Celsius). (G4 / G5)

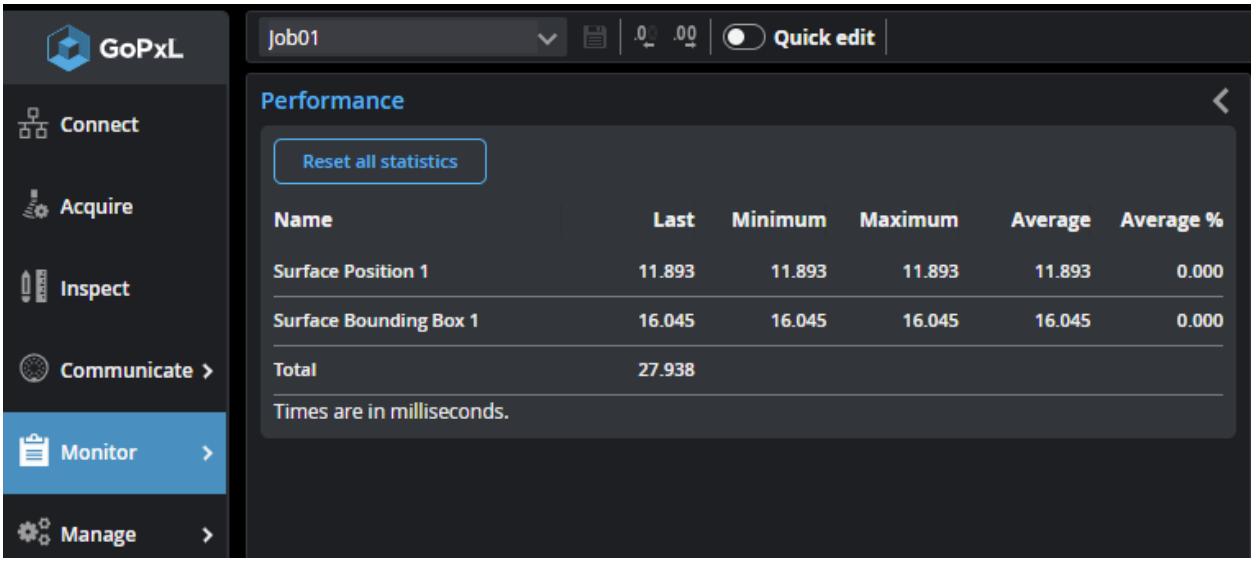
Name	Description
LED PCB temperature	LED PCB temperature (degrees Celsius). (G4 / G5)
External thermistor temp.	External thermistor temperature (degrees Celsius). (G4 / G5)
Projector illumination temp	Projector temperature (degrees Celsius) (G3).
Trigger drops	Count of camera frames dropped due to excessive trigger speed.
Camera buffer drops	Count of camera buffer drops due to excessive trigger speed. For G3, the buffer drops are shown for each camera.
Digital Output {n} drops	Count of digital output drops because the output signal could not be generated.
Digital Output {n} high count	Number of times the digital output line was set to high.
Digital Output {n} low count	Number of times the digital output line was set to low.

Control

Name	Description
GDP drop	The number of Gocator protocol drops.
Data storage drop	The number of Data Storage drops .

Performance

The **Monitor** > **Performance** page displays performance statistics (execution time in milliseconds) for each tool added in the **Inspect** page.



For each tool, the following information is available:

Performance Statistics

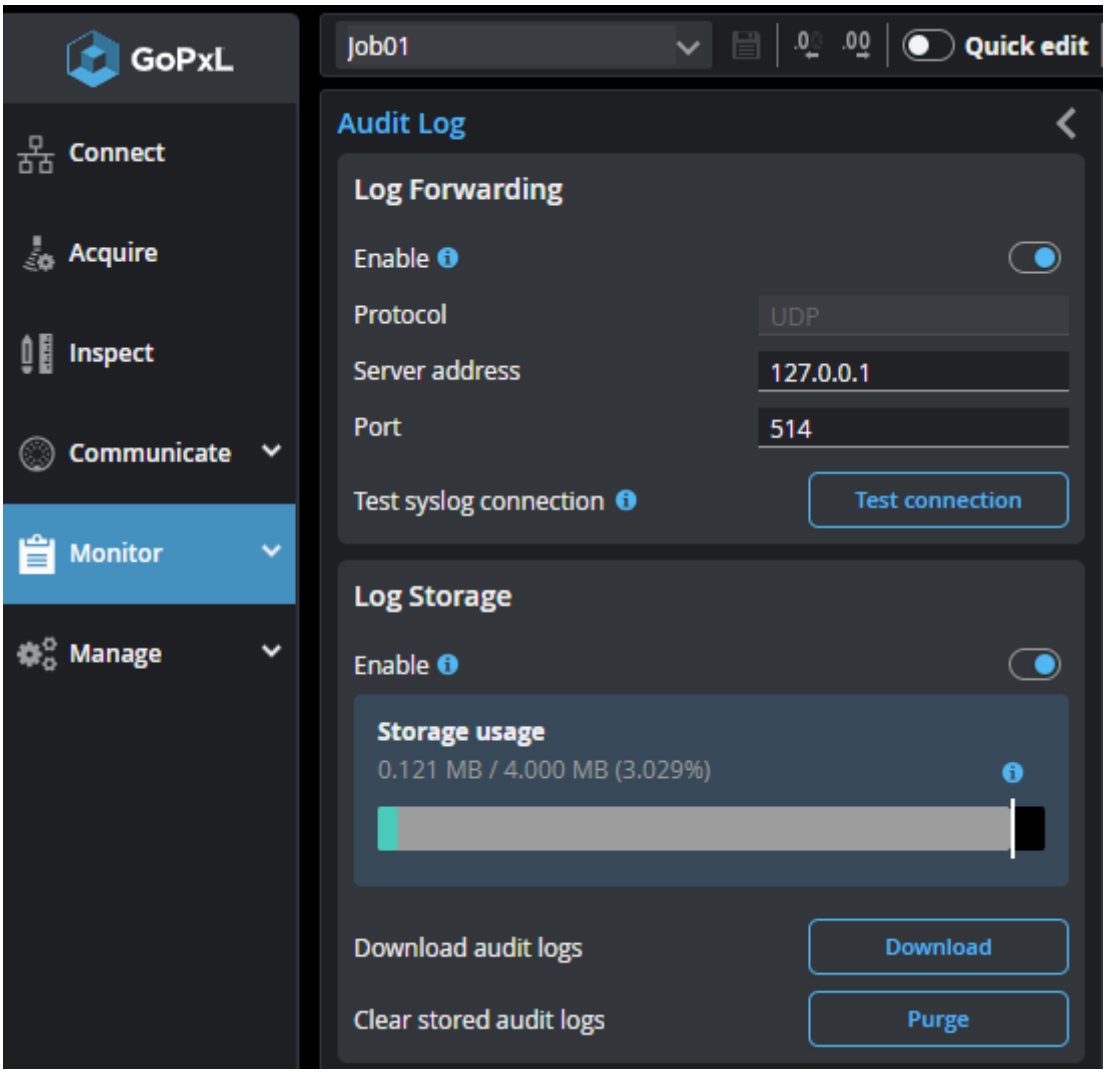
Name	Description
Last	The latest execution time of the tool.
Minimum	The minimum execution time of the tool.
Maximum	The maximum execution time of the tool.
Average	The average execution time of the tool.
Average %	The average percentage of the execution time of the tool.

Audit Log

The **Monitor** > **Audit Log** page lets you configure audit log forwarding to a syslog server, as well as local log storage. This is useful for compliance requirements in your application.

After log forwarding is enabled, GoPxL forwards event information for all change to configurations or operations in GoPxL, in real-time using the standards described in RFC 5424; for information on the standard, see https://datatracker.ietf.org/doc/html/rfc5424.

Audit log forwarding currently only supports the UDP protocol; additional transport protocols will be added in future releases.



Log forwarding settings

Setting	Description
Enable	When enabled, audit logs are forwarded in real-time to the configured syslog server.
Protocol	The protocol used for transmission. Currently only UDP.
Server address	The IP address of the syslog server.
Port	The port on which the syslog server is available. The default is 514; change this if you use a different port.
Test syslog connection	Click to test whether the syslog connection is properly configured.

You can also store logs, in the JSON format, in GoPxL's local storage, and download them to evaluate them. Use log storage with caution on sensors, as these devices have limited Flash write life. Make sure to clear the audit logs occasionally by clicking **Purge**. Storage is capped at 95% of 4 MB.

You may need to refresh the web interface page to get an accurate value for used storage.

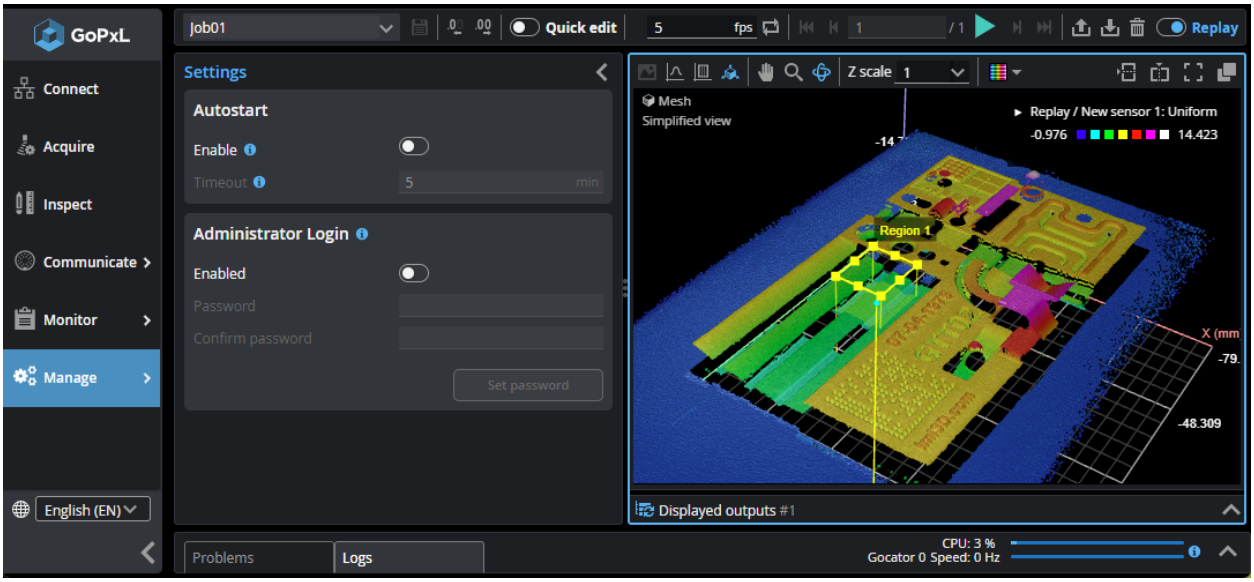
Log storage

Setting	Description
Enable	When enabled, logs are saved locally on the device.
Download	Downloads the audit log in JSON format.
Purge	Deletes the internal GoPxL audit log.

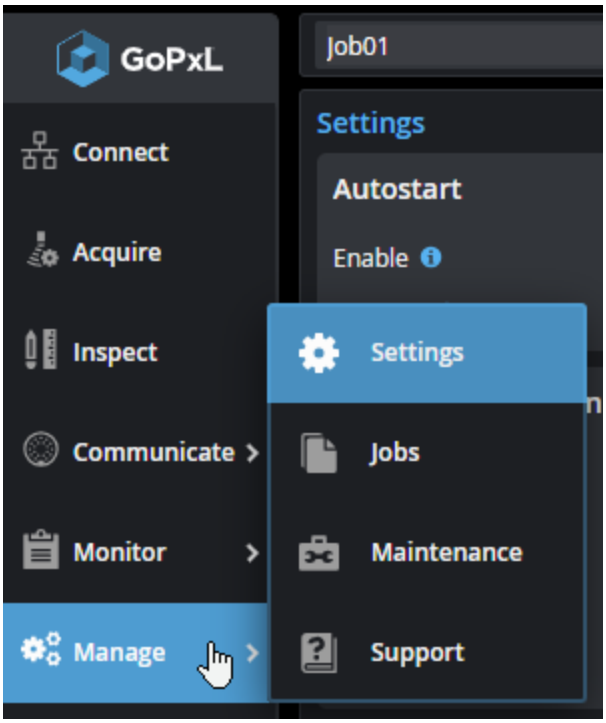
For information on file name formats, see *File Name Conventions* on page 78.

Manage - Sensor Management and Maintenance

You use the pages in the **Manage** category to perform networking, management, and maintenance tasks (such as software updates and sensor factory restores).



To access a specific management page, hover the mouse pointer over **Manage** and choose the required page.



Manage Page Name	Description
Settings	The **Settings** page lets you configure the network settings of a sensor. For more information, see *Settings* below.
Jobs	The **Jobs** page lets you create, delete, and manage jobs. For more information, see *Jobs* on page 1179.
Maintenance	The **Maintenance** page lets you upgrade a device's software, perform a backup or restore of a device, perform a factory restore of a device, or restart it. For more information, see *Maintenance* on page 1183.
Support	The **Support** page provides software version information and lets you download a support file for troubleshooting; it also lets you upload a support file to GoPxL. You can also access and download the GoPxL documentation, download the SDK, and go to LMI's online support page. For more information, see *Support* on page 1186.

Settings

The **Settings** page under **Manage** provides autostart and administrator login settings.

Autostart

If your application requires maintaining uninterrupted scanning, you can configure GoPxL to automatically restart sensors that are running after various events, by enabling **Autostart**. Unaccelerated single sensors and sensors grouped with a controlling sensor, as well as sensors accelerated by GoMax or a PC instance of GoPxL, will automatically start scanning in the situations listed below. The setting persists after restarts, and is a global setting (that is, not saved in a job). Performing a factory restore disables autostart.

- "Launch" events:
  - Power cycle
  - Restarting a GoMax unit or relaunching a PC instance of GoPxL that is accelerating sensors
  - Restoring a running sensor with a backup in which Autostart is enabled
- Network disconnections with accelerated sensors (such as when a network cable is disconnected or a switch is power cycled)



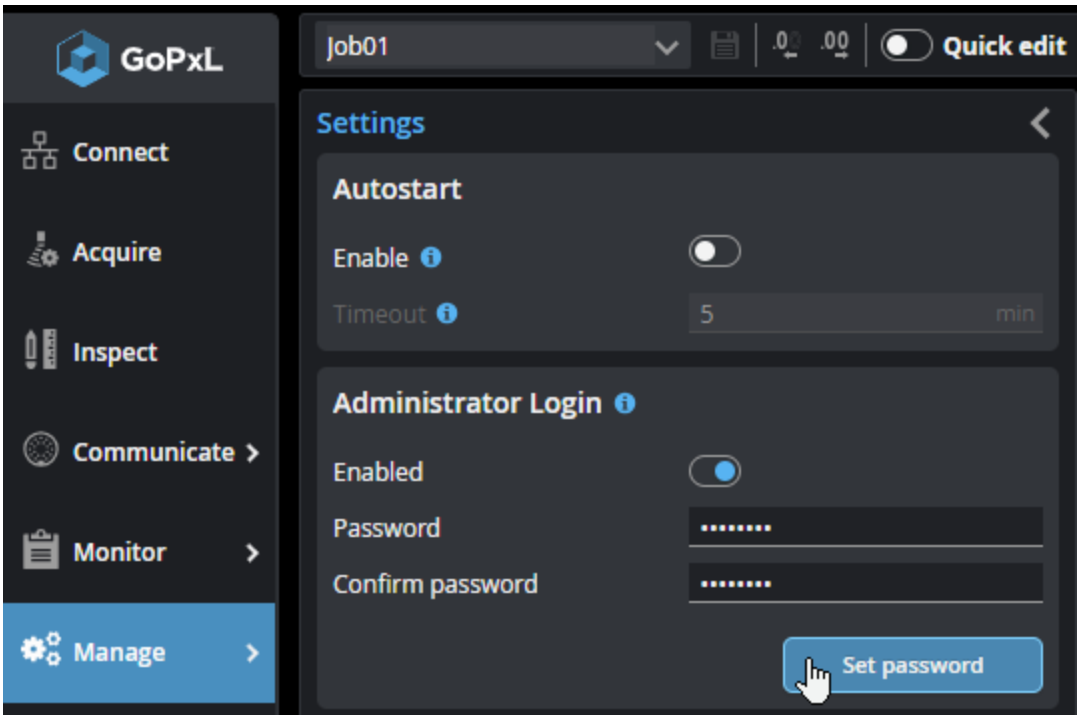
When a sensor is accelerated by GoMax or GoPxL on PC, only enable **Autostart** on the accelerating device. Similarly, if a sensor is grouped with other sensors, only enable **Autostart** on the host sensor (the sensor to which you connected other sensors in GoPxL). Otherwise, timing issues can result in.

Administrator Login

You can prevent unauthorized access to sensors by setting an administrator password. When a password is set, GoPxL displays a password field when the interface loads. To continue you must provide the password. In a password-protected system, there are no limitations in terms of functionality.



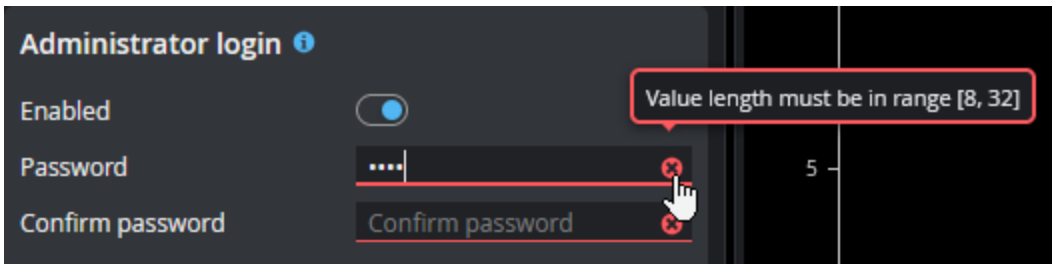
You set the administrator password on the **Settings** page:



⚠	On a sensor, if you forget the password or want to completely clear a password, you must perform a factory restore. You can do this in GoPxL from **Manage** > **Maintenance** (see *Factory Restore* on page 1185) or from the Discovery tool (see *GoPxL Discovery Tool* on page 1290). When GoPxL is running on a PC, in both you must delete the password-protected instance and create a new instance.

Note the following:

- Passwords must be 8 to 32 characters long.
- All characters are accepted.
- The password can't be empty.
- There are otherwise no constraints or checks on passwords (that is, no minimum number of special characters, no combination of uppercase / lowercase, and so on).
- After setting a password, the password is stored even if administrator login is disabled. If you re-enable the password, the previously set password is active.
- You don't need to know the current password to change the password, so anyone with access to a logged-in sensor can change the password.
- If the password is not accepted by GoPxL, hover over the red "x" icon ( ) for more information.



Date and Time

GoPxL lets you set the date and time either manually (to a specific date and time or to the client PC's current date and time), or using a Network Time Protocol (NTP) server. Setting the date and time correctly is useful in conjunction with other GoPxL services or tools to include the correct date and time, for example:

- In file names of files saved using the Data Storage service, after enabling timestamp in that service; for more information, see *Data Storage Service* on page 1158.
- In Script tools.

The service works on all Gocator devices. Manually set date and time is not kept after restarting a device or performing a factory restore.

The service does not support Daylight Savings Time.

Note that to use an NTP server, you typically will set up an NTP server on the PC (the client) connecting to the device. For accurate date-time, the NTP server needs access to the internet. You may also need to configure the firewall of the client PC to open UDP port 123.

Also, time-related functions in a Script tool return the UTC+00:00 time, ignoring the time zone set in **Time zone**.

The following settings are available:

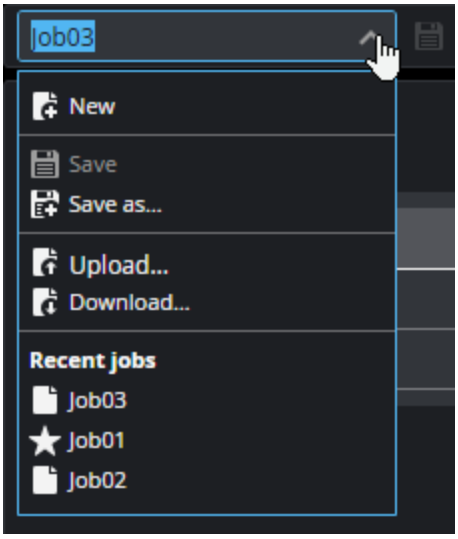
Date and Time Settings

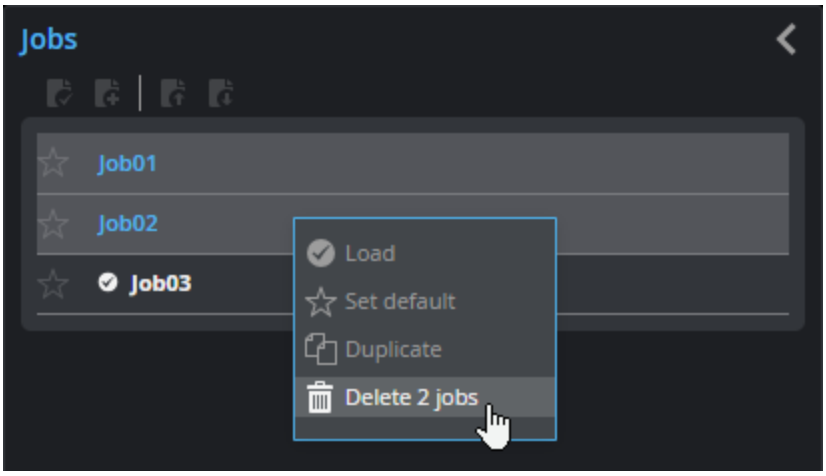
Setting	Description
Time source	One of the following: **Manual**: Lets you set the time manually or to the client PC's date and time, in **Set date & time**. **Network Time Protocol**: Lets you synchronize time to an NTP server.
Time zone	The time zone GoPxL uses to calculate the time.
NTP server	The name or IP address of the NTP server. Only displayed when **Time source** is set to Network Time Protocol.
Set date & time	The manual date and time. Only displayed when **Time source** is set to Manual.
Sync and Save	Synchronizes and saves the NTP server address and time zone setting.
Save	Synchronizes and saves the manually set date and time, and the time zone settings.

Jobs

The **Jobs** page lets you manage jobs.



	Element	Description
1	Job-related operations on global toolbar	Job menu and Save button. Use the job menu to create new jobs, save jobs, load recent jobs, and also upload or download jobs between the PC and the sensor. You can rename the currently loaded job from the job name field, simply by changing it and pressing enter or clicking outside the field.  When a job has unsaved changes, it is displayed in italics with an asterisk. Click the Save button in the global toolbar to save it. A star icon indicates the job that has been set as the default.
2	Job-related operations	These controls let you load or reload jobs, create new jobs, and upload or download jobs between the PC and the sensor. For explanations of the job related operations, see *Job-related operations* on the next page.
3	Job list	The list of available jobs. You can rename a job by double-clicking it in the list. When you move the mouse pointer over a job in the list, additional tools are displayed that let you rename the job, duplicate it, or delete it. When a job has unsaved changes, it is displayed in italics with an asterisk. Click the Save button in the global toolbar to save it. You can select multiple jobs in the list using Ctrl-click and Shift-click; when you right-click on one of the selected jobs, you can delete all the selected jobs.

	Element	Description
		
4	Default job indicator	Clicking the default job indicator ( ) sets the job as the default. This job will load when the device starts, or if the device is running through a PC instance of GoPxL, when the instance starts.
5	Loaded job indicator	The loaded job indicator ( ) shows which job is currently loaded.

The following table describes the job-related operations.

Job-related operations

Operation	Description
	Loads or reloads the job that is currently selected in the job list.
	Creates a new job called "untitled {n}" and loads it.
	Saves the current job. (Disabled if the job contains no changes.)
	Uploads or downloads a job between GoPxL and the client computer. Job files have a .gpjob extension.
	Lets you edit the name of the job under the mouse pointer. You can also double-click on the job's name to stat editing it.
	Duplicates the job under the mouse pointer.
	Deletes the job under the mouse pointer.

Characters Allowed in Job Names

When setting a job's name, you can use alphanumeric characters (numbers and letters), including accented characters in languages other than English and characters in languages such as Chinese

and Japanese. Some special characters are also allowed.

The following special characters are also supported:

Allowed Special Characters

space	& (ampersand)
- (hyphen)	( ) { } [ ] (parentheses, brackets, and braces)
_ (underscore)	+ (plus)
. (period)	, (comma)
' (single quote)	; (semi-colon)
@ (at)	= (equals)
# (pound)	

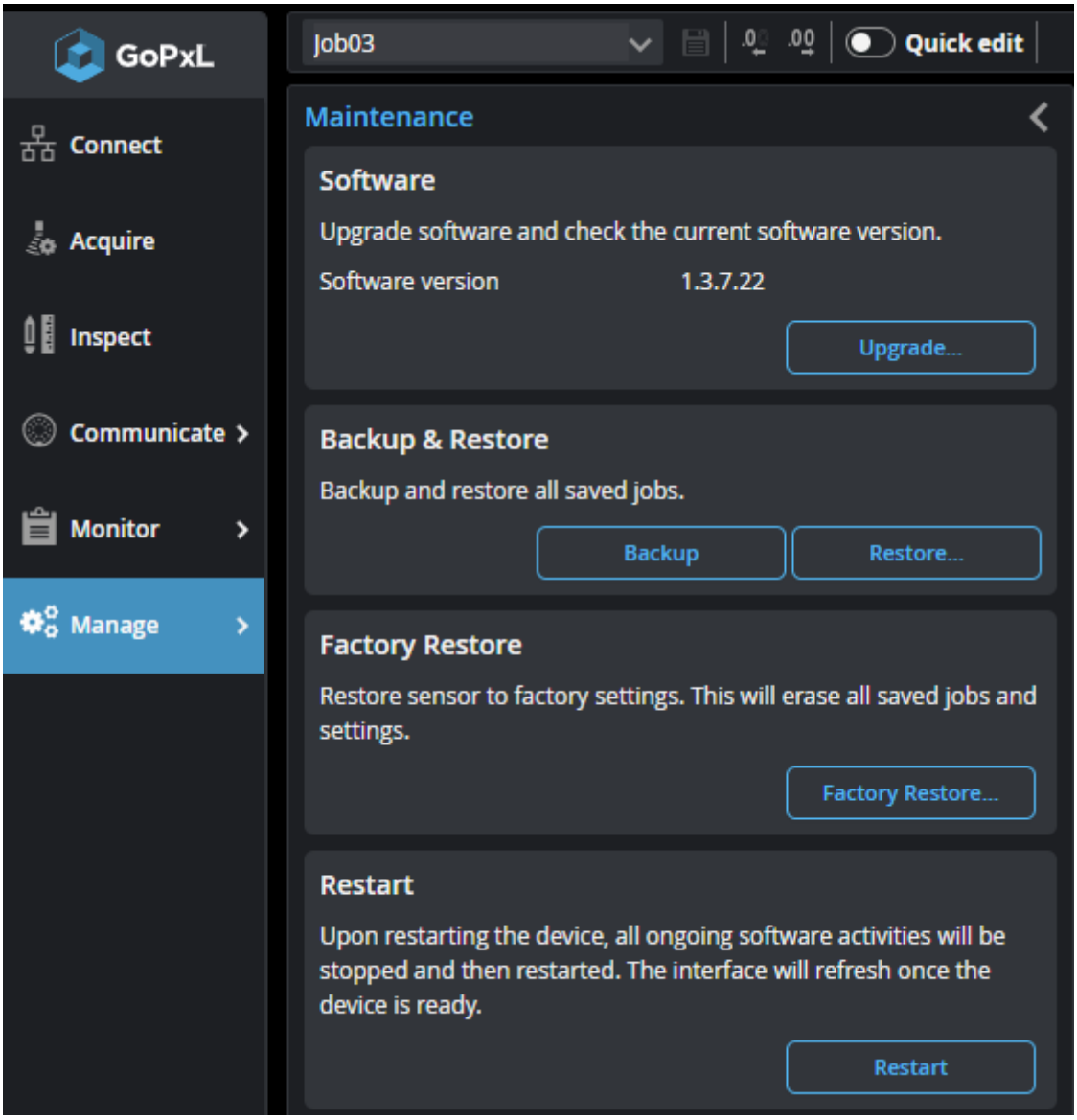
The following special characters are *not* supported:

Disallowed Special Characters

< (less than)	| (vertical bar or pipe)
> (greater than)	? (question mark)
: (colon)	* (asterisk)
" (double quote)	% (percent)
/ (forward slash)	~ (tilde)
\ (backslash)	$ (dollar)

Maintenance

You use the **Maintenance** page in the **Manage** category to perform basic maintenance.



The **Maintenance** panel lets you do the following:

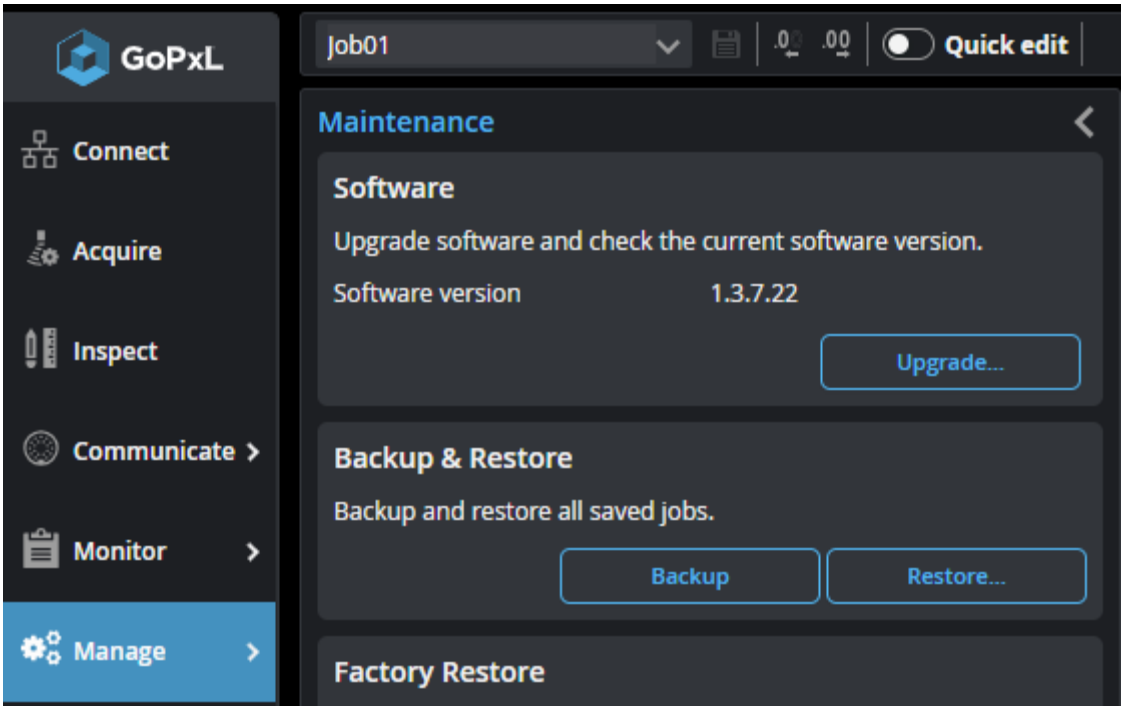
- **Upgrade...**: Upgrade the device's software. Note that you can't upgrade a sensor if you are running it through a PC instance of GoPxL. For more information on upgrading a sensor's software, see *Software Upgrade* on the next page.
- **Backup** / **Restore...**: Back up and restore all saved jobs and recorded data. Note that backing up and restoring on a PC instance of GoPxL only affects the PC instance, not the sensor. For more information, see *Backup and Restore* on the next page.

- **Factory Restore...**: Restore GoPxL to factory defaults; this applies to sensors and PC instances of GoPxL. This erases *all* saved jobs and settings, as well as files created by tools. Performing a factory restore on a PC instance of GoPxL connected to a sensor *only* affects the PC instance, not the sensor; doing this also disconnects the sensor from the PC instance.
- **Restart**: Restart the sensor. Note that you can't restart a sensor running through a PC instance of GoPxL. To do this, you must remove the sensor from the PC instance; for more information, see *Stopping Acceleration* on page 1203.

Software Upgrade

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

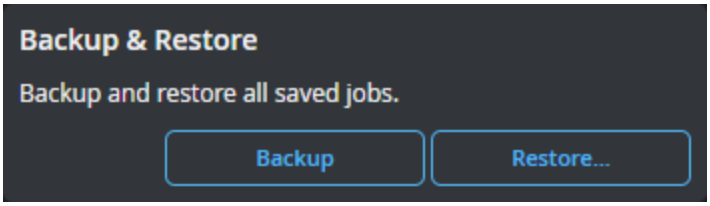
Note that if you are running a sensor through a PC instance of GoPxL, you can't upgrade its software. To do this, you must temporarily remove it from the sensor group in the PC instance, access the sensor directly at its IP address, and perform the upgrade. You can then add it back to the sensor group on a PC instance. (Note that it must be the same software version.) For more information, see *Stopping Acceleration* on page 1203.



Backup and Restore

Sensor backups let you keep the state of a sensor or system in a safe location, and restore a sensor to a previous state in case you have made unwanted changes. You create backups in the **Maintenance** page. Backup files contain the following:

- All jobs
- Current recorded data
- Layout set up on the **Connect** page
- Alignment performed on the **Acquire** page
- Other configurations such as scan settings and control settings



> You should always create a sensor 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.

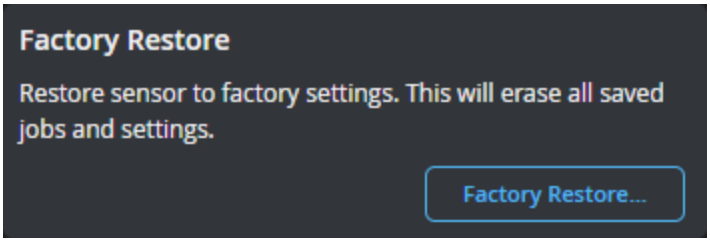
When you create a backup, your browser may ask if you want to allow downloading the file.

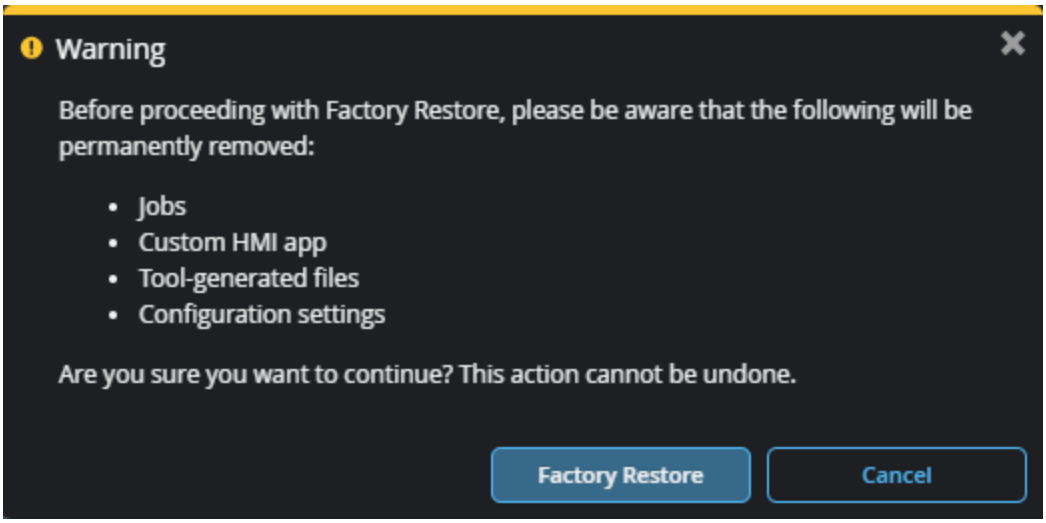
If you are running a sensor through a PC instance of GoPxL, performing backup, restore, and factory restore operations only affects the PC instance of GoPxL: jobs and recorded data saved on the sensor are not backed up and are not overwritten by a restore.

For information on file name formats, see *File Name Conventions* on page 78.

Factory Restore

You can perform a factory restore to quickly remove all jobs and configurations, as well as the files created by tools (such as Surface Pattern Matching, Surface Track, and so on). This also removes Custom HMI apps.



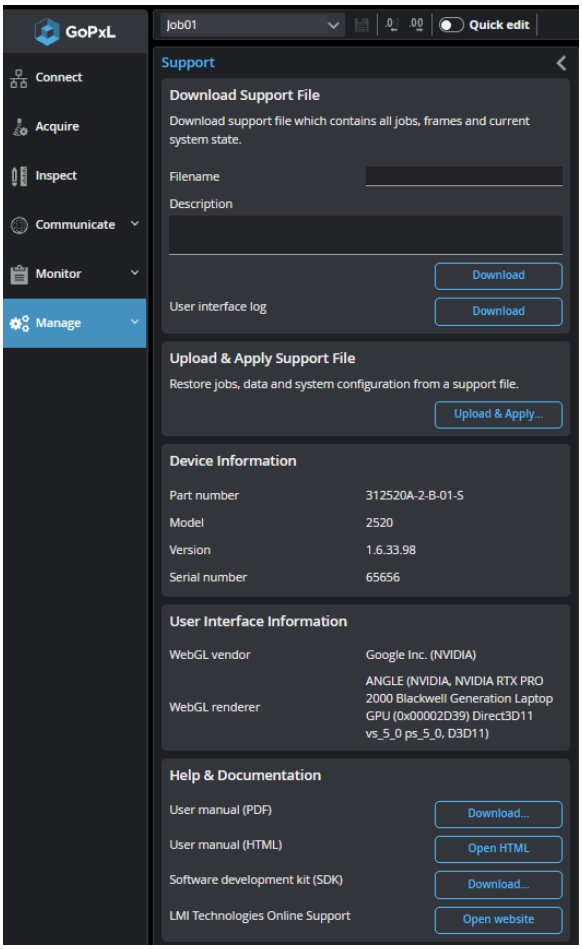


You should create a backup file of your sensor in case of unwanted changes and in the unlikely event that a sensor fails and a replacement sensor is needed. If this happens, the new sensor can be restored with the backup file.

Support

The **Support** page in the **Manage** category lets you download or upload a support file and get software version information. Support files are useful for supporting development and diagnosing issues. Additionally, you can download a user interface log; you would normally only need to download the log if told to do so by LMI's support team.

You can also access the user manual, download the SDK, and go to online support from here.



Support files contain everything in backup files, but also contain *additional diagnostic information*:

- All jobs
- Current recorded data
- Layout set up on the **Connect** page
- Alignment performed on the **Acquire** page
- Other configurations such as scan settings and control settings

For information on file name formats, see *File Name Conventions* on page 78.

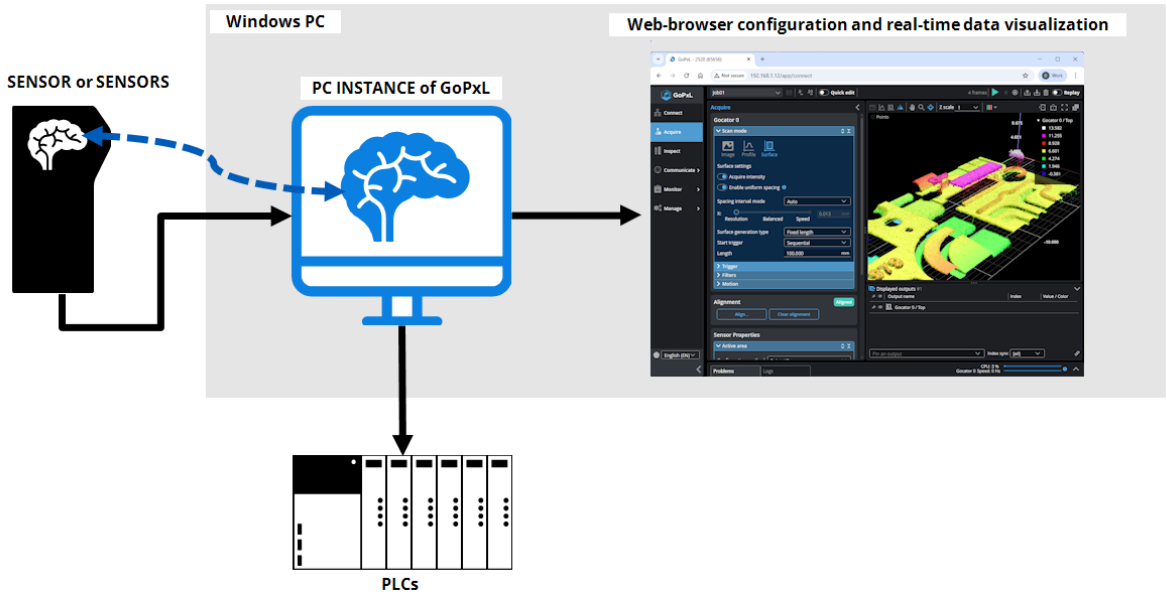
Running GoPxL on a Windows PC

GoPxL is the software that runs on a Gocator device. You typically connect directly to a sensor's IP address via a web browser, and use the GoPxL software running on the device to configure and run it. (By default, a sensor's IP address is 192.168.1.10. If you can't find a sensor, see *GoPxL Discovery Tool* on page 1290.)

You can also run GoPxL on a Windows PC. We call GoPxL running on a PC a "PC (or local) instance of GoPxL."

A PC instance of GoPxL lets you do the following:

- Accelerate sensors: For applications where performance is critical, you should run a sensor through a PC instance of GoPxL. After you connect a sensor to GoPxL running on a PC, the instance automatically starts accelerating it, handling the processing of the scan data, rather than the sensor. This improves system performance, potentially letting you increase scan rates or add more tools. For more information, see *Accelerating a Sensor* on page 1200.



- Load previously recorded scan data to examine it or to configure measurement tools offline, without a sensor, using that data. For more information, see *Loading Scan Data* on page 1204.

> For information on the ports a PC instance uses (for example, in order to ensure ports are not blocked over your network), see *Required Ports* on page 45.

If your Windows account has Administrator rights, you can install a PC instance of GoPxL as a Windows service. A PC instance installed (and running) as a service can run a sensor without the

need for a user to be logged into Windows. For more information, see *Running GoPxL Manager in Administrator Mode* on page 1192.

The user interface of GoPxL running on a sensor and GoPxL running on a PC is almost identical. When running a sensor through a PC instance of GoPxL, you can't upgrade or restart the sensor. You must remove the sensor from the PC instance and then connect to the sensor directly to upgrade the sensor.

> Job files you create in a PC instance of GoPxL are separate from any jobs that are on the sensor. Be sure to back up jobs in a PC instance if ever you need to run them directly on the sensor.

System Requirements

The following are the minimum system requirements for using GoPxL on a Windows PC:

- Processor: x86-64 (Core i3 or equivalent)
- RAM: 8 GB
- Hard drive: 64 GB
- Operating system: Windows 10 or 11 (64-bit only)
- Display: 1366 x 768

To accelerate more sensors or to run the system at higher speeds, use a computer with more system resources.

When using GoPxL on a Windows PC to accelerate a G3, the PC instance 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.
- NVIDIA graphics driver 576.57 or higher. LMI recommends using the latest driver version available and downloading the driver directly from the NVIDIA website to get the driver, rather than using the Windows Device Manager.

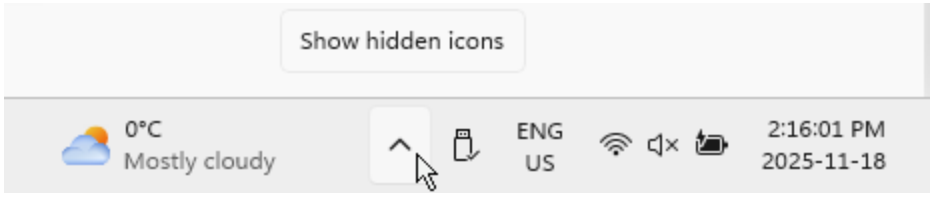
To exceed the capabilities of GoMax ORIN in terms of training times and inference, an RTX 4070 or higher is recommended.If you choose to run this tool on a PC instance of GoPxL, the PC must have a graphics card with a recent Nvidia GPU that supports Nvidia GeForce Game Ready Driver version 576.57 or higher. To verify the driver version of your video card, launch the NVIDIA Control Panel, and in the main panel of the application, make note of the driver version.

Launching GoPxL on a Windows PC

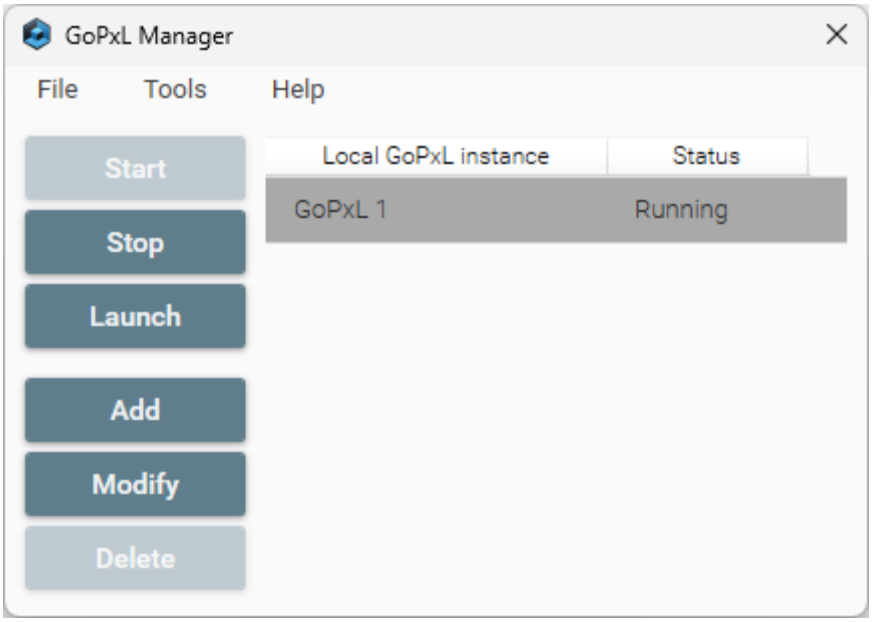
The first time you launch GoPxL for Windows, a GoPxL instance is automatically created and started, and the GoPxL interface opens in your default browser.

GoPxL Manager starts minimized in the Windows system tray.

Click the "Show hidden icons" button to find GoPxL Manager.



You use the GoPxL Manager application (which is started when you start GoPxL.exe) to add, remove, and configure instances. The following assumes you have connected a sensor to a network and that you can access its web interface.



The GoPxL Manager application. An instance is automatically added the first time you launch GoPxL.

To launch GoPxL on a PC

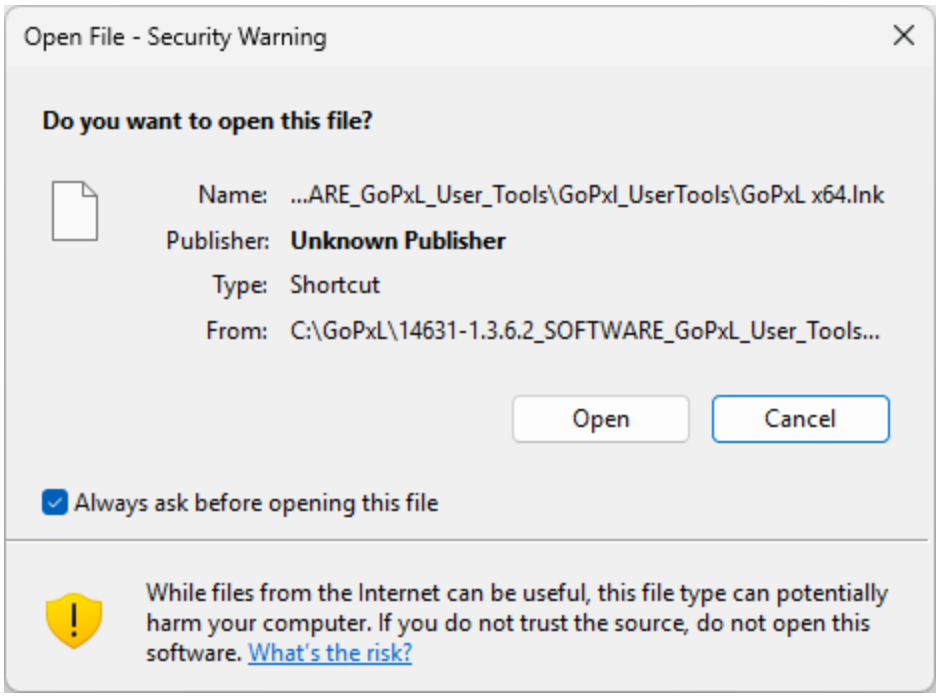
1. Download and unpack the GoPxL Utilities package (`14631-x.x.x.x_GoPxL_Utilities.exe`) by double-clicking on it.

   If you are using GoPxL on PC to accelerate a sensor, the version of the package must match the version of GoPxL on the sensor.

2. From the `GoPxL` folder, launch `GoPxL x64`.

   GoPxL Manager creates a local, PC-based instance of GoPxL and launches it in a new tab in your default browser, using IP address 127.0.0.1.

   Windows may display a security warning asking if you want to launch GoPxL. Click **Open** to continue.



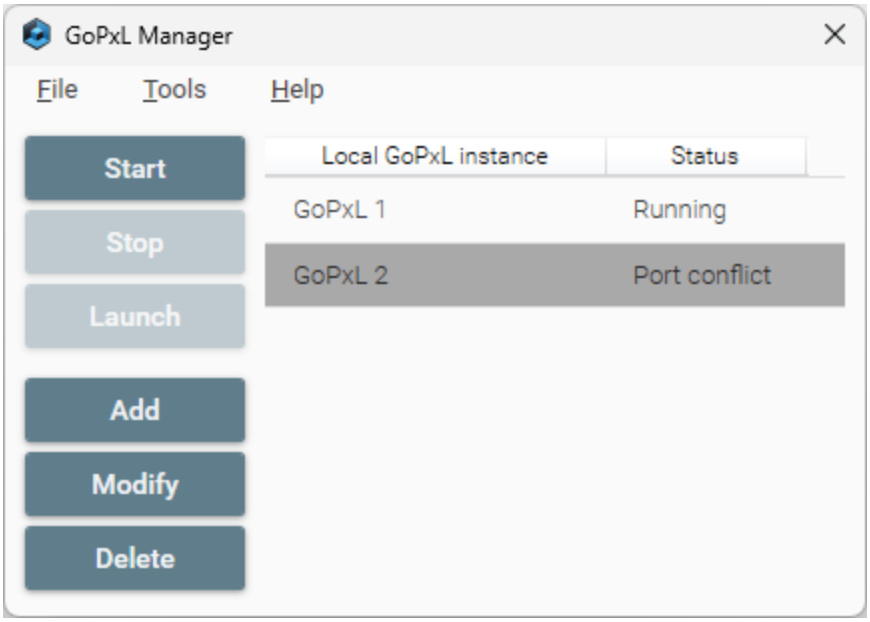
Additionally, your firewall may ask if you want to allow the GoPxL Service Application to communicate over your network. To avoid issues, LMI recommends allowing communication over both Public and Private networks.

Note that GoPxL Manager starts minimized in the tray area.

At this point, you can do one of the following:

- Add one or more sensors to the PC instance to accelerate the sensors. For information on adding a sensor to a sensor group, see *Accelerating a Sensor* on page 1200.
- Open scan data someone has previously recorded (in a .gprec, .gpbak, or .gpsup file) and examine the data or use it to configure measurement tools without taking your sensor system offline. For more information, see *Loading Scan Data* on page 1204. (For information on the file formats used with GoPxL, see *File Formats* on page 76.)

All ports used by PC instances must be unique. In some situations, it is possible to accidentally set the ports of two or more instances to the same port.



To correct this, select one of the instances, click **Modify**, and change the duplicated port to a different one. You can also delete any instances that use duplicate ports.

> You can select multiple instances in GoPxL Manager and perform most of the actions on the selected instances.
>
> Press Ctrl + Left click to toggle whether an additional instance is selected. Press Shift + Left click to select a range of instances (from the currently selected instance to the one you Shift + Left click).

Running GoPxL Manager in Administrator Mode

To perform certain operations, you need to start GoPxL Manager in Administrator mode.

- Change certain applications settings (for more information, see *Application Settings* on page 1194).
- Install or uninstall a PC instance as a service, or delete it.
- Install licensing for tools like GoPxL Anomaly Detector.

> Before switching to Administrator mode, you should stop all running PC instances of GoPxL.

To run GoPxL Manager in Administrator mode, click **Run As Admin** either in the GoPxL Manager Settings dialog (see *Application Settings* on page 1194), or in the Modify GoPxL Instance Configuration dialog (see *Instance Settings* on page 1196).

> After you have finished making any of the changes listed above, you should fully exit GoPxL Manager to avoid permission issues with your instances. For more information, see *Closing GoPxL Manager and Stopping All PC Instances* on the next page.

Closing GoPxL Manager and Stopping All PC Instances

After you have finished making any of the changes listed in *Running GoPxL Manager in Administrator Mode* on the previous page, you should fully exit GoPxL Manager to leave Administrator mode to avoid permission issues with your instances as described below. You use the same steps if you just need to exit GoPxL Manager and stop all PC instances. Note that clicking the X button at the top right in GoPxL Manager only hides the utility's main window.

To close GoPxL Manager and all PC instances

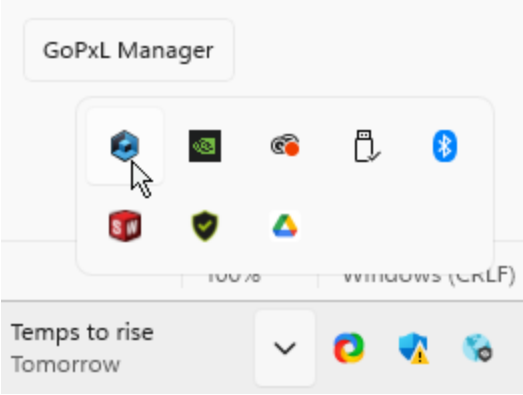
1. Look for the small GoPxL Manager icon ( ) in the taskbar's notification area.

   This is normally at the bottom of the screen, to the right.

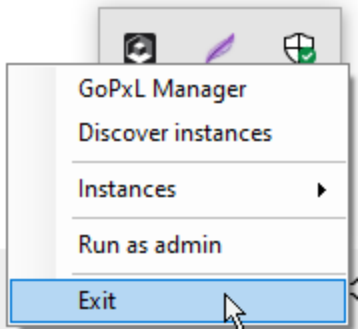
2. If you don't see the GoPxL Manager icon, click the "Show hidden icons" arrow.



   A menu containing the hidden icons appears.

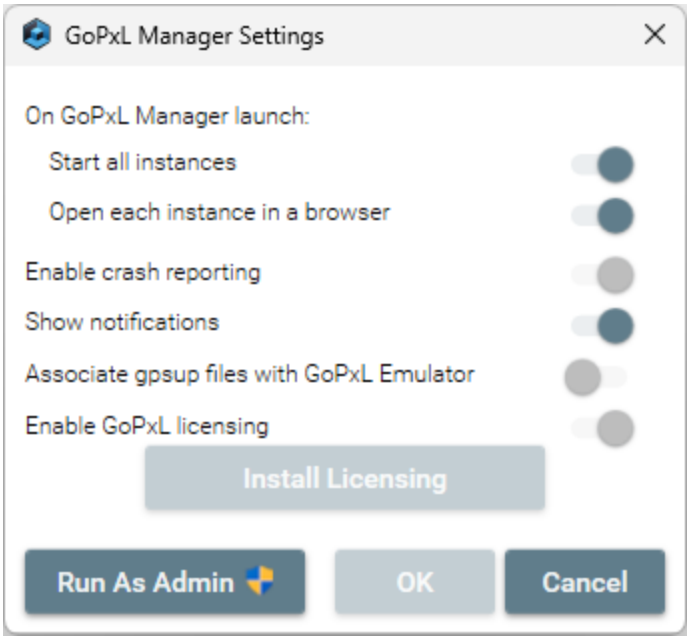


3. Right-click the GoPxL Manager icon.
4. In the menu that appears, click Exit.

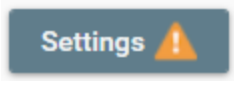


Application Settings

The following application settings are available. To open the settings, click **Tools** > **Settings** at the top of the application.



Parameters

Parameter	Description
On GoPxL Manager launch	**Start all instances** When enabled, all instances not installed as a service will automatically start when GoPxL Manager starts. **Open each instance in a browser** Each instance opens in the default browser. (Instances installed as a service will start when the computer is started.)
Enable crash reporting	If a PC instance of GoPxL crashes, a report will be generated. This can be useful if you need to troubleshoot problems with LMI support. LMI recommends leaving crash reporting enabled. If you choose to disable crash reporting, an icon on the **Settings** button will indicate that it is disabled.  To change this setting, you must be running GoPxL Manager as an administrator. For more information on using GoPxL Manager in Administrator mode, see *Running GoPxL Manager in Administrator Mode* on page 1192.
Show notifications	GoPxL Manager will show notifications in the tray area, provided that notifications

Parameter	Description
	are enabled in Windows.
Associate gpsup files with GoPxL Emulator	Double-clicking a support file (.gpsup) launches the support file in the emulator. To enable this, you must first launch GoPxL Manager in admin mode by clicking **Run as Admin**. If you try to launch a different version of GoPxL Emulator than the one in which you enabled this, the association will become disabled; you must then re-enable it for it to work with the version you have just launched.
Enable GoPxL licensing	In order to use tools such as GoPxL Anomaly Detector with live scan data, you must purchase a license and dongle. Then in GoPxL Manager, you must enable licensing and install the licensing software if necessary. For more information see *Installing Licensing Software* below. To change this setting, you must be running GoPxL Manager as an administrator. For more information on using GoPxL Manager in administrator mode, see *Running GoPxL Manager in Administrator Mode* on page 1192.

Installing Licensing Software

If you have purchased a license and dongle for using tools such as GoPxL Anomaly Detector, you must install the licensing software, Wibu-Systems CodeMeter.

To install the licensing software

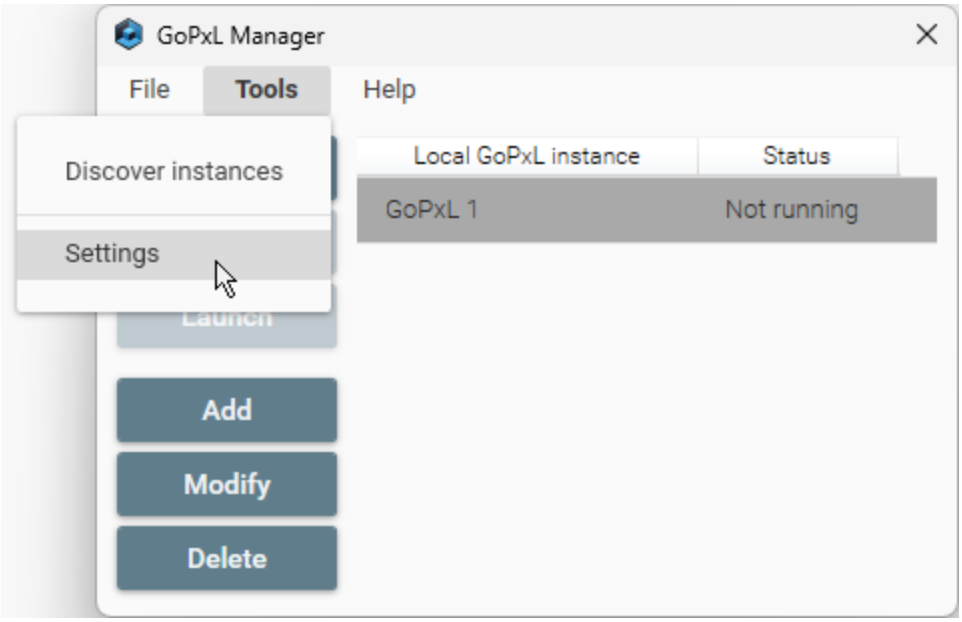
1. If it is not already running, start `GoPxL.exe` and open GoPxL Manager.

   For more information, see *Launching GoPxL on a Windows PC* on page 1190.

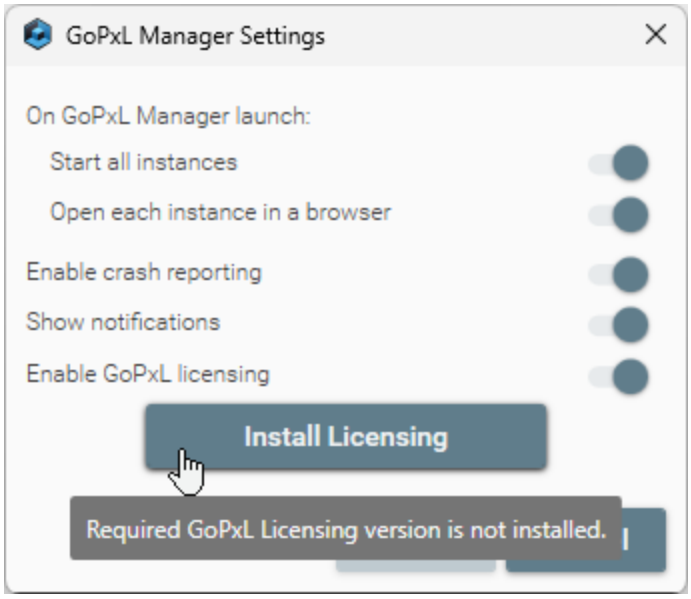
2. Restart GoPxL Manager in Administrator mode.

   For more information, see *Running GoPxL Manager in Administrator Mode* on page 1192.

3. Click **Open** > **Settings**.



4. In GoPxL Manager Settings, if licensing isn't already enabled enable it.
5. Click **Install Licensing**.

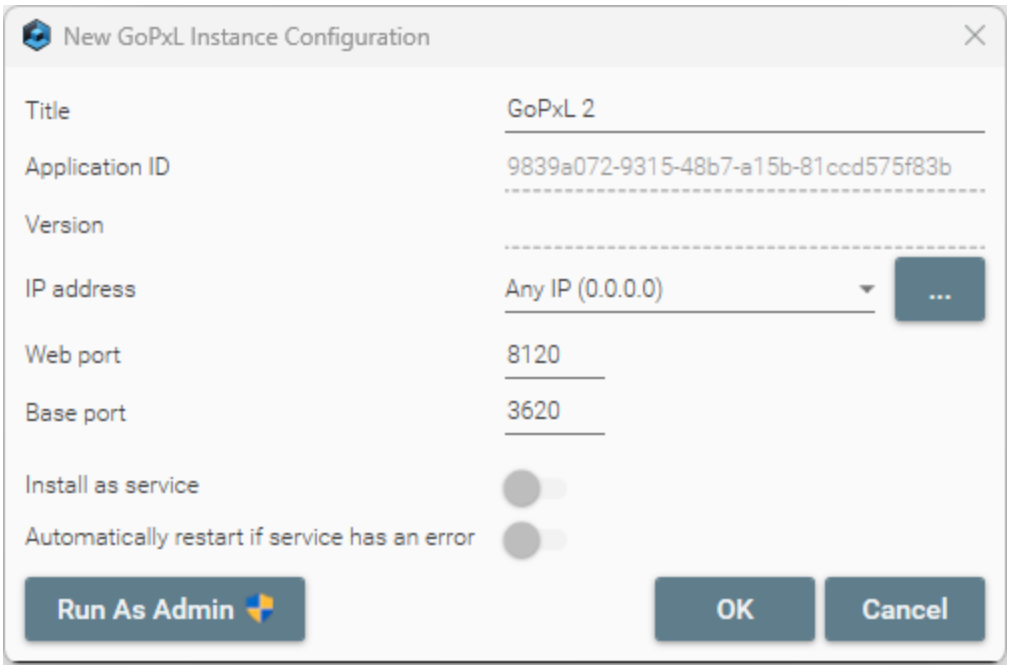


6. Follow the instructions in the CodeMeter installer.

   In the Custom Setup step, leave the selection of features as the default.

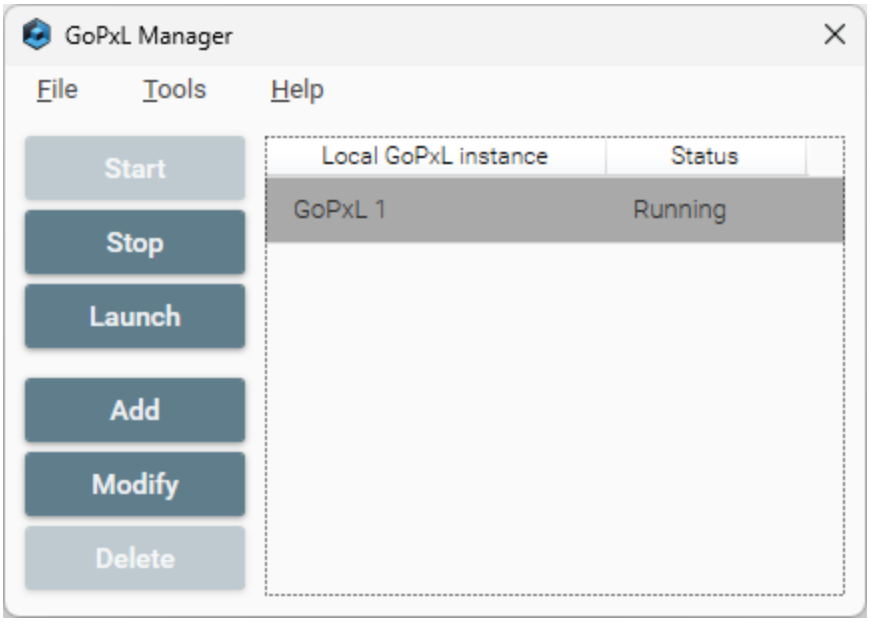
Instance Settings

When adding or modifying an instance, you can change various settings (see *Instance Configuration* on page 1198).



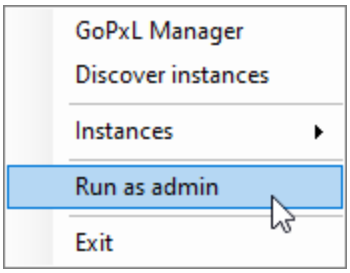
To add an instance, click **Add** and configure it if needed, and then click OK.

To modify an instance, select it in the list of instances and click **Modify**. To make changes to an instance, it must be stopped; select the instance and click **Stop** if needed.

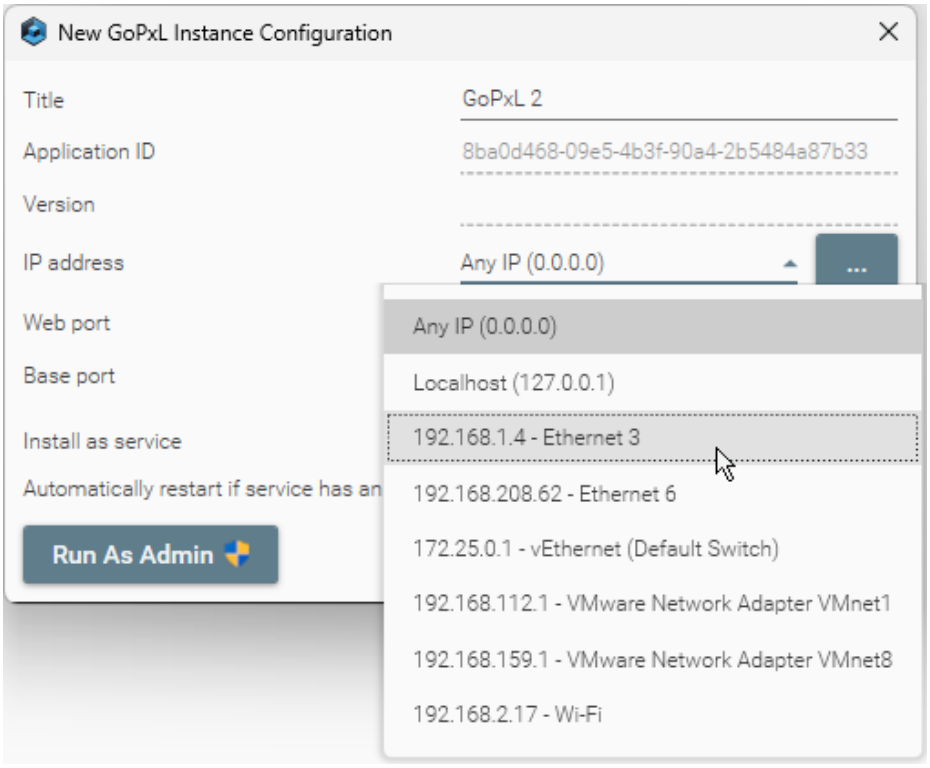


GoPxL Manager with one instance running.

To change the settings of a service instance or delete it, you must be running GoPxL Manager as an administrator. You can do this by clicking **Run As Admin** in the Instance Configuration dialog (see above), which restarts GoPxL Manager with administrator rights. You can also run right-click its icon in the Windows taskbar and click **Run as admin**.



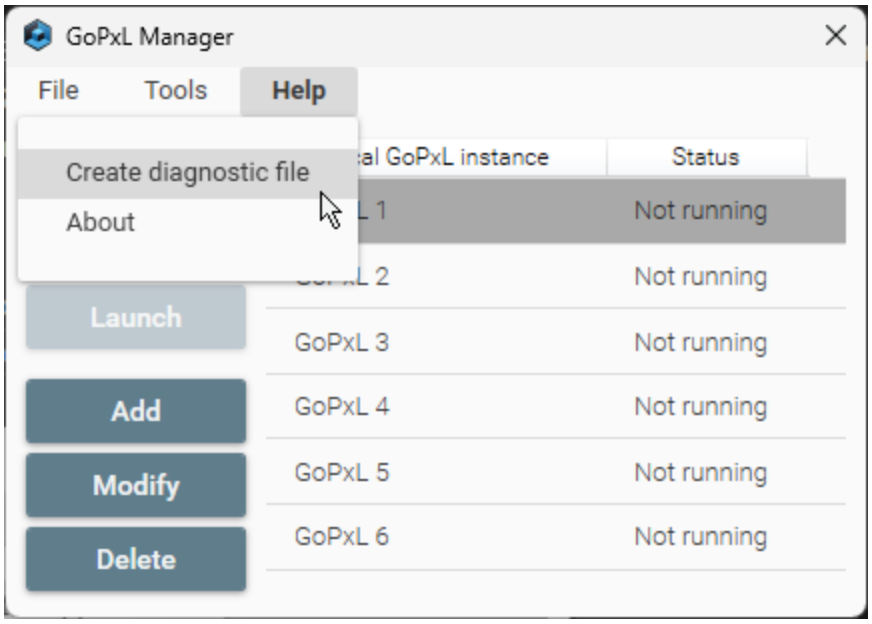
Instance Configuration

Parameter	Description
Title	The name of the instance in GoPxL Manager.
Application ID	The ID of the GoPxL instance. Not editable.
Version	The version of the GoPxL instance. Not editable.
IP address	The IP address of the instance (that is, the IP the instance is bound to). The drop-down lists the available interfaces on your PC.  Using the address you provide followed by the instance's Web port in a web browser connects to GoPxL. You can find the Web port of an instance in the Instance Configuration dialog or in the Discovery tool by clicking **Discover Instances**. Choose a specific Ethernet or Wi-Fi interface to bind the instance to that interface only. The instance will be visible to other PCs on your network at that IP address, provided that this is not blocked by a firewall.

Parameter	Description
	Choose "Any IP (0.0.0.0)" to bind the instance to *any* available interfaces. You can then connect to the sensor using any of the interfaces available on the PC (Ethernet, Wi-Fi, localhost, and so on). Except for the localhost address, the instance will be visible to other PCs on your network at those IP addresses, provided that this is not blocked by a firewall. Note that when you launch the GoPxL interface of an instance set to "Any IP" from GoPxL Manager, it opens at http://127.0.0.1/ followed by the Web port. Also, in the Discovery tool, the first IP found by the tool is listed in the table. If you are using PROFINET, you must set this to a specific interface: PROFINET does not support "Any IP" or Localhost. Choose Localhost to bind the instance to http://127.0.0.1/. The instance will only be visible to the PC it's running on.
Web port Base port	The main ports the instance uses. These must be unique for each instance. **Web port** The port you use when connecting to the instance, for example, to `http://127.0.0.1:8120/`. **Control port** The port you use with the GoPxL SDK and REST API (see *GoPxL SDK and REST API* on page 1234).
Install as a service	A PC instance installed (and running) as a service can run a sensor without the need for a user to be logged into Windows.
Automatically restart if service has an error	If an instance of GoPxL is configured to be a service and it has an error, it will restart automatically.

Creating a Diagnostic File

To help troubleshoot issues related to PC instances of GoPxL, you can create a diagnostic file containing logs and instance files. You usually only need to do this when asked to do so by LMI support.



After choosing Create diagnostic file, Windows will prompt you for a location.

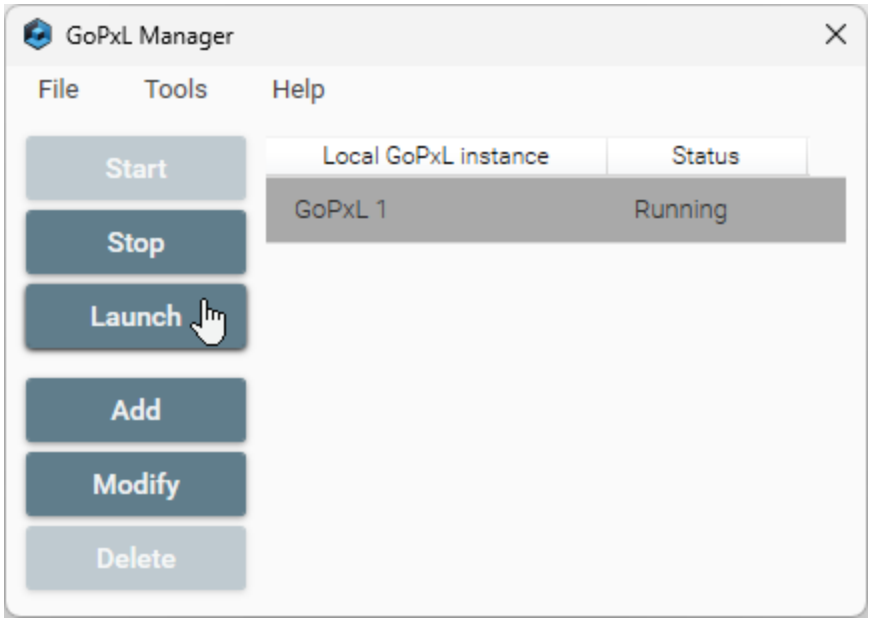
Accelerating a Sensor

You accelerate a sensor (or group of sensors) by connecting it to a PC instance of GoPxL.

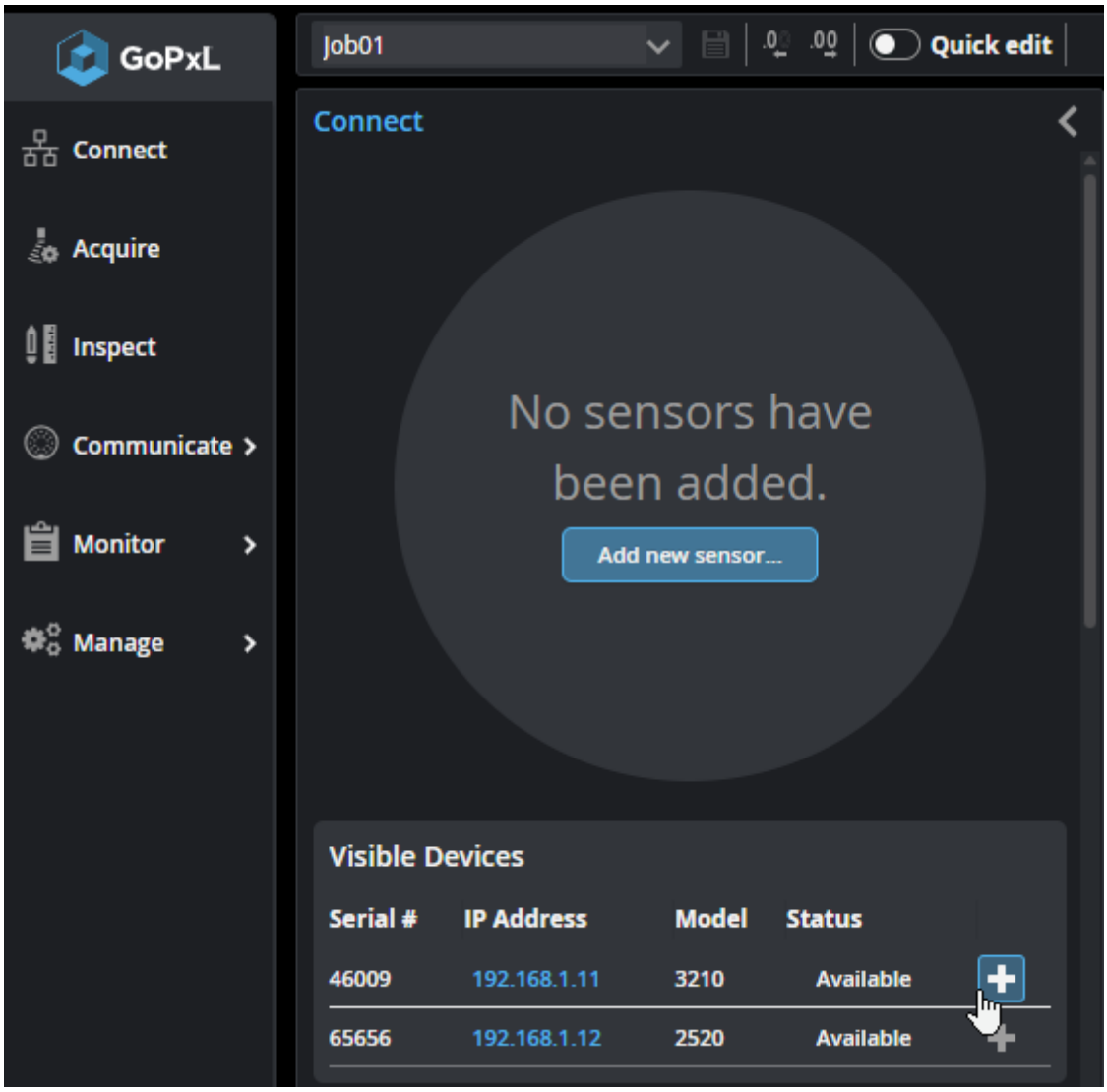
To accelerate a sensor

1. In a web browser, go to the IP address of the PC instance.

   You can also launch the PC instance from GoPxL Manager.

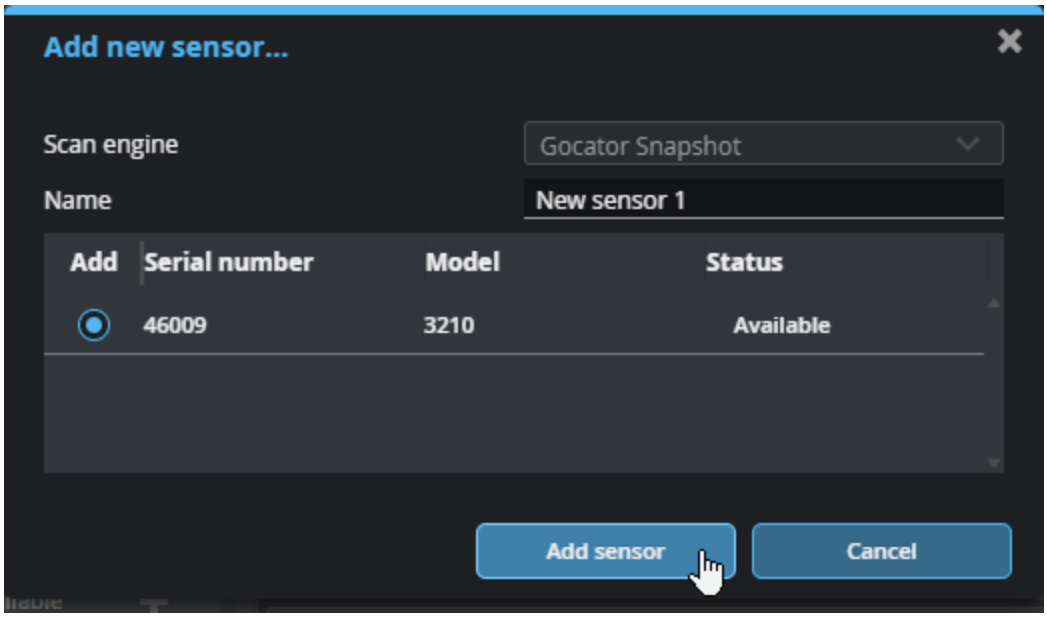


2. On the **Connect** page, in the **Visible Devices** area, click the plus symbol next to the sensor you want to add.

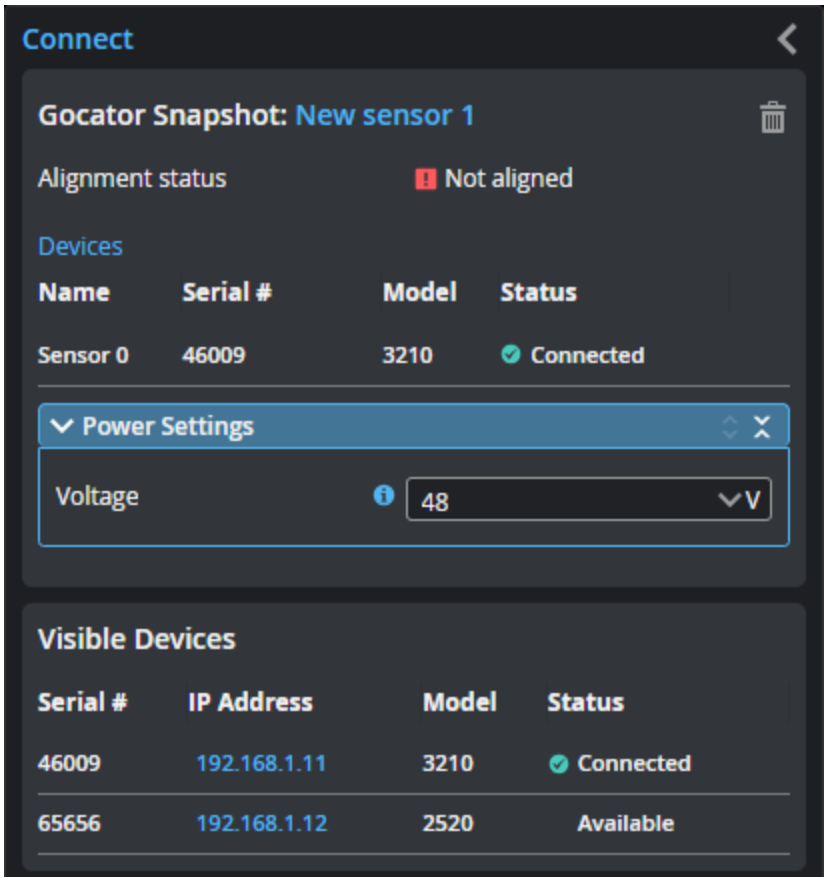


3. In **Add new sensor...**, if the sensor is not already selected, click next to it in the **Add** column, and click **Add Sensor**.

   You can optionally set the name of the sensor.



The physical sensor is now connected to the PC instance of GoPxL and is automatically accelerated.



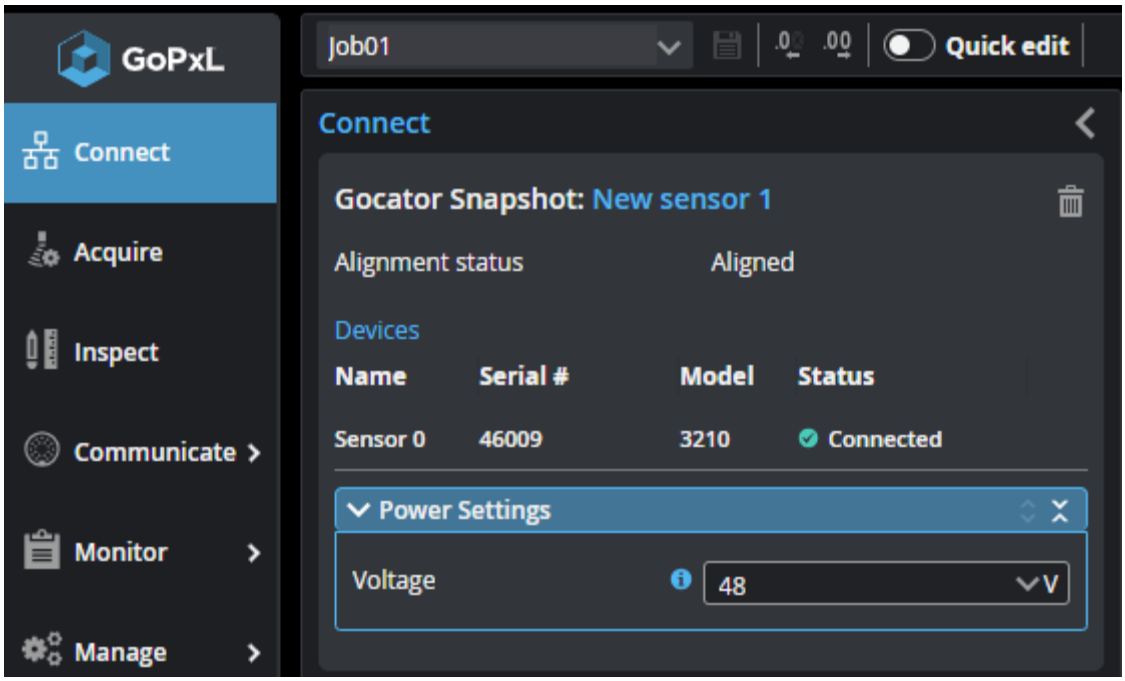
You access the sensor via the IP address of the PC instance, through which you control and configure it.

Stopping Acceleration

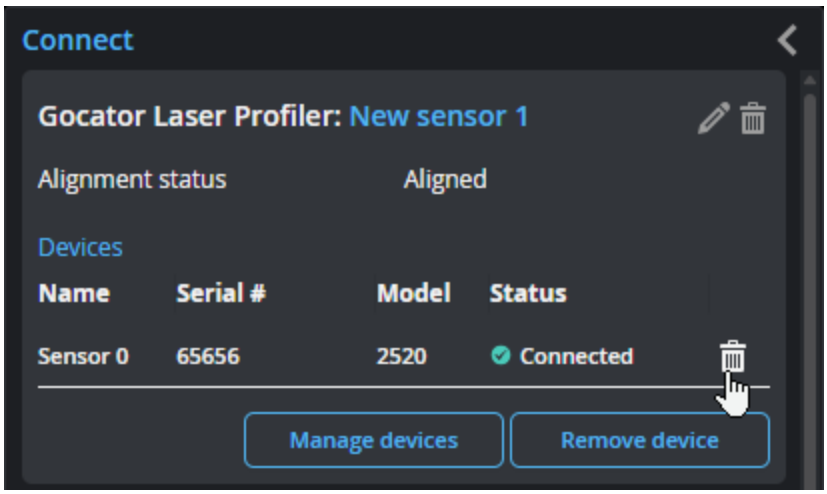
To stop accelerating a sensor, you remove it from the PC instance of GoPxL it's connected to. You also need to remove a sensor from a PC instance to upgrade the sensor's firmware.

To remove a sensor from a PC instance

1. In the GoPxL interface of the PC instance, go to the **Connect** page.

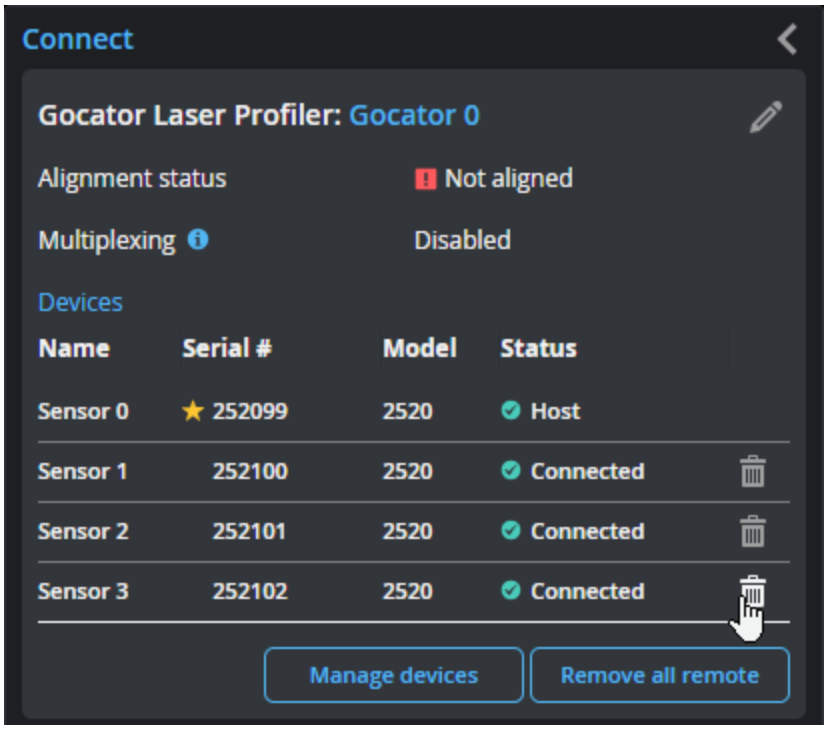


2. With a single accelerated line profile sensor, click the trashcan icon or **Remove device**.

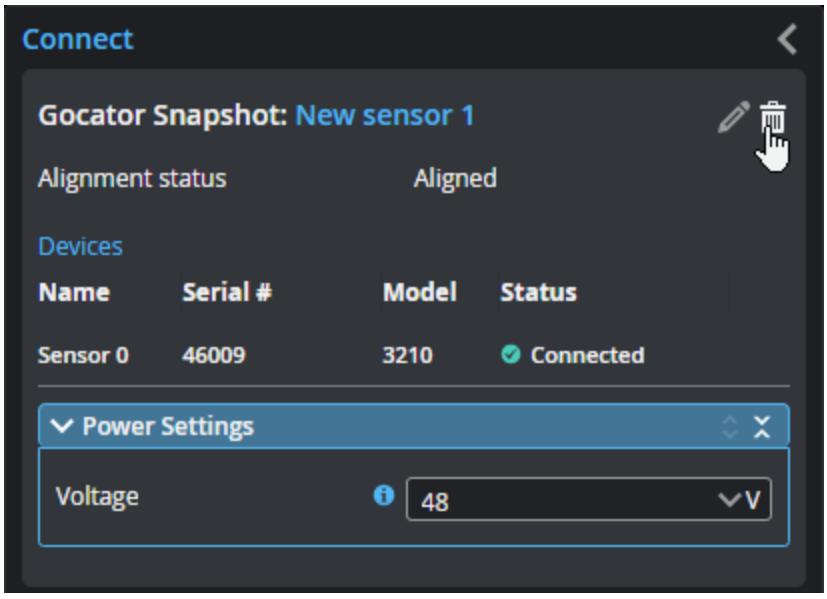


   Remove sensor from the group: Choose this if you have started configuring settings on the **Acquire** page and just need to add a different sensor to the sensor group.

3. With a multi-sensor line profile system, click a trashcan icon to remove a single sensor, or **Remove all remote** to remove all sensors.



4. With G3 sensors, click the trashcan icon.



The sensor is removed from the PC instance.

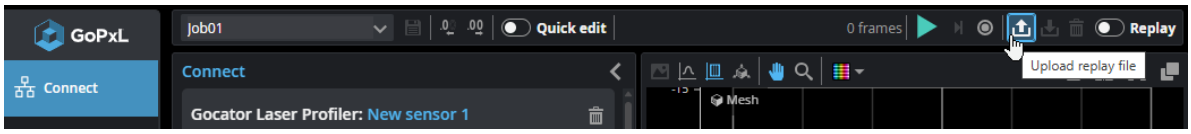
Loading Scan Data

To examine previously recorded scan data offline, you load a .gprec file into the PC instance of GoPxL.

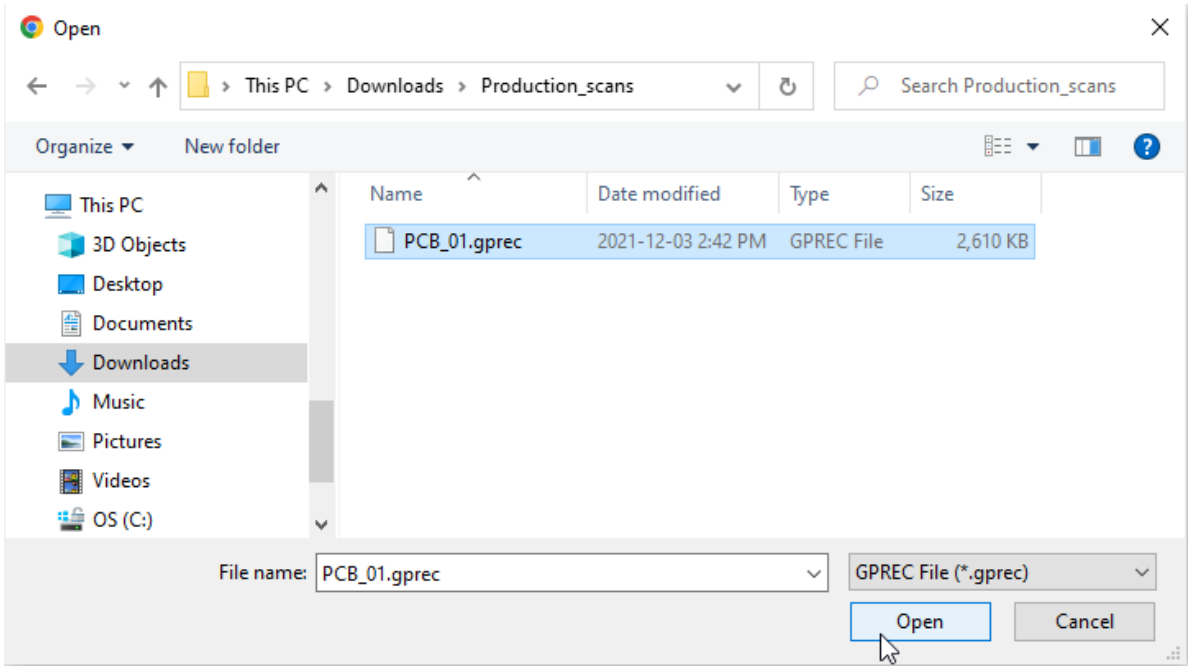
You can also load a backup file or a support file to examine previously recorded scan data.

To load recorded data (.gprec)

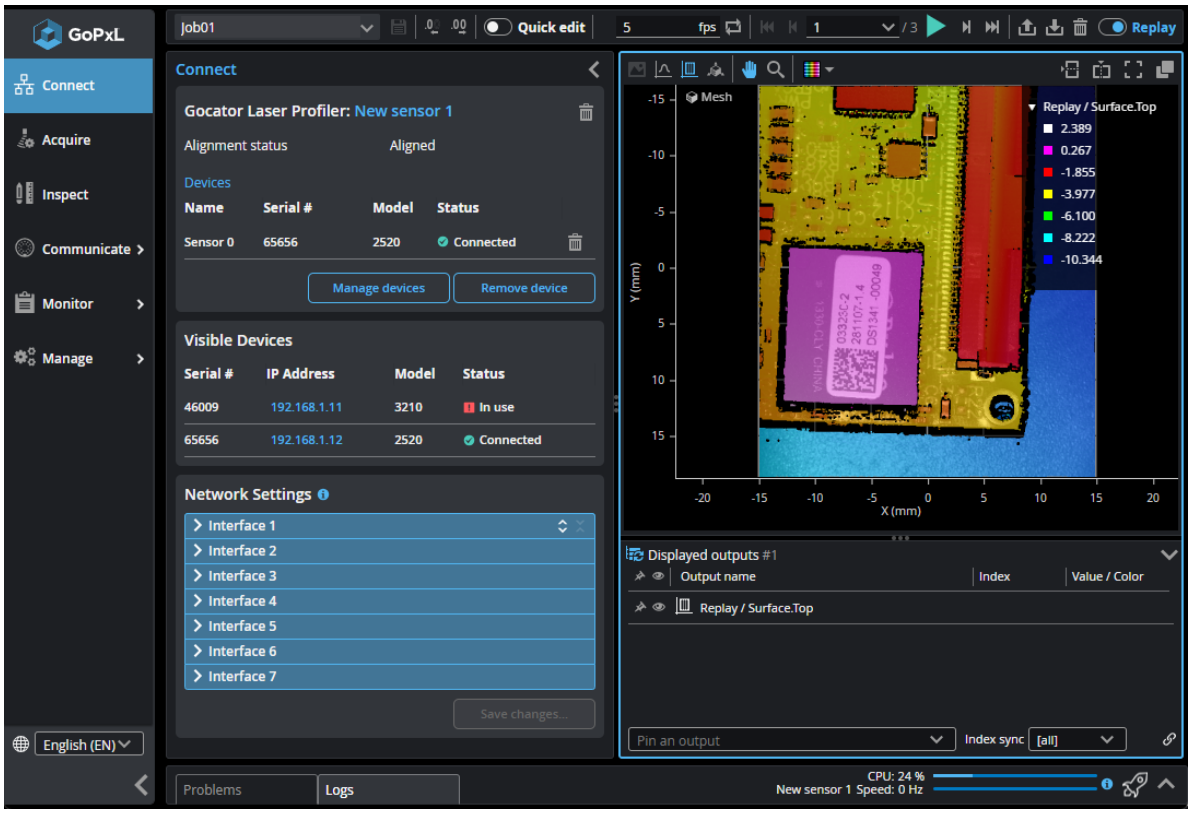
1. In the GoPxL interface, above the data viewer, click the Upload replay file button.



2. In the **Open** dialog, locate and select the recording (.gprec) you want to load, and click **Open**.



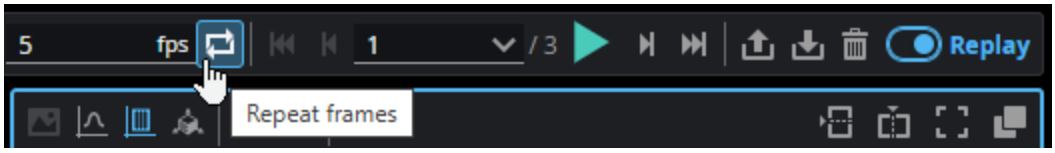
The recording loads in GoPxL, and GoPxL automatically switches to Replay mode.



You can now examine the recorded data and configure tools run on it.

3. Use the replay controls above the data viewer to move through the frames, go to a specific frame, toggle and configure the speed of repeat frames mode (continuous loop), and so on.

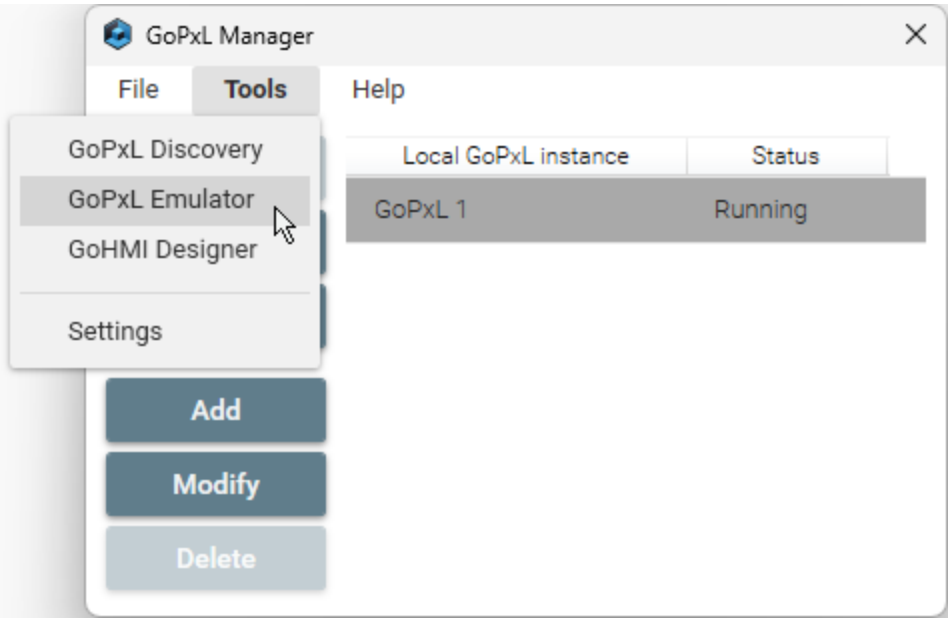
   Hover over the replay controls for tooltips describing their function.



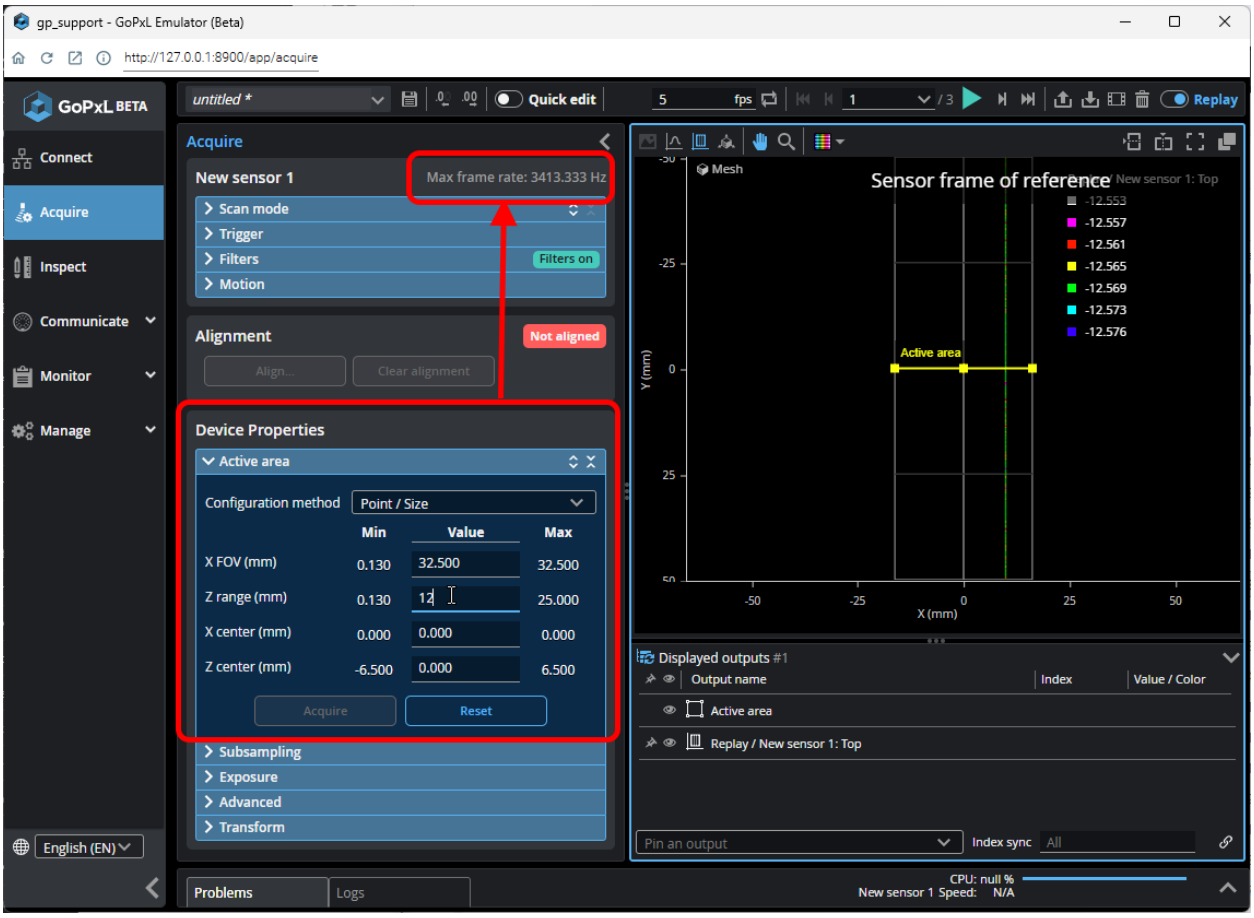
Using repeat frames mode with a low frame speed is useful for adjusting measurement tool settings and seeing the impact on scan data, without having to click Start or step through the frames of scan data manually. Repeat frames mode is also useful for demos.

GoPxL Emulator

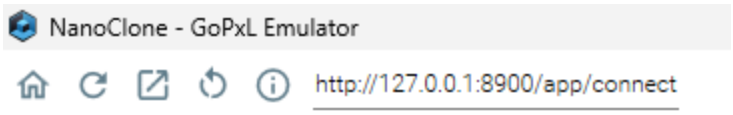
You can launch one or more instances of a GoPxL emulator in the GoPxL Manager, under the **Tools** menu. GoPxL supports launching multiple emulator instances.



You can open support files (.gpsup) for 3D sensor models only; single sensors or multi-sensor systems are supported. The emulator also supports files from sensors that are accelerated. After you select a support file, the emulator starts. You can then evaluate the maximum frame rate for active area changes, on the **Acquire** page.



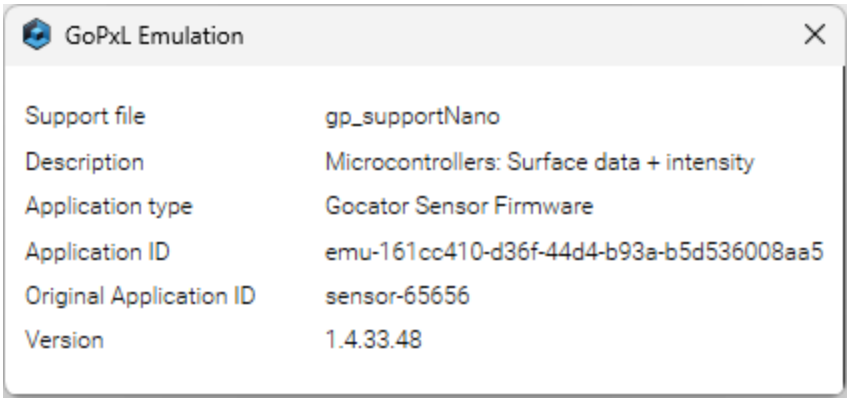
You can associate .gsup files with the emulator by enabling **Associate gpsup files with GoPxL Emulator** in GoPxL Manager; for more information, see *Running GoPxL on a Windows PC* on page 1188. Lastly, you can emulate support files saved from the emulator.



The emulator supports the following:

- Click the Home ( ) and Refresh ( ) buttons to go to the default page and refresh the current page, respectively.
- Click the Open in external browser button ( ) to open the emulated GoPxL instance in your default web browser.
- Click the Restart button ( ) to restart the emulation; this can be useful if you've made changes to the emulated sensor's job or have cleared the replay data.
- Click the Info button ( ) for information on the support file.

Note that **Version** refers to the version of GoPxL from which the support file was saved.



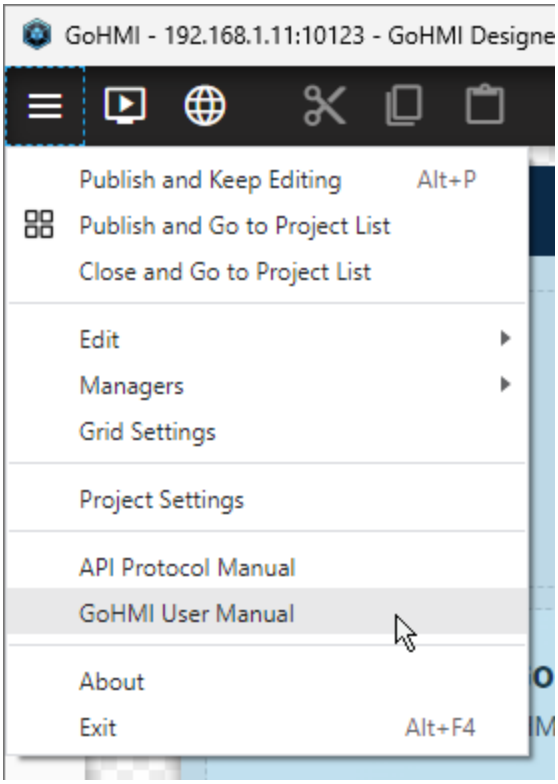
Note the following limitations:

- Support files from GoPxL 1.3 and earlier versions cannot be loaded.
- Support files from GoMax and Gocator 2D Smart Camera are not yet fully supported.

GoHMI and GoHMI Designer

GoHMI lets you create flexible human-machine interfaces (HMIs) and provide a custom GUI containing system and inspection metrics. Your HMI can also be touch-enabled. You create and edit HMIs in GoHMI Designer. You can design different layouts for different devices so that, for example, your HMI will adapt to the screen it is being displayed on (a PC monitor, a laptop, a tablet, or a phone), as well as its orientation on the screen.

For complete information on creating an HMI app, open the menu in the upper left of the designer, and choose one of the manuals.



If you don't see changes you have made in the GoHMI app, or an HMI you've just uploaded, you may need to clear your browser's cache to make it available: the GoHMI is cached for 60 seconds.

Before using the designer, you must enable GoHMI in GoPxL; for more information, see *Enabling and Configuring GoHMI in GoPxL* on page 1216.

GoHMI and GoHMI Designer only support a single HMI app, called "GoHMI." Also, GoHMI can only be enabled on one PC instance at a time. If you try to start GoHMI another instance, GoPxL displays an error.

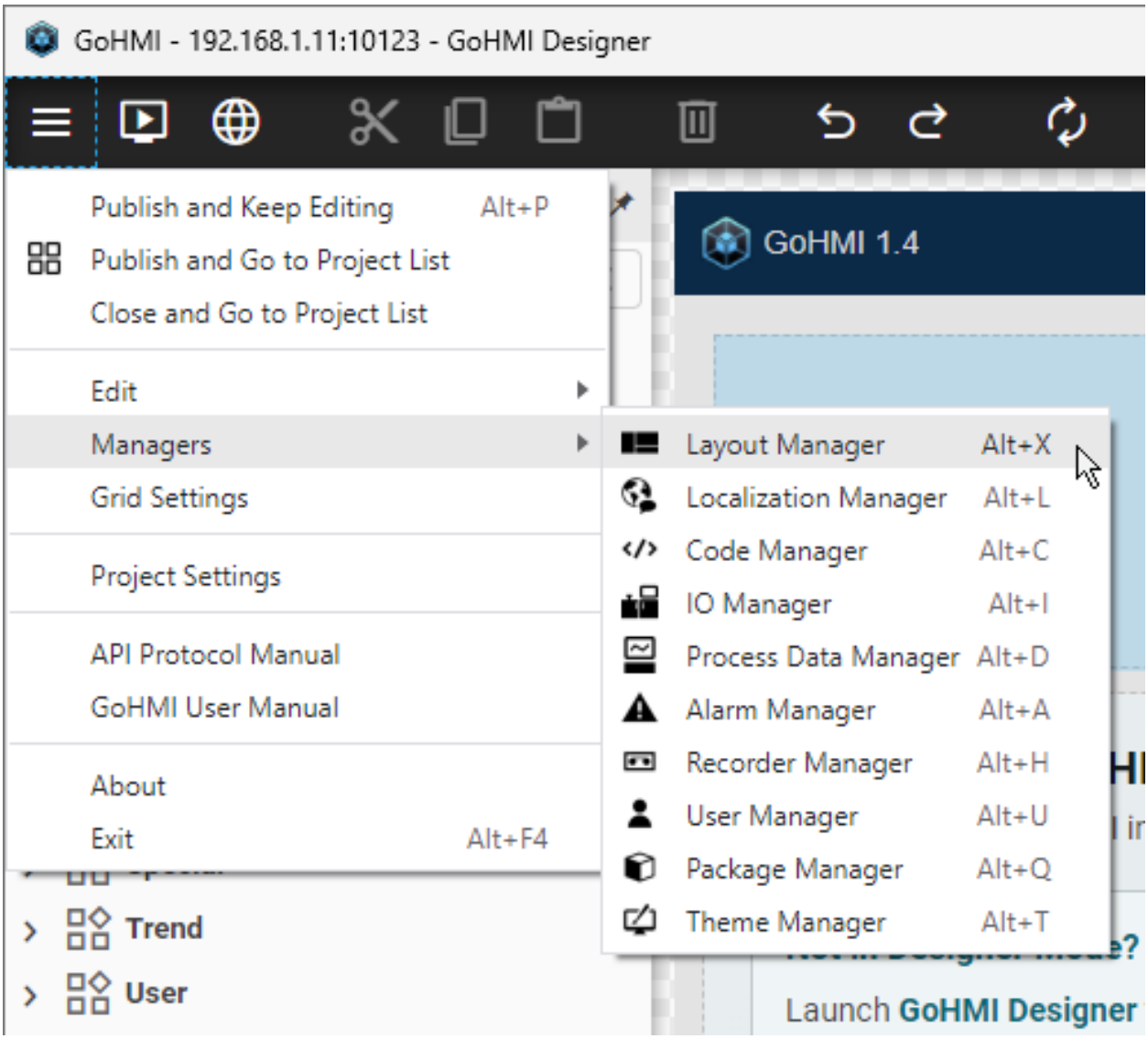
LMI provides a default HMI app with GoHMI that you use as a starting point or simply to explore the capabilities of GoHMI. When you launch GoHMI Designer, the default app loads in the designer.

GoHMI Designer is available in the GoPxL Utilities package (`14631-x.x.x.x_GoPxL_Utilities.exe`), available on LMI's Product Downloads page (https://lmi3d.com/product-downloads/). After you have downloaded the package, unzip it and make note of the folder's location; the shortcut to the designer is in the `GoPxL` folder.

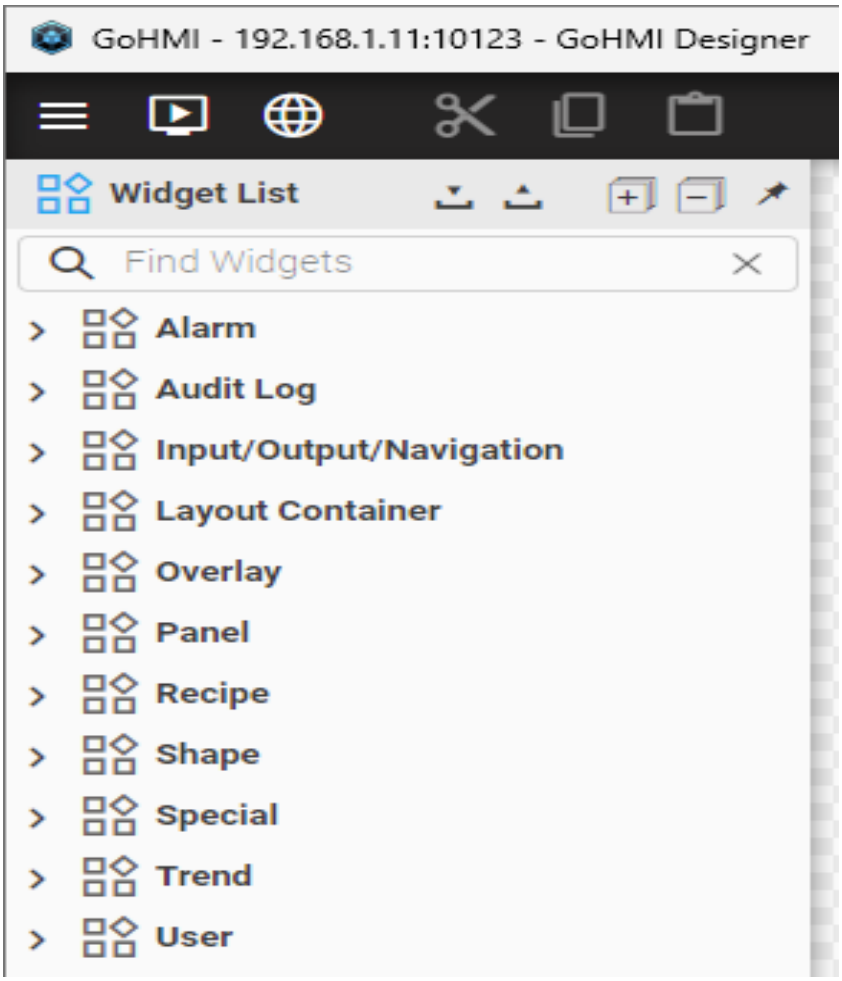
🗋	GoHMI Designer requires the Microsoft Visual C++ Redistributable package. This package is usually already installed on most PCs. If you are unable to launch HMI Designer, you may need to install this package, which can be downloaded at https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170.

⚠	If you decide to run your HMI application on a sensor, instead of on a PC instance or a GoMax device, you *should not* use the HMI data trending functionality. Data trending writes data repeatedly to storage, which can shorten the lifespan of the device's flash memory. Also, if the sensor's storage becomes full, the device may not be able to start. If you need data trending, you should run the HMI on a PC or on GoMax.

GoHMI Designer organizes the available design functionalities and settings into "managers" and settings pages. After you have opened an HMI app in GoHMI Designer, you can access the managers from the upper left menu in **Managers**.



In the Layout Manager, the side panel lets you access the list of widgets (user interface controls such as buttons) and layout containers you use to create your HMI. Click on "Widget List" to open the list. The widgets are grouped into categories. Below, the "Widgets" category is open.



GoHMI application layout development involves the concept of containers. Containers contain widgets or other containers (that is, containers can be nested). Containers can display their contents horizontally (for example, a row of buttons and fields) or vertically (for example, a stack of buttons). By nesting containers and adjusting their alignment, you have complete control over your HMI app's layout and design.

The following provides a high level view of what you need to do to start creating an HMI. The first steps below refer to general acquisition and measurement tool setup, which is described elsewhere in the manual. The remaining steps are described in further details below.

1. Install a sensor or group of sensors.

   For more information, see the installation and networking topics under *Installation, Setup & Hardware Overview* on page 23.

2. Configure the sensor acquisition settings.

   For information on configuring acquisition settings, see *Acquire - Alignment and Acquisition* on page 122.

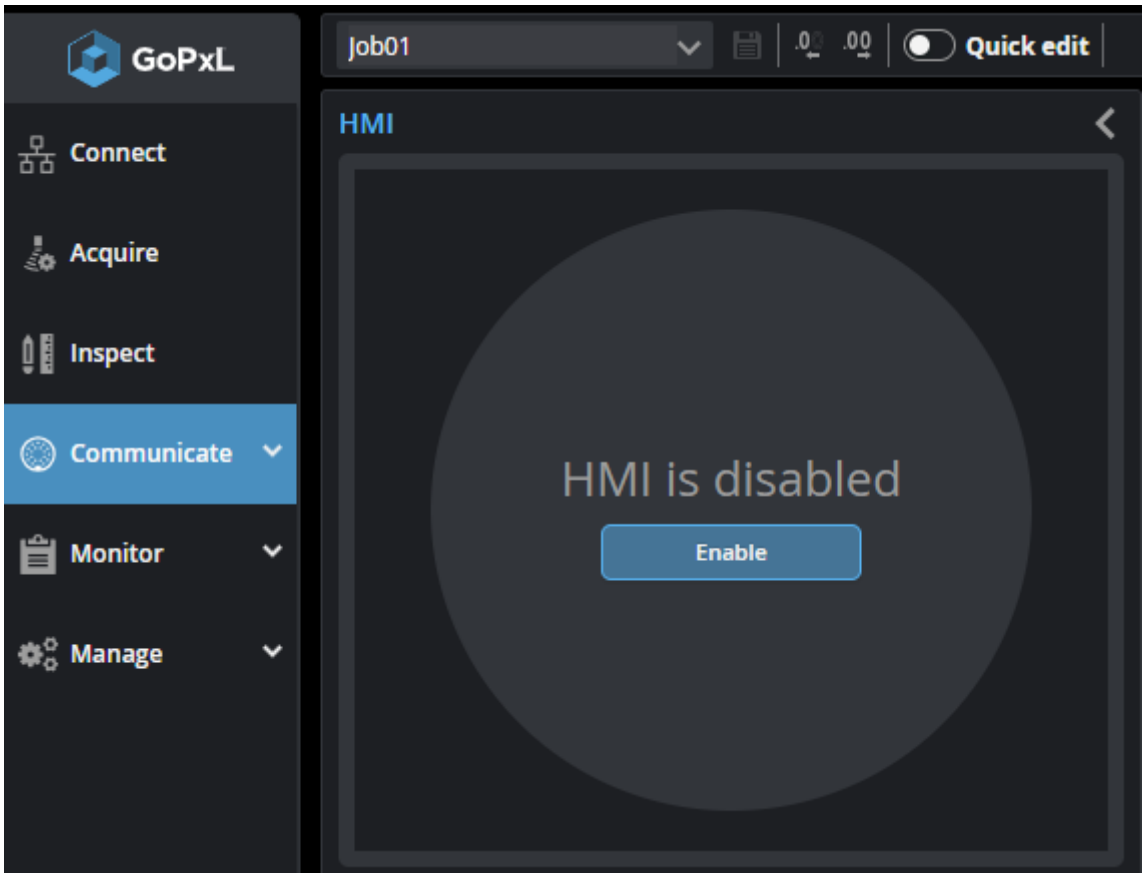
3. Add and configure measurement and processing tools.

   For more information, see *Inspect - Measurement and Processing Tools* on page 162 or the topics for specific tools.

4. (Optional) If your application requires the Gocator communication protocol or any of the industrial protocols, enable and configure these.

   For more information on the protocols available in GoPxL, see *Protocols (PLCs and other hardware)* on page 1257.

5. Enable the HMI control settings in the **Communicate** > **HMI** page using the toggle and configure GoHMI.



   For more information, see *Enabling and Configuring GoHMI in GoPxL* on page 1216.

6. Launch GoHMI Designer.

   For more information, see *Launching GoHMI Designer* on page 1220.

The following lists the browser requirements of GoHMI apps:

- Chrome 80 (released 04.02.2020)
- Firefox 74 (released 10.03.2020)
- Edge 80 (released 07.02.2020)
- Safari 14 (released 16.09.2019)
- Opera 67 (released 25.02.2020)

Overview of Managers and Settings Pages

For information on the editors (called "managers") you use in GoHMI Designer, see the following table.

Overview of managers and settings pages

Task	Manager	Description
Creating HMIs with Multiple Views	• Layout Manager	Creating views in your HMI using panels in GoHMI Designer.
Laying Out Your HMI	• Layout Manager	Styling your HMI in GoHMI Designer.
Using Widgets	• Layout Manager	Using the widgets provided in GoHMI Designer.
Styling the HMI	• Layout Manager	Laying out views and widgets in GoHMI Designer.
Using UI Actions	• Layout Manager	Defining UI-actions (such as mouse clicks on buttons in the HMI) to implement simple behaviors in GoHMI Designer.
Localizing the HMI	• Localization Manager	Localizing your HMI in GoHMI Designer.
Using the Integrated Code Editor	• Code Manager	Using scripts to add capabilities and features to your web HMI or to modify the appearance of widgets.
Processing Data Interface and Data Items	• IO Manager • Process Data Manager	Configuring the data interface and connecting your HMI to a sensor system using GoHMI Designer.
Simulating Process Values	• Process Data Manager	Simulating process values in GoHMI Designer to evaluate or to demonstrate your web HMI without a connected sensor system.
Using Alarms and Alarm Lists	• Alarm Manager	Defining alarms and messages, store them as historical alarms, and display the corresponding messages in the HMI.
Using Recorder and Trend Display	• Recorder Manager	Recording process variables and displaying them in a trend diagram in the HMI. The integrated data recorder supports both live trends and trends from historical data.
Using Recipes and Recipe Widgets	• Recipe Template Manager	Generating recipe templates and useful metadata such as "article numbers," "customer description," and so on, which serve only for documentation. In applications, recipe process values can be applied or captured.
Managing Users and Access Groups	• User Manager	Creating users and user groups and how to assign access rights.
Managing Packages	• Package Manager	Extending GoHMI Designer with additional packages for widgets, themes, and starter apps.

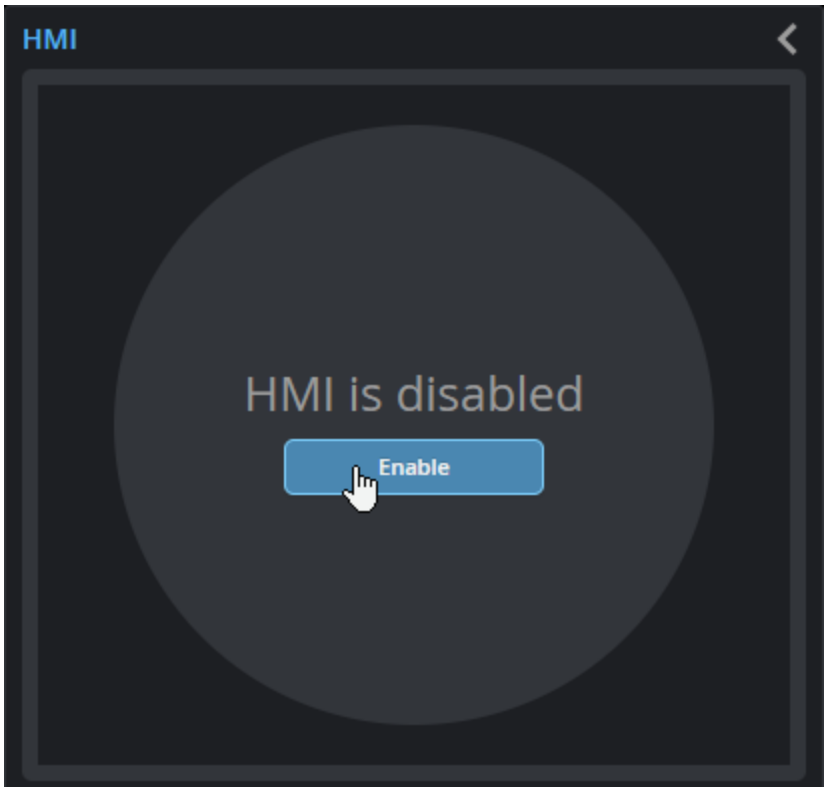
Adjusting App Settings	• HMI Project Settings	Changing application settings such as project settings in GoHMI Designer.
Adjusting Grid Settings	• Grid Settings	Adjusting the snap-to grid size in the designer.

Enabling and Configuring GoHMI in GoPxL

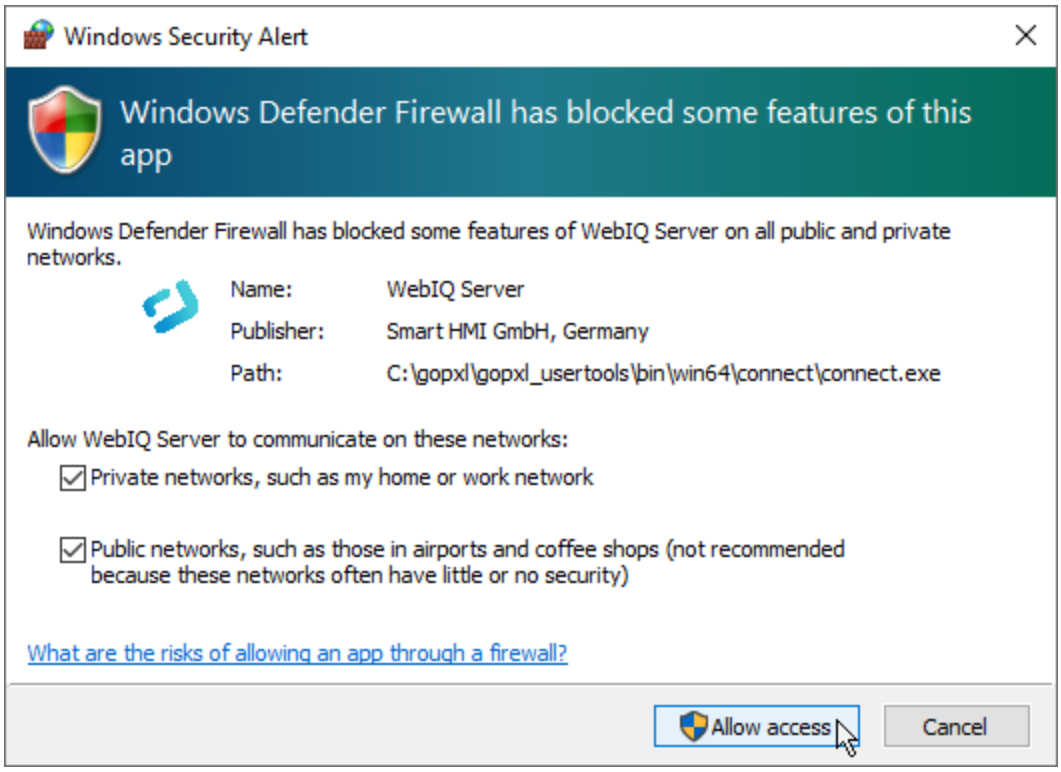
This section describes how to enable and configure GoHMI in the GoPxL interface. Before continuing, make sure you have created a sensor system, configured acquisition, and added and enabled the tools and measurements you want to display in the HMI.

To enable and configure GoHMI

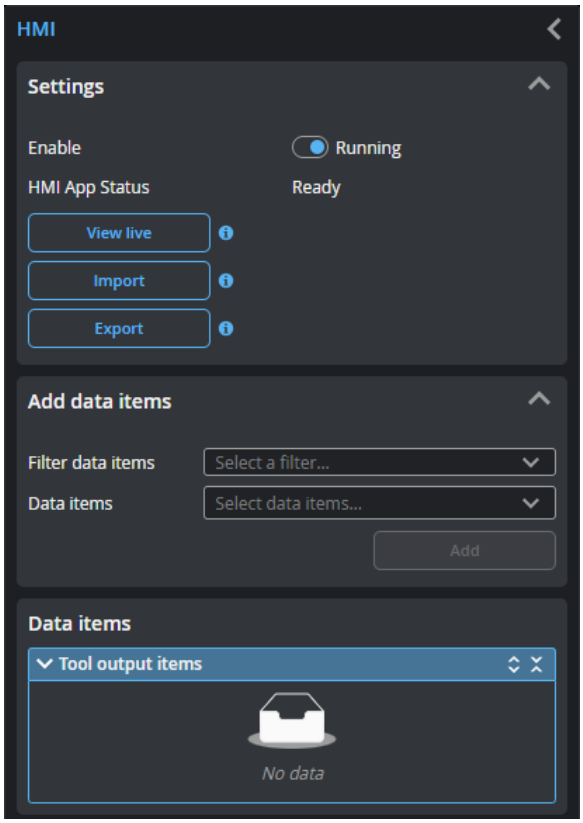
1. On the **Communicate** page, in the **HMI** panel, enable the HMI service.



   If you are running GoPxL on a PC, a Windows Security Alert referring to WebIQ (the service on which GoHMI Designer runs) may appear. Enable both networks.



The HMI service launches, and the HMI panel displays the following:



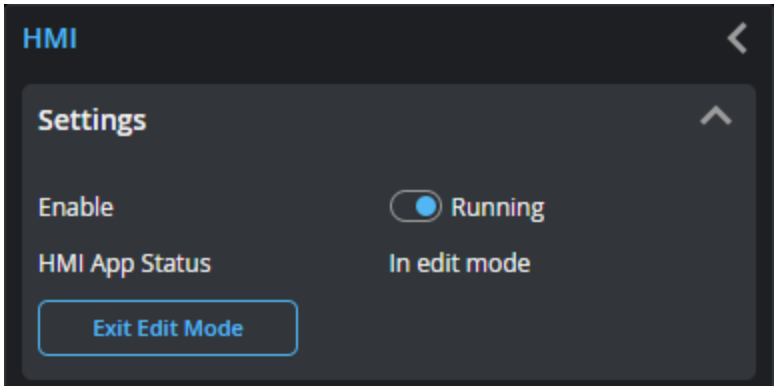
In addition to letting you disable the service and see the HMI app's status, the **Settings** panel lets you perform various operations.

When the application is running, you can open the HMI app in your browser by clicking the **View live** button. You can make note of the URL of the HMI in this tab for future use.

Import lets you import an HMI application.

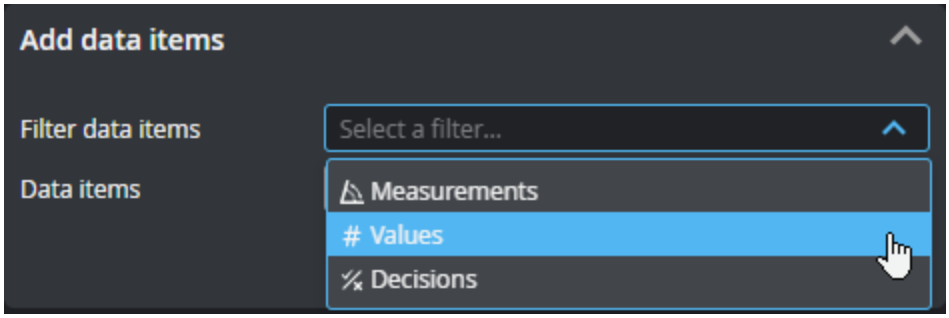
Export lets you export the HMI application currently available on the device (or in the PC instance), to back it up or transfer it to another sensor or PC.

2. If you see "In edit mode" as the HMI status instead of "Ready", you must remove it from the GoHMI Designer.

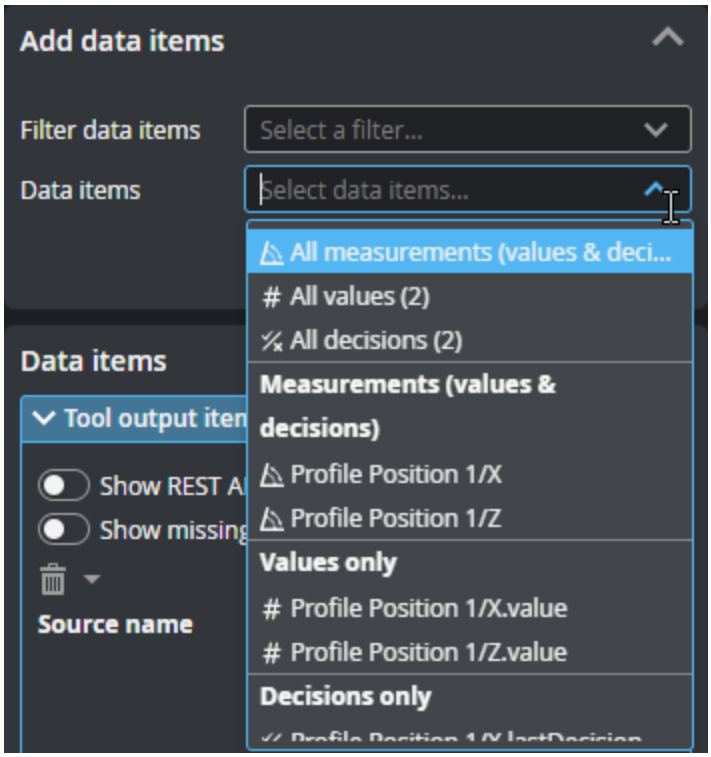


The status **In edit mode** means that the HMI app is currently loaded into the GoHMI Designer workspace. If you know you have finished making changes in GoHMI Designer, and have closed it without publishing it (to the sensor or PC instance of GoPxL), you can "force" exiting edit mode by clicking **Exit Edit Mode**. Note that doing this removes any unpublished changes in GoHMI Designer. For information on properly closing the designer and publishing changes, see *Publishing Changes or Exiting* on page 1225.

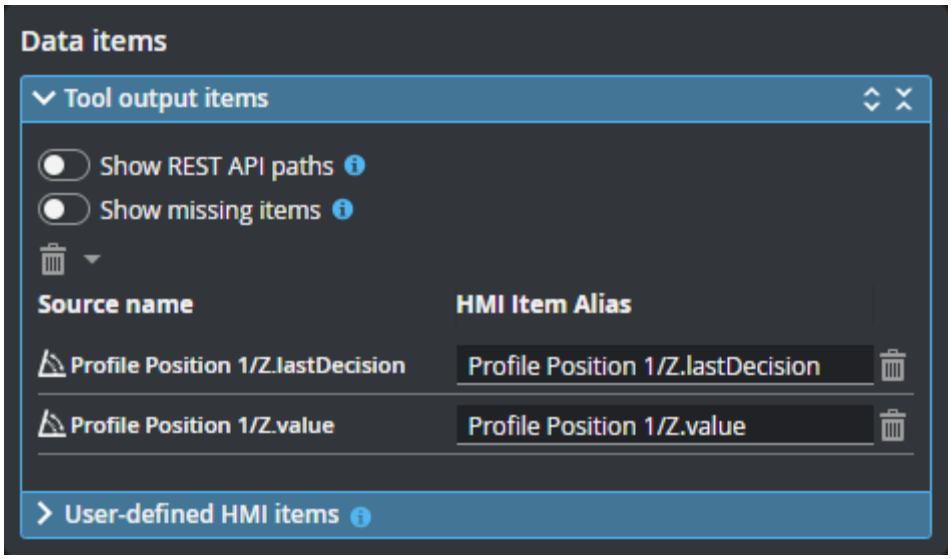
3. (Optional) You can filter the outputs to limit them to only values or decisions.



4. In the **Add Data Items** section, expand the **Data outputs** drop-down and click an output you want to send to the HMI.



After a few seconds, the output appears in the list of outputs that GoPxL will send to the HMI.



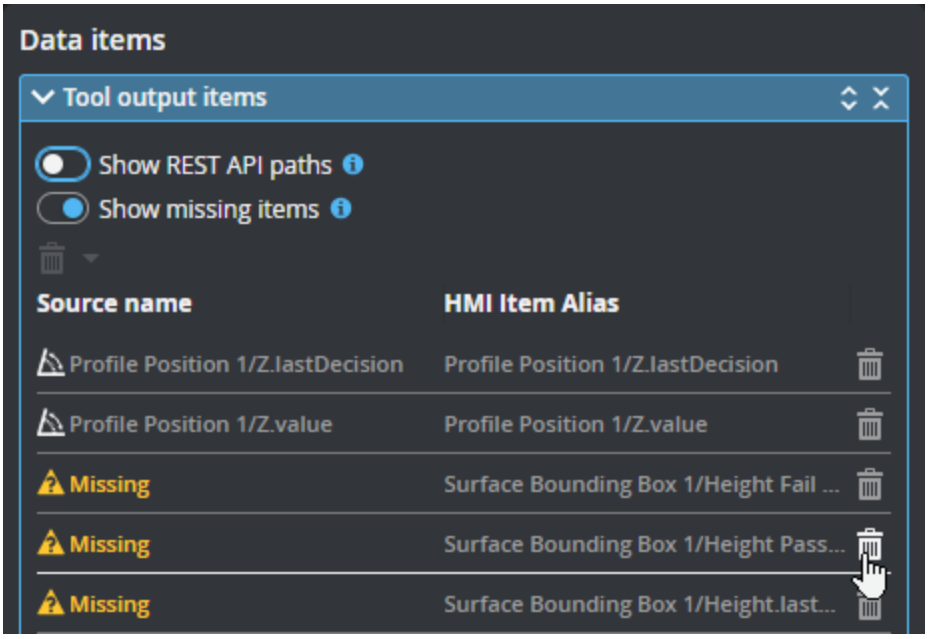
If you don't see the output you want to add in **Data items**, check that you have enabled it in the measurement tool, on the **Inspect** page.

5. (Optional) Toggle **Show REST API paths** to switch between showing the HMI item aliases and REST API of the outputs.

   HMI item aliases are how data outputs appear in GoHMI Designer.

   REST API paths are how the data output is represented in the REST API.

If GoPxL outputs referenced in an HMI are accidentally disabled or renamed in GoPxL, or the tool containing the outputs is removed, a **Show missing items** toggle will be displayed. Toggle this on to see which outputs are missing, and resolve the issue as required.

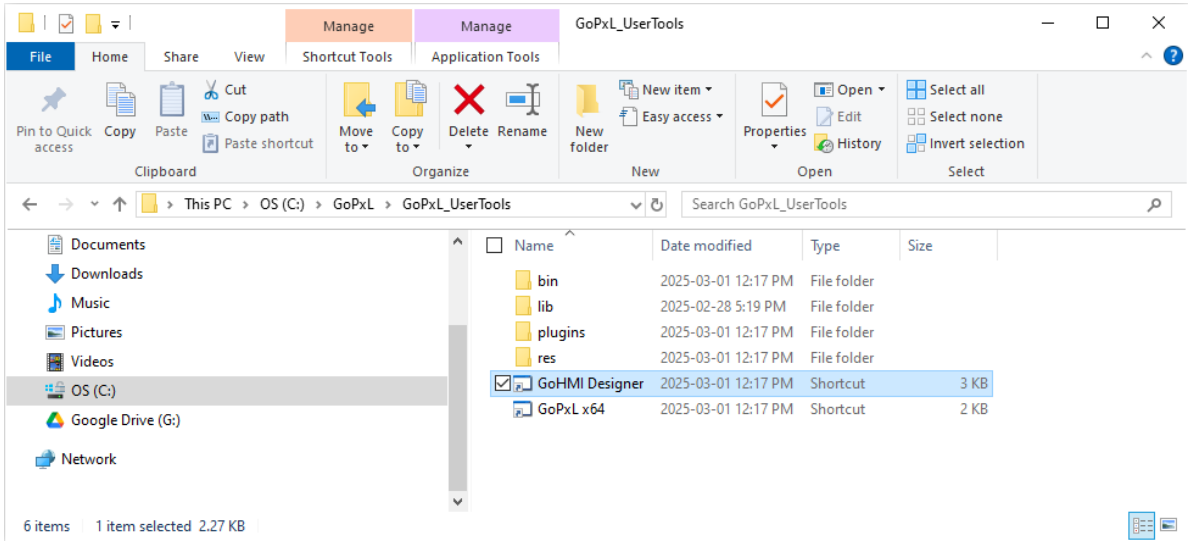


Launching GoHMI Designer

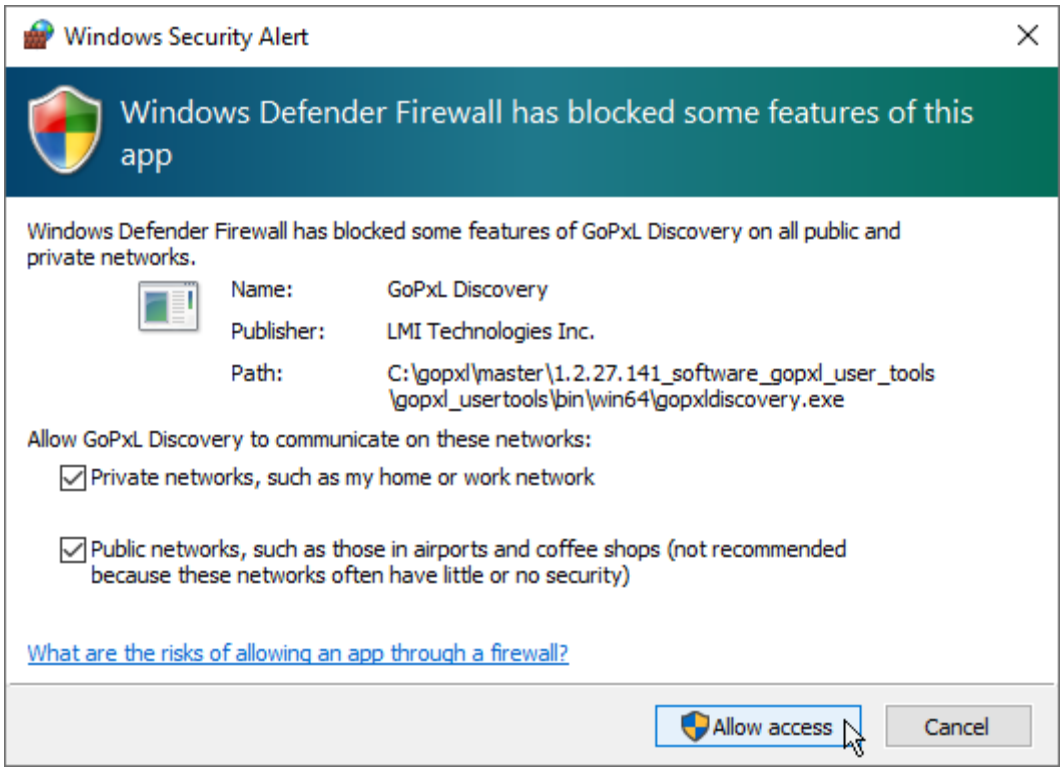
After you have enabled the HMI service in GoPxL and configured it by adding the outputs you want to send to the HMI, you can launch GoHMI Designer to create or edit an HMI.

To launch GoHMI Designer

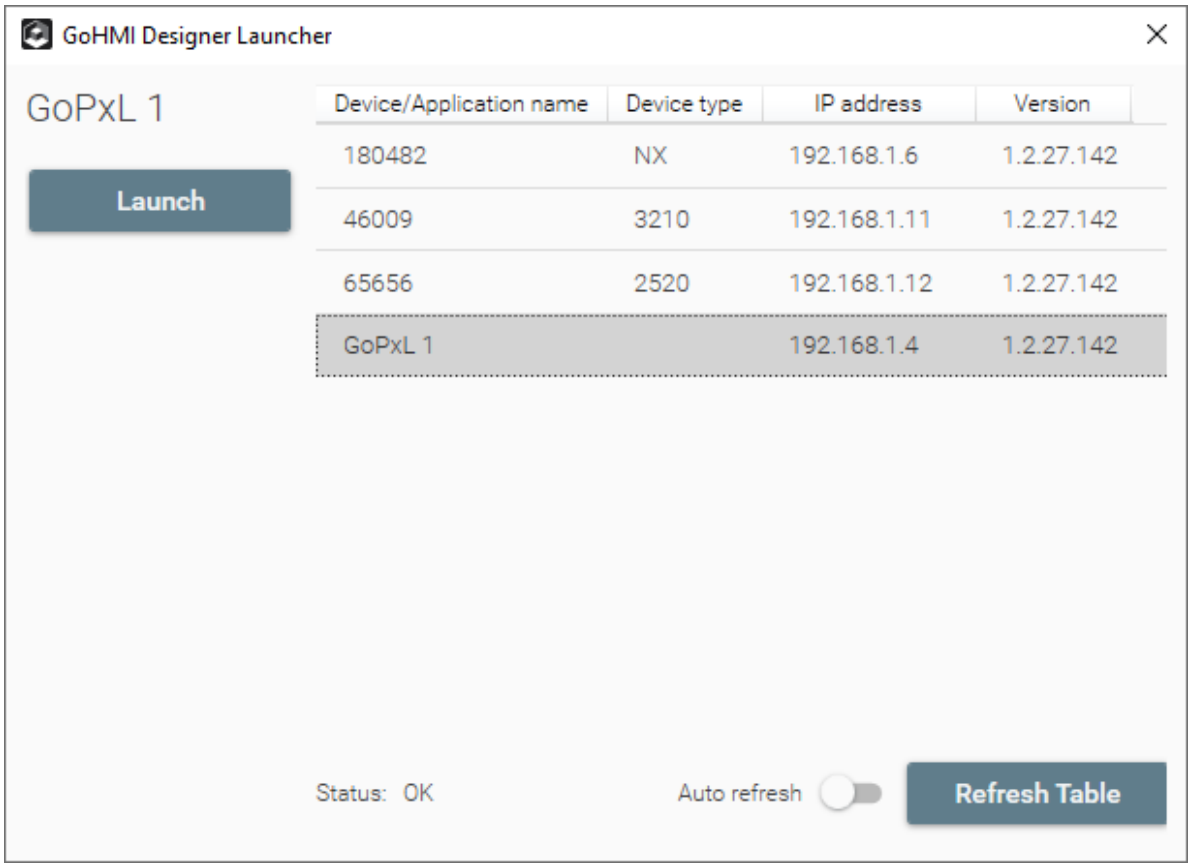
1. Go to where you extracted the GoPxL tools, and double-click the GoHMI Designer shortcut.



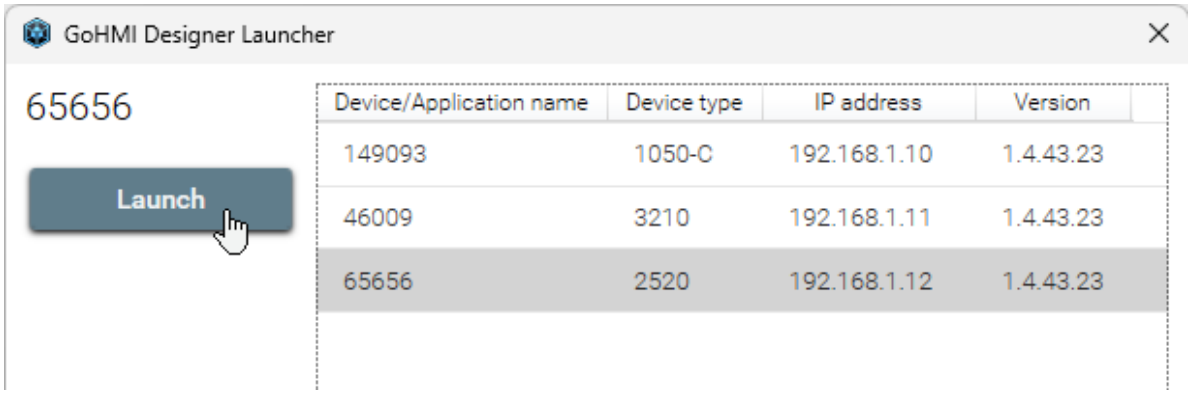
2. (Optional) If a Windows Security Alert opens, enable both networks and click **Allow access**.



GoHMI Designer Launcher opens and lists the available instances of GoPxL. In the following, we see two sensors, a PC instance of GoPxL called "GoPxL 1," and a GoMax device.



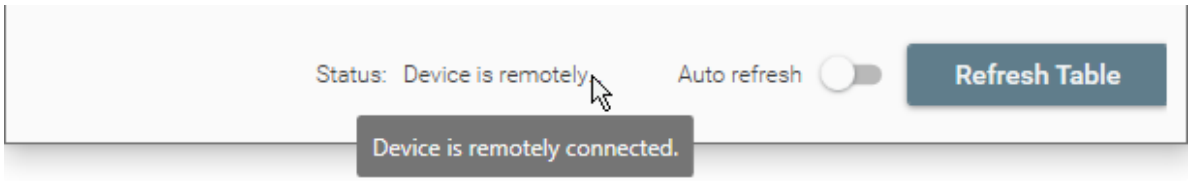
3. In the launcher, select the row of the GoPxL instance for which you want to launch GoHMI Designer and click **Launch**.



You may need to refresh the table to get the most up-to-date list by clicking **Refresh Table**. You can toggle **Auto Refresh** to have the table refresh automatically.

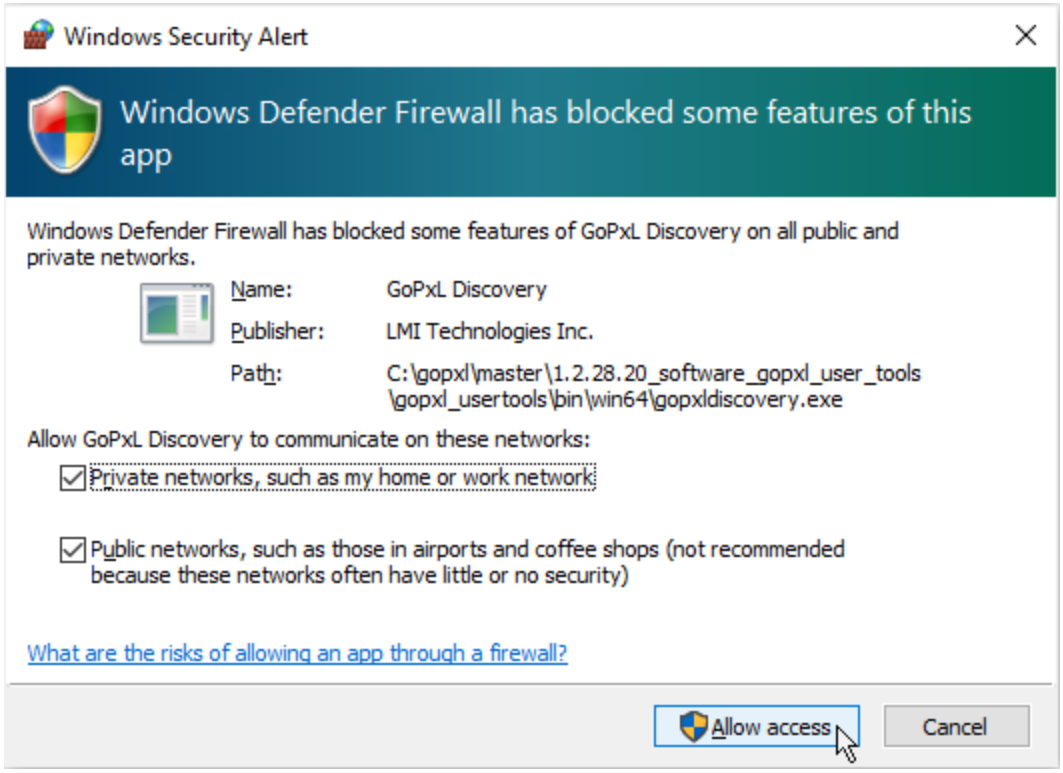
Note that you can only launch one instance of GoHMI Designer at a time.

If a sensor is currently running through a PC instance of GoPxL, its status is "Device is remotely connected." In this case, you can only launch GoHMI Designer from the PC instance through which the sensor is running.

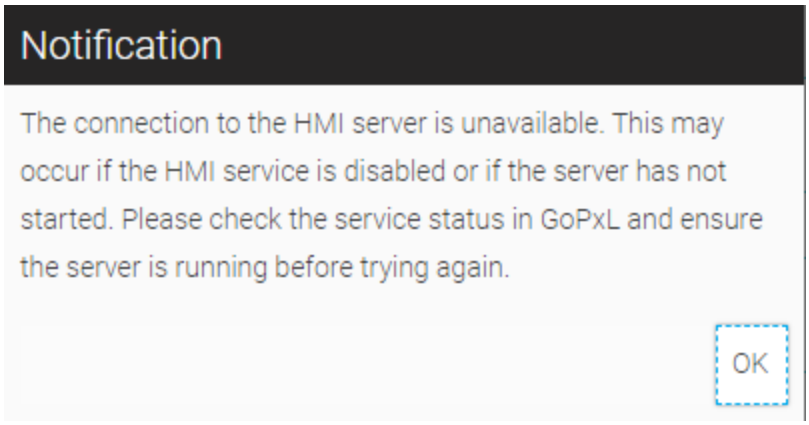


The launcher closes and GoHMI Designer starts.

A Windows Security Alert may appear. If so, allow the designer to communicate over both networks.



If you see a notification warning that the connection to the server has been interrupted, make sure you have enabled the HMI service in GoPxL. After enabling the service, click OK in the notification. (For more information, see *Enabling and Configuring GoHMI in GoPxL* on page 1216.)



4. If necessary, log in to the designer.

   By default, use "gohmi" for the user name and "gohmi" or "gohmi123" for the password.

Opening, Editing, and Saving a Project

This section describes how to open the GoHMI project (called GoHMI), modify it, and save it.

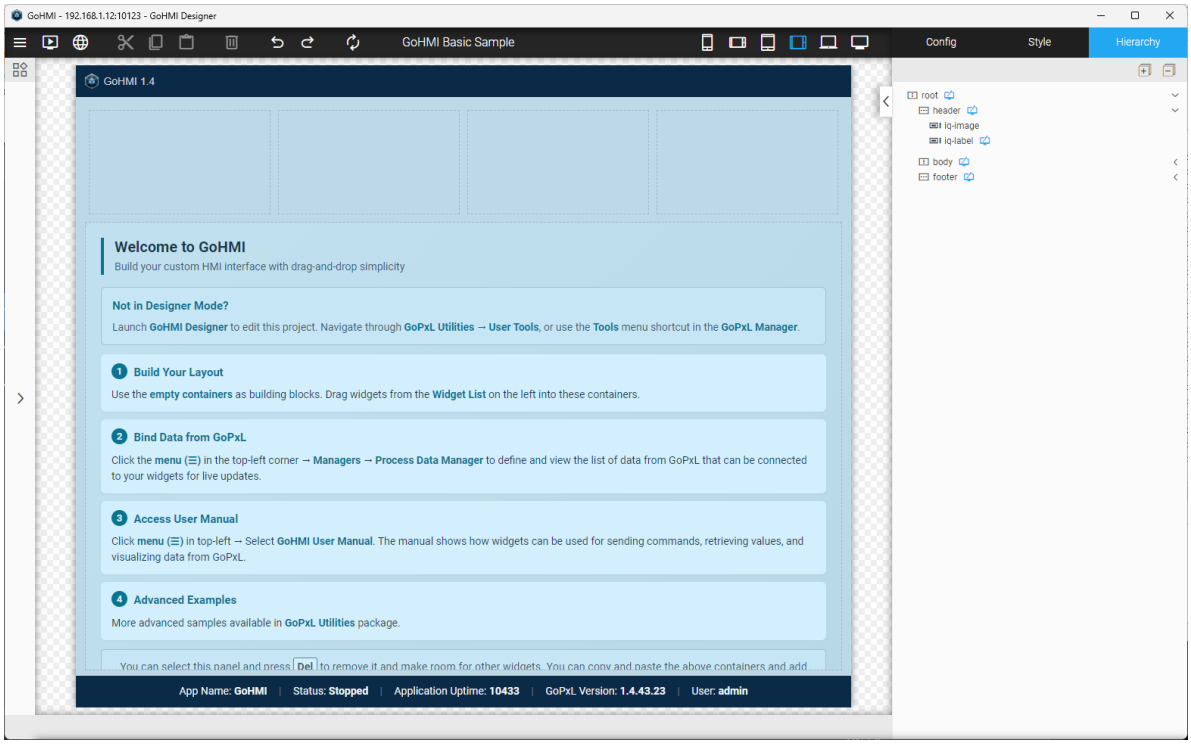
> GoHMI Designer only supports a single HMI application at a time.

To open a project

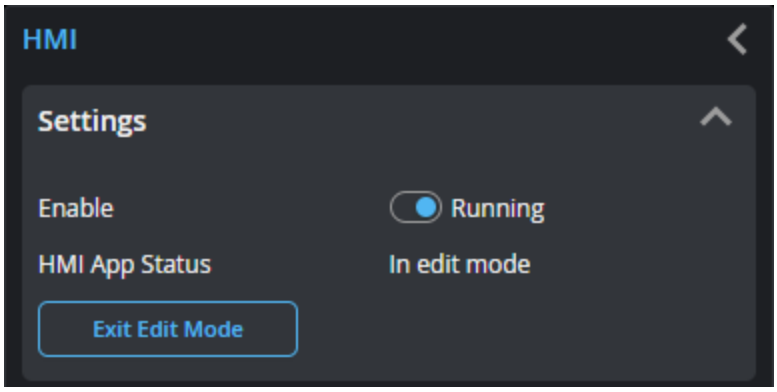
1. In the GoHMI Designer main screen, hover the mouse over the GoHMI project and click it.



The project opens in the Layout Manager.

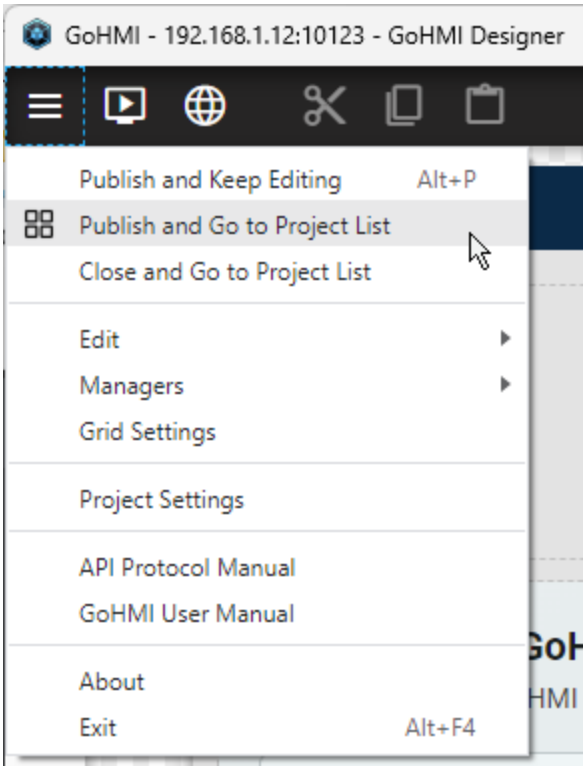


Note that after the HMI loads in GoHMI Designer, in the GoPxL interface the HMI app status becomes "In edit mode."



Publishing Changes or Exiting

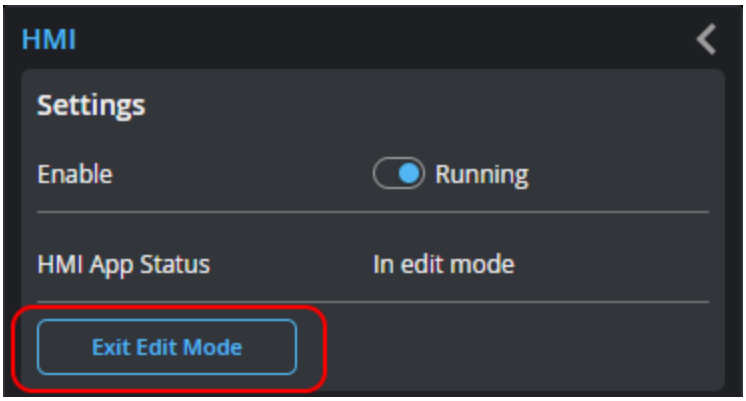
As you work on an HMI, the designer automatically saves changes locally. To save changes onto a device (sensor or GoMax) or into a PC instance of GoPxL, you must "publish" the HMI app using one of the two options from the main menu at the upper left of the Designer.



When you have finished working on your HMI, in the menu, you should choose either **Publish and Go to Project List** or, if you have previously published all changes, **Close and Go to Project List**. Then, from the project list, close the designer by clicking the Close icon (✕) at the upper right.

In some situations, such as when you have closed the designer without publishing unneeded changes, you can "force" exiting edit mode by clicking **Exit Edit Mode** in GoPxL, in the

Communicate > **HMI** page. Note that doing this removes any unpublished changes in GoHMI Designer, so make sure it is safe to do this.



Changing Mapped Outputs in GoPxL

Before trying to add or remove measurements at a later date to outputs sent to the HMI application, the **HMI App Status** in GoPxL must say "Ready". If **HMI App Status** shows "In edit mode" and if you have unpublished changes in the designer that you want to keep, follow the steps in *Launching GoHMI Designer* on page 1220 and then, in GoHMI Designer, publish the changes. For more information on publishing changes, see *Publishing Changes or Exiting* on the previous page. Otherwise, you can force GoPxL to exit edit mode by clicking **Exit Edit Mode**, but you will lose any unpublished changes.

Importing and Exporting an HMI

You can import an HMI app from the GoPxL interface. After importing an HMI app, you can then edit it in GoHMI Designer. After you have imported an HMI you can also launch it in a browser from the GoPxL interface by clicking **Start** and then **View live**.

⚠	Importing an HMI app overwrites the HMI app currently in GoHMI Designer. Consider exporting the HMI app from the GoPxL interface on a regular basis.

To back up an HMI you have created or to provide it to someone else, you can export it.

🗋	Imported and exported HMI apps are zip files.

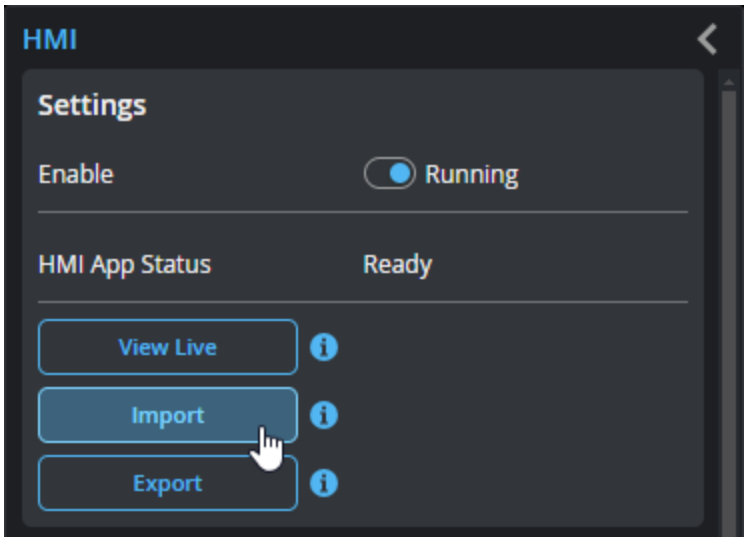
To import an HMI app

1. Make sure you have backed up the current HMI or that it is OK to overwrite it.

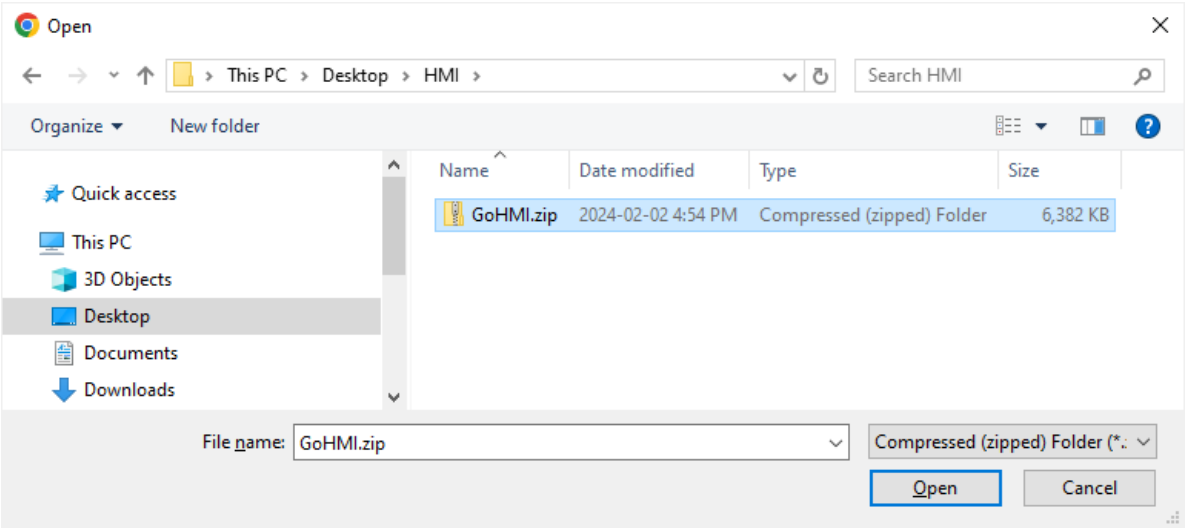
   Importing an HMI app overwrites the app in GoHMI Designer.

   For information on exporting an HMI, see *To export an HMI app* on the next page.

2. In the GoPxL interface, click **Import**.



3. In the Open dialog that displays, go to the location of the zipped HMI app project and click **Open**.

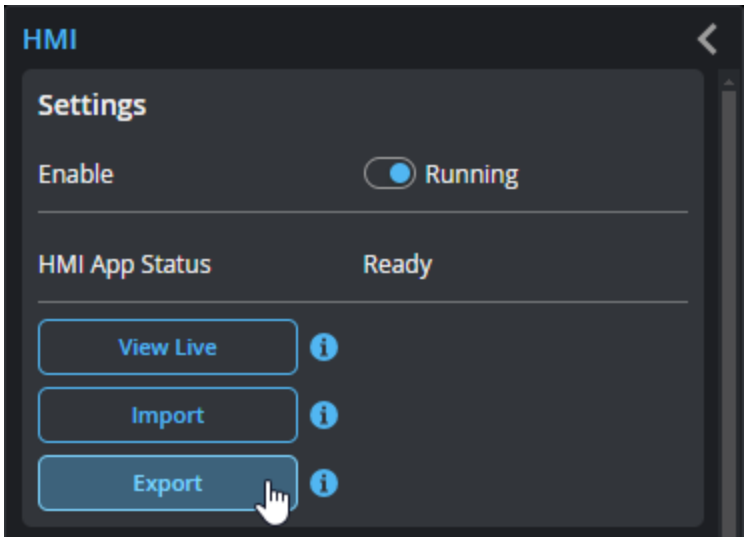


GoPxL imports the HMI app. The app is now available for editing in GoHMI Designer (see See *Opening, Editing, and Saving a Project* on page 1224).

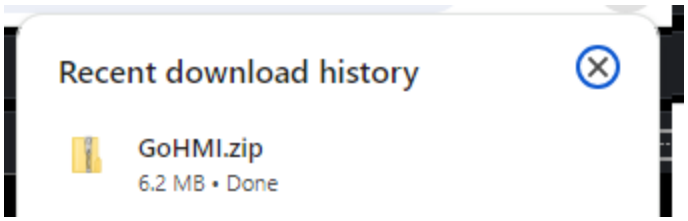
If you want to back up your HMI app or import it into another device or PC instance, you export it in GoPxL interface.

To export an HMI app

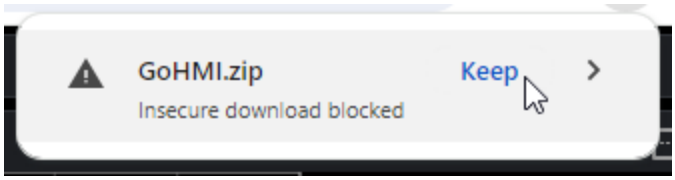
- In the GoPxL interface, click **Export**.



GoPxL downloads the HMI app and saves it as a zip file (GoHMI.zip).



You may need to tell your browser to accept the download (Chrome).



Using a Data Viewer in an HMI

You add a data viewer to an HMI by adding an IFrame widget in GoHMI Designer, and then copying a link from the GoPxL web interface (in the Displayed Outputs panel) and pasting it into the IFrame. The link tells the HMI the IP address of the sensor, as well as which sensor output and measurements it should display.

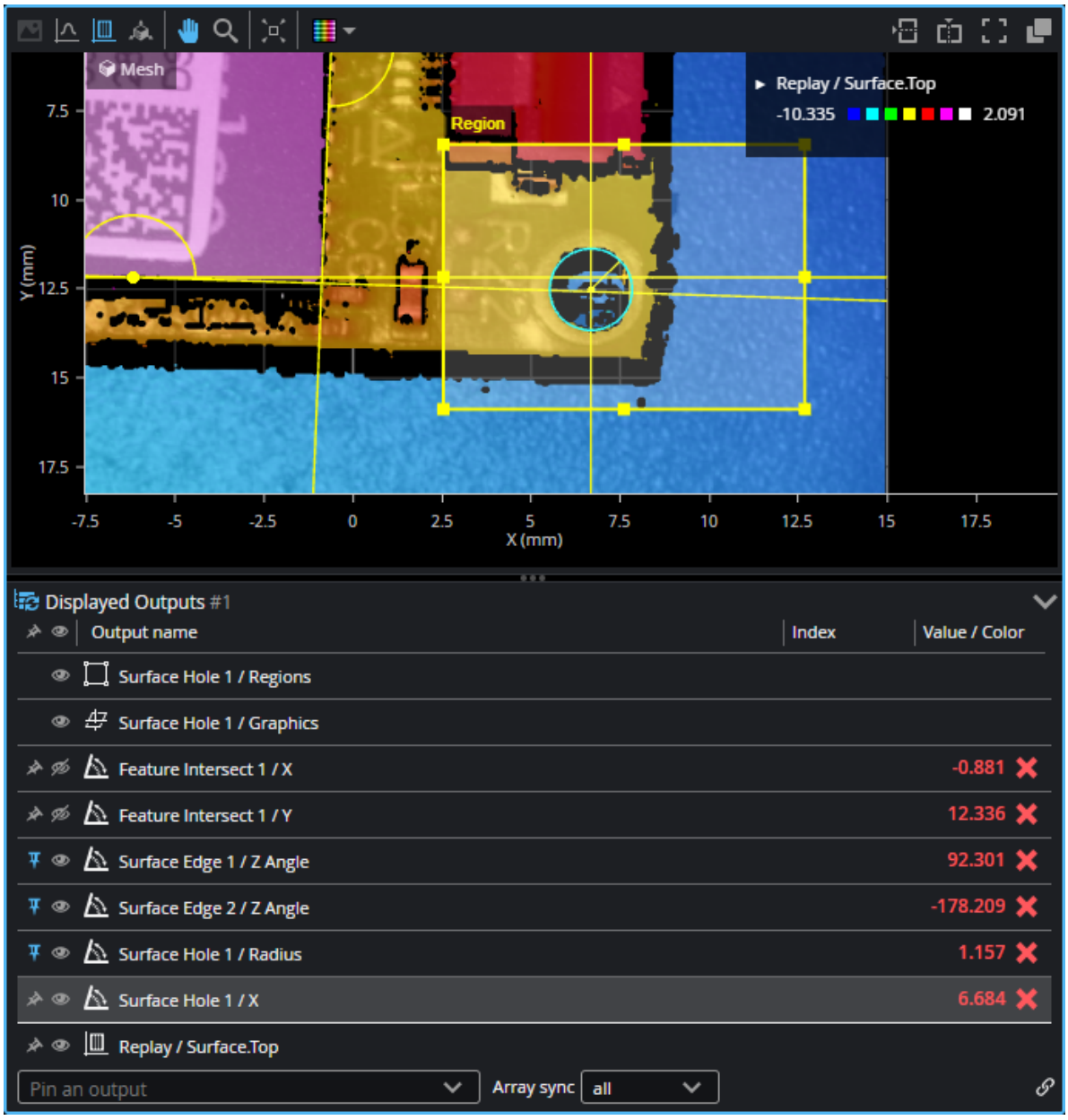
> The default HMI app includes an IFrame widget configured to display the data viewer. The IP address used in the link is 127.0.0.1:8100, which is valid for a PC instance of GoPxL (if one is running at that address and port).
>
> If you want to try out the default app with your sensor, follow the steps below to get the link from your sensor, and update the IFrame's **Source URL** field with that link (see steps 1 to 3, and step 8).

To add a data viewer to an HMI

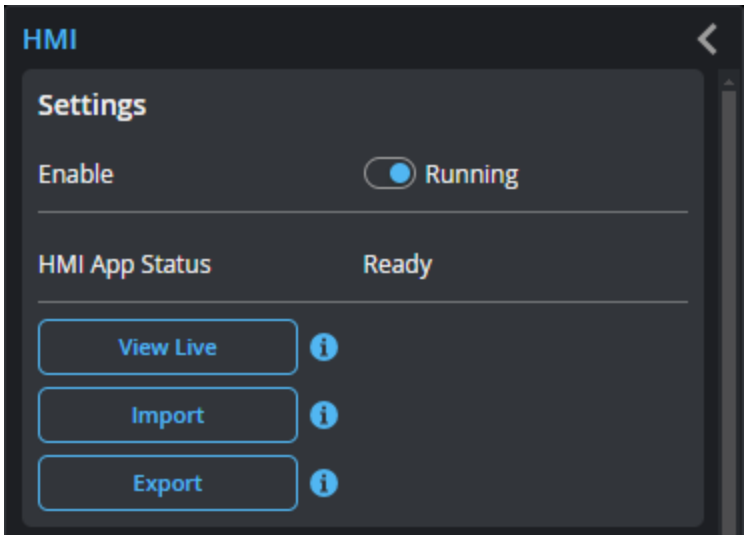
1. In GoPxL, set up a data viewer as required.

   For example, you can pin measurements so that they appear in the data viewer.

2. Under the data viewer window, make sure the Displayed Outputs pane is expanded.



3. Click the link icon ( ) in the lower right of the Displayed Outputs pane to copy the path for the data viewer.
4. Make sure GoHMI is enabled and configured in the **Communicate** > **HMI** page.

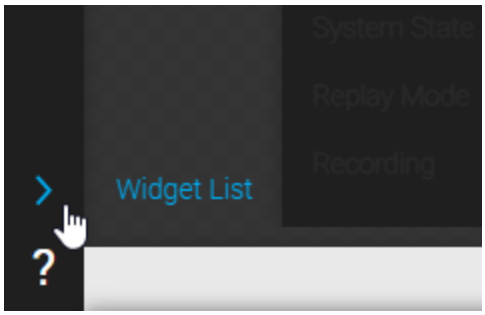


For more information, see *HMI* on page 1153.

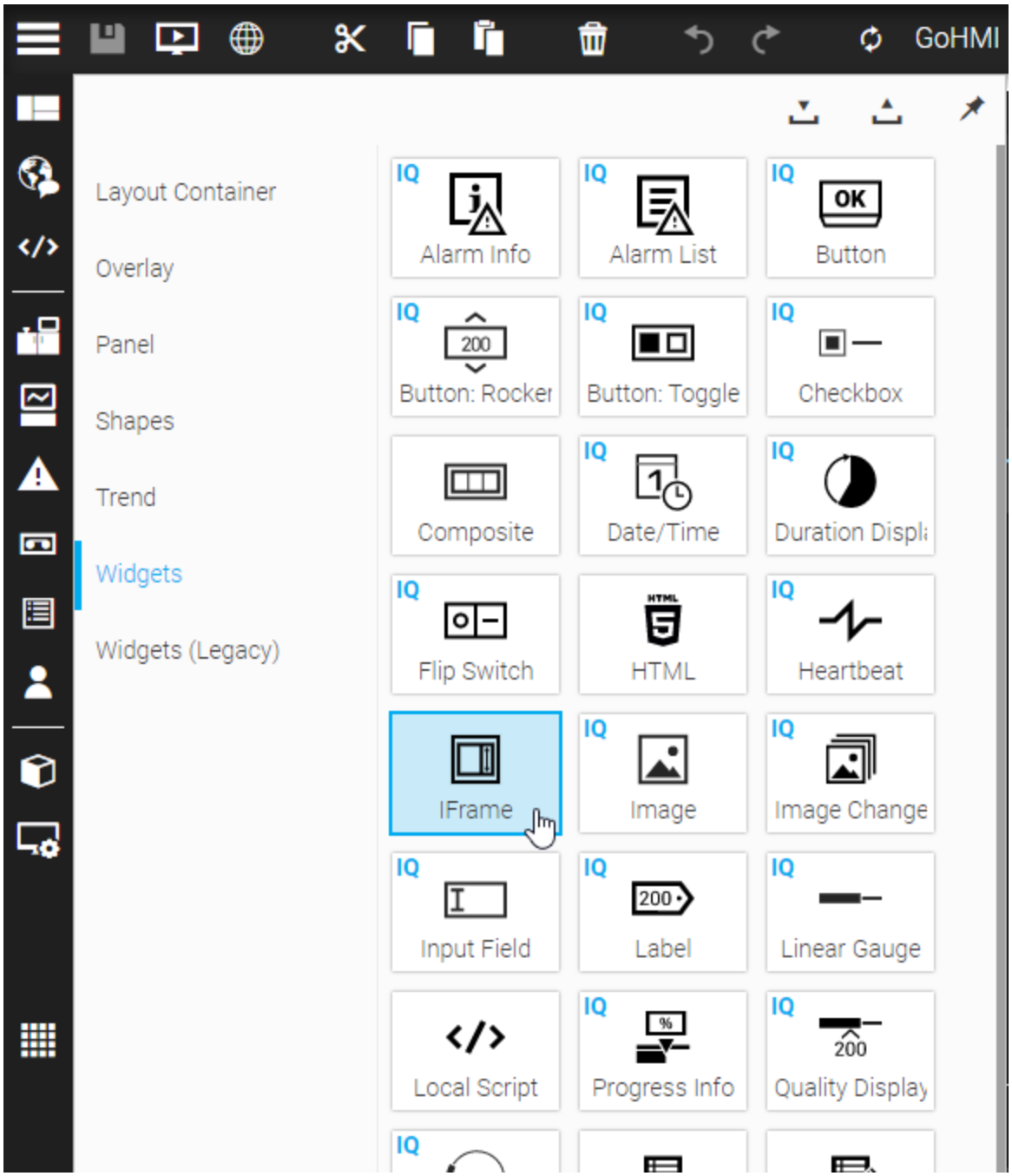
5. Launch GoHMI Designer.

   For information on launching GoHMI Designer, see *Launching GoHMI Designer* on page 1220.

6. In GoHMI Designer, in the Layout Manager, expand the widget list.

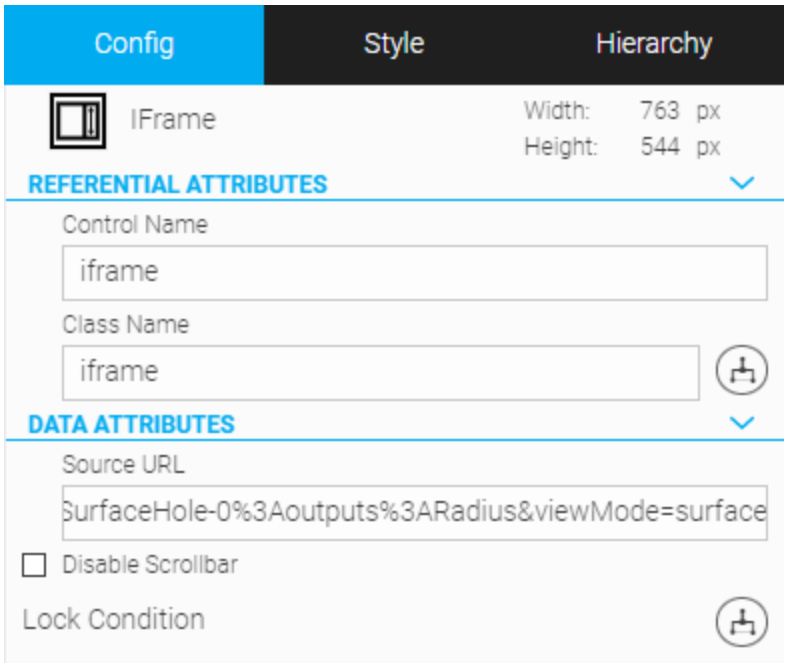


7. From the widget list, drag an IFrame widget into the HMI.



Resize the IFrame widget if necessary.

8. In the **Config** tab, paste the URL you copied in step 1 into **Data Attributes** > **Source URL**.

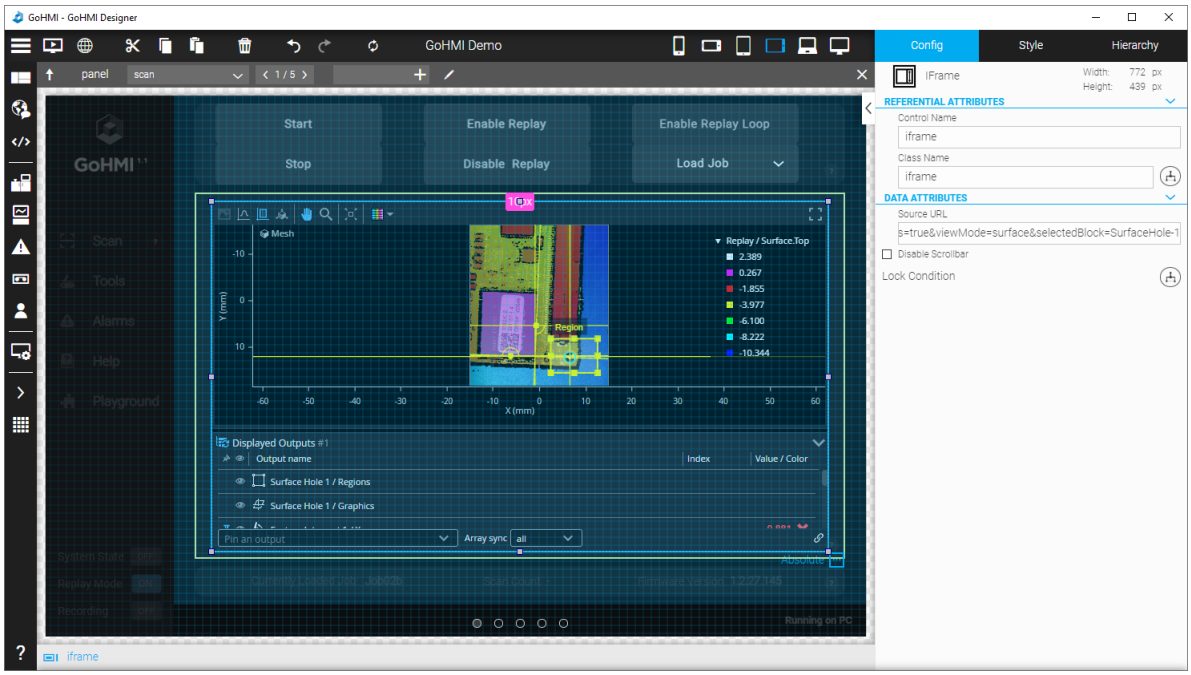


The URL will look something like this:

```
http://127.0.0.1:8100/?host=127.0.0.1&webPort=8100&visualizerOnly=true&displayedOutputs=Surface.Top,tools%3AFeatureIntersect-5%3Aoutputs%3AX,tools%3AFeatureIntersect-5%3Aoutputs%3AY,tools%3ASurfaceEdge-2%3Aoutputs%3AZAngle,tools%3ASurfaceHole-1%3Aoutputs%3AX&viewMode=surface
```

After a few seconds, the data viewer from the connected instance of GoPxL appears in the IFrame widget.

The following shows what the default HMI app looks like after updating the IFrame widget's source URL.



Limiting Flash Memory Write Operations

Several operations and REST API / 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 or GoHMI.



Power loss during flash memory write operation will also cause sensors to enter rescue mode.

GoPxL SDK and REST API

The GoPxL Software Development Kit (GoPxL SDK) includes open-source software libraries and documentation that you can use to programmatically access and control GoPxL devices.

> The GoPxL SDK and REST API use the term "scanner" instead of "sensor group." The two are interchangeable.

> The GoPxL SDK uses mm, mm^2, mm^3, hertz, microseconds, and degrees as standard units unless otherwise mentioned.

The following sections describe the Software Development Kit (GoPxL SDK).

Installation Guide

Location of GoPxL SDK and Documentation

The GoPxL SDK, and the SDK and REST API references are available in the `14630-x.x.x.x_GoPxL_SDK.zip` package.

The zip file contains both a C++ and a .NET SDK organized as follows:

- `GoPxL_SDK_Cpp\`
- `GoPxL_SDK_dotNet\`

Visual Studio Solution Files and Linux Makefile

For Windows, Visual Studio solution files are included for both the C++ and .NET SDKs.

- C++: `GoPxL_SDK_Cpp\Gocator`
- .NET: `GoPxL_SDK_dotNet\Gocator`

For Linux, first ensure that you have the g++ compiler package and then run the following command from the `GoPxL_SDK_Cpp\Gocator\GoPxLSdk` directory:

```
make -f GoPxLSdk-Linux_X64.mk
```

Note that only the C++ SDK is supported on Linux.

Building the SDK

To compile in Visual Studio, you may need to retarget the solution to the installed Windows SDK version. To retarget a solution, open the solution in Visual Studio, right-click the solution in the

Solution Explorer, choose **Retarget solution** in the context menu. In the dialog box that appears, select the desired version from the drop-down list and click **OK**.

The compiled Linux output libraries can be found in the `GoPxL_SDK/lib/linux_x64` directory.

Running SDK Applications

When running an SDK application, be aware that performance may vary significantly between Debug and Release configurations. The Debug configuration disables optimizations and includes additional debugging information, which may result in slower execution times, specifically during asynchronous data retrieval.

System Requirements

The GoPxL SDK can be compiled using the following environment.

For Windows:

- Visual Studio 2017 or later
- Windows SDK (included with Visual Studio installation)

For Linux:

- GCC X64 compiler, version 7.5.0 (GccX64_7.5.0-p3)
- GNU Make

Accessing Documentation

The full SDK class references and REST API resources are available at the following locations:

GoPxL SDK Class Reference

C++: `GoPxL_SDK_Cpp\doc\class-reference.html`

.NET: `GoPxL_SDK_dotNet\doc\class-reference.html`

REST API Resources

C++: `GoPxL_SDK_Cpp\doc\api-protocol-manual.html`

.NET: `GoPxL_SDK_dotNet\doc\api-protocol-manual.html`

Sample Projects

Sample projects showing how to perform various operations are provided, each one demonstrating a specific area. Examples are located in the following folders:

- C++: `GoPxL_SDK_Cpp\Gocator\GoPxLSdk\samples`
- .NET: `GoPxL_SDK_dotNet\Gocator\GoPxLSdkNet\samples`

Header Files

When including header files from the GoPxL SDK, reference the GoPxLSDK directory as the source directory. For example:

`#include <GoPxLSdk/GoRestClient.h>`.

Additionally, note that the SDK header files reference files from the kApi directory. For example:

`#include <kApi/kApiDef.h>`.

Getting Started with the SDK

GoPxL offers a simple programming API interface for connecting, configuring, and acquiring data from sensors and the GoMax vision accelerator.



Example code shown below is for C++.

All GoPxL SDK programs should have the sequence described below, at a minimum.

Connecting a Device

The following sequence assumes you know the IP address of the device. If you do not know the IP address, or if the address is dynamic, an application can discover the devices that are available on the network using the GoPxL SDK discovery capability. For more information, see *Discover Devices* on page 1238.

1. Initialize the SDK library

   For C++ SDK programs only, construct the SDK library by calling `GoApiLib_Construct`. You must call `GoApiLib_Construct` before any API functions are used. (.NET SDK programs don't require this.)

```
1 // Constructs GoPxL API core framework.
2 GoApiLib_Construct(&goApiLib)
```

   After completing your operations with the SDK library, don't forget to release the resources using `kDestroyRef()`.

```
1 // Releases resources and resets the object handle to kNULL
2 kDestroyRef(&goApiLib)
```

2. Instantiate a `GoSystem` object and connect the device.

```
1 // Define device IP address and control port
2 constexpr const char* DEVICE_IP_ADDRESS = "192.168.1.10";
3 constexpr const uint16_t CONTROL_PORT = GO_PXL_SDK_DEFAULT_CONTROL_PORT; // Defaults
  is 3600
```

```
4
5
6  // Parse device IP address
7  kIpAddress deviceIpAddress;
8  kIpAddress_Parse(&deviceIpAddress, DEVICE_IP_ADDRESS);
9
10 // Create a GoSystem object
11 GoSystem system;
12
13 // Set up connection
14 system.SetAddress(deviceIpAddress);
15 system.SetControlPort(CONTROL_PORT);
16
17 // Connect to the device
18 system.Connect();
```

GoSystem Class Member Functions

The `GoSystem` class represents a single Gocator system. A Gocator system consists of one or more sensors/devices. Below are the member functions of the GoSystem class

```
1  // SetAddress: Sets the IP address used for connecting to the device.
2  void SetAddress(const kIpAddress& ipAddress);
3
4  // Address: Gets the IP address used for connecting to the device.
5  kIpAddress Address() const;
6
7  // SetControlPort: Sets the control port used for connecting to the device.
8  void SetControlPort(k16u port);
9
10 // ControlPort: Gets the control port used for connecting to the device.
11 k32u ControlPort() const;
12
13 // Connect: Connects to the device.
14 void Connect();
15
16 // Disconnect: Disconnects from the device.
17 void Disconnect();
18
19 // IsConnected: Checks if the device is connected.
20 bool IsConnected();
21
22 // Start: Starts the device.
23 void Start();
24
25 // Stop: Stops the device.
26 void Stop();
27
28 // RunningState: Gets the current state of the device.
29 State RunningState();
30
31 // Client: Returns a reference to the REST API client for performing operations.
32 GoRestClient& Client();
```

```
33
34 // SensorPath: Retrieves the REST API resource path of a sensor based on its serial
   number.
35 ResourcePath SensorPath(SerialNum serialNum);
```

Discover Devices

If a device IP address is not known (for example, because it is assigned through DHCP), you can discover the devices available on the network using the GoPxL SDK discovery function:

1. Include the `GoDiscoveryClient` header file.
2. Create an instance of `GoDiscoveryClient` and call the `BlockingDiscover` function with a specified timeout.
3. Access the list of discovered instances.

```
1 #include "GoDiscoveryClient.h"
2 auto discovery = std::make_unique<GoDiscoveryClient>();
3 discovery->BlockingDiscover(3000);
4 auto& instances = discovery->InstanceList();
```

4. After discovering the instances, you can loop through the instances to get IP addresses and other instance-related information.

```
1 // Print instance IP address and GDP port.
2 for (const auto& instance : instances) {
3    std::cout << "IP Address: " << kIpAddress_AsString(instance.GetIpAddress()) <<
  std::endl;
4    std::cout << "GDP Port: " << instance.GetGdpPort() << std::endl;
5 }
```

For more information, see the *Discover* sample.

Configure Device Settings

You configure GoPxL device settings, which are represented as JSON resources, through JSON-based REST API commands. The `GoRestClient` class provides access to read and write settings. You can use methods like Call, Read, Update, Create, and Delete, among others, to interact with the device settings.

Start by instantiating a `GoSystem` object. Through this object, you can directly access the `GoRestClient` instance via `system.Client()`. Then, to perform `Read()` or other operations, simply continue chaining the methods, for example, `system.Client().Read()`.

These methods follow a similar format:

```
"command /path/to/callable/resource {'argument':'value'}"
```

For example the following command:

```
call /jobs/commands/rename { "sourceName" : "job1", "destName" : "renamedJob1"
}
```

can be implemented as follows:

```
// Rename job.
auto payload = GoJson(R"({
    "sourceName" : "SDK-demo",
    "destName" : "SDK-demo-job"
})");

system.Client().Call("/jobs/commands/rename", payload);
```

For the full list of resources available, see the *GoPxL 1.1 - REST API Resources and User Guide*:

C++: `GoPxL_SDK_Cpp\doc\api-protocol-manual.html`

.NET: `GoPxL_SDK_dotNet\doc\api-protocol-manual.html`

It's important to understand that GoPxL device settings are divided into two general types:

- sensor-level: These settings only apply to individual sensors.
- scanner-level: These settings apply to all of the sensors in a group of sensors (a "scanner").

To help explain the functions, some sections below include definitions of the API resources.

Accessing Sensor Resources

The most common use case is modifying settings that apply to individual sensors, such as exposure or active area.

If you know a device's serial number, you can access the corresponding resource using the `SensorPath` function. You can then modify the settings under the sensor's resources.

```
// Update single exposure value.
auto sensorPath = system.SensorPath(sensorSerialNumber);
auto payload = GoJson(R"({
    "parameters" : {
        "exposureSettings" : {
            "exposureMode" : 0,
            "singleExposure" : 400
        }
    }
})");

system.Client().Update(sensorPath, payload);
```

Accessing Scanner Resources

You access system-level settings (those that apply to a group of sensors) through the `scanner` resources.

If you know the engine ID and scanner ID (available in the REST API reference documentation), you can use the following to access the sensor group and modify the trigger setting:

```
// Access and modify trigger settings for the scanner.
auto scannerPath = "/scan/engines/LMILaserLineProfiler/scanners/scanner-0";
auto payload = GoJson(R"({
    "parameters" : {
        "triggerSettings" : {
            "source" : 3
        }
    }
})");

system.Client().Update(scannerPath, payload);
```

Configuring scanner resources through the SDK is less common. These resources are related to physical setup. Normally, you configure these settings through the user interface during the initial system installation and setup, and the settings don't need to change for different production runs.

Accessing an Array Element

Some resources are returned as an array. One example is the `visibleSensors` resource, which can have multiple elements.

The following returns an array of sensors under the `/scan/visibleSensors` resource path.

```
sensors = system.Client().Read("/scan/visibleSensors/").GetResponse().Content().At
("sensors");

// Get the first element in the array
auto sensor = sensors.Begin();
auto sensorSerialNumber = sensor.Value().At("/serialNumber").Get<std::string>();

// Alternatively, iterate through each sensor in the array
for (auto sensor = sensors.Begin(); sensor != sensors.End(); ++sensor)
    auto sensorSerialNumber = sensor.At("/serialNumber").Get<std::string>();
}
```

Receive Device Data

GoPxL streams out acquired data (such as Profile and Surface data, and measurements) using the GoPxL Data Protocol (GDP). To receive the data, you use a `GoGdpClient` instance, which you first

need to connect to the device. You can use `GoPxLSdk::MessageType` to identify the type of data received and the associated stamp information. Stamps contain encoder and trigger timing information, current time, and frame index, and are grouped into datasets with their corresponding data.

Data can be received on the same thread used for controlling the sensors (that is, synchronously) or on a different thread (that is, asynchronously). Note that data is received as the `GoDataSet` type. For more information on `GoDataSet`, see *GDP Data Types* on page 1244.

For synchronous data retrieval, use the `ReceiveData()` method:

1. Instantiate a GoPxL Data Protocol client and connect to a device.

```
#include <GoPxLSdk/GoGdpClient.h>

constexpr k16u GDP_PORT = GO_PXL_SDK_DEFAULT_GDP_SERVER_PORT; // Default port is 3601

// Instantiate a GoGdpClient object
auto gdpClient = std::make_unique<GoGdpClient>();

// Connect the GoGdpClient to the device using sensor IP and GDP port
gdpClient->Connect(SENSOR_IP, GDP_PORT);
```

2. Receive data.

```
// Establish a synchronous data transfer with a 3-second timeout.
gdpClient->ReceiveDataSync(3000);

// Retrieve the received data set.
const GoDataSet& receivedDataSet = gdpClient->DataSet();
```

   Then loop through each message (a unit of data received from the GoPxL sensor) to handle the data based on its type (Profile, Surface, Measurements, and so on) using the `MessageType` enum. For example:

```
for (size_t i = 0; i < receivedDataSet.Count(); ++i)
{
    switch (receivedDataSet.GdpMsgAt(i).Type())
    {
        case GoPxLSdk::MessageType::STAMP:
            // Handle stamp message
            ...

        case GoPxLSdk::MessageType::UNIFORM_SURFACE:
            // Handle uniform surface data
            ...
    }
}
```

For more information on synchronous data retrieval, see the `ReceiveProfile` and `ReceiveSurface` samples.

For asynchronous data retrieval, you use the following steps:

1. Instantiate a GoPxL Data Protocol client and connect to a device.

2. Start the data reception thread

   Call `GoGdpClient::ReceiveDataAsync(func)` to initiate data reception asynchronously. This starts a background thread (`receiveThread`) to receive and deserialize messages continuously.

3. Handle received data.

   Upon receiving individual data messages, they are reassembled into a `GoDataSet` object. The data reception continues until you call `GoGdpClient::Close()`.

4. Process the data.

   The received data sets are added to a message queue (`Go::MsgQueue<GoDataSet> dataQueue`). A separate thread (`dataThread`) removes the data sets from the queue for processing.

5. Access the data

   Define a callback function to handle the received data sets.

The following is an example of retrieving data asynchronously:

```
void onData(const GoDataSet& receivedDataSet)
{
    for (size_t i = 0; i < receivedDataSet.Count(); i++)
    {
        switch (receivedDataSet.GdpMsgAt(i).Type())
        {
            case GoPxLSdk::MessageType::STAMP:
            {
                const GoGdpStamp& stampMsg = (const
GoGdpStamp&)receivedDataSet.GdpMsgAt(i);
                // Process stamp message
                ...
            }
            // Handle other message types as needed
        }
    }
}

// Connect to GDP and start receiving data asynchronously
gdpClient->Connect(SENSOR_IP, GDP_PORT);
gdpClient->ReceiveDataAsync(onData);
```

For details, see the `ReceiveAsync` sample.

The following lists the `GoGdpClient` class methods you can use to interact with the GoPxL Data Protocol (GDP) server.

```
// Constructs GoGdpClient.
GoGdpClient();

// Destructs GoGdpClient.
~GoGdpClient();

// Connects to the GdpServer.
void Connect(kIpAddress ipAddr, k16u port);

```

```
// Closes the TCP connection.
void Close();

// Receives data synchronously from the GDP server.
void ReceiveDataSync(k64u receiveTimeoutInMilliseconds);

// Receives data asynchronously from the GDP server.
void ReceiveDataAsync(std::function<void(const GoDataSet& receivedDataSet)>& func);

// Gets the pointer to the received data set over the Goctor Data Protocol.
const GoDataSet& DataSet() const;
```

Receive Metrics Information

Health information, along with various metrics, is transmitted using the REST API. To handle received metrics data, set up a callback function as shown below:

```
// Callback function for handling received metrics data.
static void onMetrics(const GoStreamResponse&)
{
    // Extract payload & content from the received data
    auto content = notification.Payload();
  int appUptime = content.Get<int>("/appUpTime");
  ...
}

// Set up the callback function for receiving metrics data asynchronously.
system.Client().SetStreamHandler(onMetrics);

// Start the stream to receive metrics data from the specified path
system.Client().StartStream("/system/metrics");

// Stop the stream and disconnect from receiving metrics data, call StopStream
system.Client().StopStream("/system/metrics");
```

For more information, see the `ReceiveMetrics` sample.

Accelerated Devices

Control of devices accelerated by GoMax or a PC instance of GoPxL uses the same function API. The only difference is that you must use the IP address of the GoMax device or the PC running GoPxL when connecting to the device (that is, in `system->Connect()`).

GDP Data Types

Value Types

The GoPxL SDK 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	For the definitions of this type, in the folder containing the GoPxL SDK, see Platform\kApi\kApi\kApiDef.h.
kIpAddress	IP address

Output Types

Data outputs are encapsulated as data messages in the GoPxL SDK. The following message types are available in the SDK.

Output Types

Type	Description
GoGdpBoundingBox	Bounding Box results output based on part matching results
GoGdpFeaturePoint	Feature point output
GoGdpFeaturePlane	Feature plane output
GoGdpFeatureLine	Feature line output
GoGdpFeatureCircle	Feature circle output
GoGdpImage	Video image

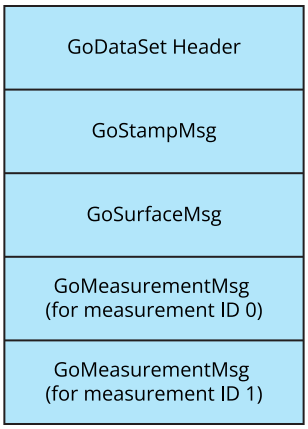
Type	Description
GoGdpMeasurement	Measurement output
GoGdpMesh	Mesh data output
GoGdpNull	Contains no valid data. Received when a measurement tool's data output doesn't generate data when triggered. This messages allow user to synchronize message from different outputs based on number of message received.
GoGdpProfilePointCloud	Profile point cloud data (non-uniform spacing)
GoGdpProfileUniform	Uniform spacing / resampled profile data
GoGdpRendering	Graphics objects associated with a tool outputs*
GoGdpSpots	Detected spots when overlay exposure is enabled*
GoGdpStamp	Acquisition stamp
GoGdpSurfacePointCloud	Surface point cloud data (non-uniform spacing)
GoGdpSurfaceUniform	Uniform spacing / resampled surface data

*Not commonly used in a production system.

All of the messages above are extended from `GoGdpMsg`. For more information, see *GoDataSet* below.

GoDataSet

Data is passed to the data handler in a `GoDataSet` object. The `GoDataSet` object is a container that can hold any type of data, including scan data (sections or surfaces), measurements, or results from various operations. Data inside the `GoDataSet` object is represented as messages. The following illustrates the content of a `GoDataSet` object of a Surface mode setup with two measurements.



The following lists the available GoDataSet types:

```
enum struct MessageType : k16u {
    SIGNAL                  = 1,    // Signals that data on a stream is invalidated.
    NULL_TYPE               = 10,   // Null data type that contains a status code.
```

```
    STAMP                   = 11,   // Stamp information.
    UNIFORM_PROFILE         = 12,   // Uniform profile data.
    PROFILE_POINT_CLOUD     = 13,   // Raw profile data.
    UNIFORM_SURFACE         = 14,   // Raw surface data.
    SURFACE_POINT_CLOUD     = 15,   // Raw surface data.
    IMAGE                   = 16,   // Image data.
    SPOTS                   = 17,   // Spot data associated with the image.
    MESH                    = 18,   // Mesh data.
    MEASUREMENT             = 19,   // Measurement data.
    RENDERING               = 70,   // Graphical data associated with an output.
    POINT_FEATURE           = 71,   // Point feature data.
    LINE_FEATURE            = 72,   // Line feature data.
    PLANE_FEATURE           = 73,   // Plane feature data.
    CIRCLE_FEATURE          = 74,   // Circle feature data.
    HEALTH                  = 100   // Health data.
};
```

Measurement Values and Decisions

There are two outputs for each measurement: a value and a decision. A measurement value is returned as a 64-bit number.

The measurement decision specifies the state of the measurement value as follows:

Measurement Decisions

Decision	Description
1	The measurement value is between the maximum and minimum decision values (inclusive). This is a pass decision.
0	The measurement value is outside the maximum and minimum. This is a fail decision.

Limiting Flash Memory Write Operations

Several operations and REST API / 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 or GoHMI.



Power loss during flash memory write operation will also cause sensors to enter rescue mode.

Python GDK

The GoPxL Python GDK lets you develop custom measurement and processing tools that you can incorporate into your application. Once integrated into a GoPxL or GoMax firmware, the tools behave like the built-in ones. Python GDK tools are also available in the PC-based GoPxL software; for more information, see *Running GoPxL on a Windows PC* on page 1188.

As with the Script tool, you use Python to add functionality and custom algorithms to GoPxL. However, using the Python GDK to create tools has these advantages:

- You can expose parameters to control the algorithm.
- Your intellectual property is better protected, if you bundle the custom tool in a firmware.
- You can debug your Python code in an IDE that supports Python.

GoPxL includes a Python interpreter and tool server that provide support for Python-based tools and the Script tool. When you use GoPxL on PC, the Python engine is started by default as part of launching a GoPxL instance.

When you develop custom tools for GoPxL, you can use an external Python interpreter, instead of the included one, and optionally a virtual environment. Doing this is useful if you will develop tools using specific libraries or environments, for example, implementing an AI program with the GDK.

Requirements and Installation

If you are using the distribution of Python provided with the GDK, there is nothing to install. If you are using a Python distribution installed on your PC, LMI recommends version 3.8.10, as later versions have not been tested with the GDK.

You should also install the following packages using `pip install`.

- numpy
- opencv-python (optional, currently 4.5.4.60)

You can get the Python GDK by downloading and unzipping the GoPxL Utilities package (14631-x.x.x.x_GoPxL_Utilities.exe, available on LMI's Product Downloads page). The Python GDK is under the `GoPxL/plugins/GdkPy` folder.

The following lists the files you will typically need to use or modify in the `/plugins/GdkPy` folder:

- `configure_developer_start.bat`: Disables the built-in Python server in PC instances of GoPxL. When the built-in server is disabled, the `run_server.py` program (see below) provides Python GDK tools to any PC instance run from the parent folder. You *must* run this batch file before starting a PC instance to debug custom tools, or if you need to use your

own distribution of Python to run the tools. .

- `configure_built_in_start.bat`: Enables the built-in Python server in PC instances of GoPxL. You *must* run this script to re-enable the internal server before starting an instance of GoPxL if you are not using your own distribution of Python.
- `run_developer_server.bat`: Starts the Python tool server program (run_server.py) from the GDK's Python distribution. You should use this to start the Python tool server, instead of `run_server.py`.
- `run_server.py`: The Python tool server program, which loads the tools in the `plugins/GdkPy/user_tools/` (and `plugins/GdkPy/samples/`) and sets up logging. The script runs in the Python environment that executes it. You typically run this program from within a debugger, but you can also run it from the command line to run tools that use your own Python distribution. For information on debugging, see *Debugging Custom Tools* on page 1252. For information on deploying tools, see *Deploying Your Custom Tools* on page 1254.
- `gdk` folder: This folder contains the `tool.py` and `tool_api_def.py` files. Together, these files provide the tool API and data types you use to create custom tools, respectively. You should familiarize yourself with the content of these files.
- `samples` folder: This folder contains several sample Python GDK tools. You should copy these into the files into the `user_tools` folder as a starting point for your own custom tools.
- `standard_tools` folder: This folder contains the Script tool and the library it uses. You do not need to modify these files.
- `user_tools` folder: You should put your custom Python GDK tools and any required libraries in this folder. When you run `run_server.py`, the tool server loads these tools.
- `utilities` folder: This folder contains utilities you use to add a custom tool to a GoPxL firmware or GoPxL for PC. For more information, see *Deploying Your Custom Tools* on page 1254.

Developing Custom Tools

For full code completion such as IntelliSense and debugging support, we recommend using Visual Studio Code or an IDE that supports Python. In the following, we use Visual Studio Code.

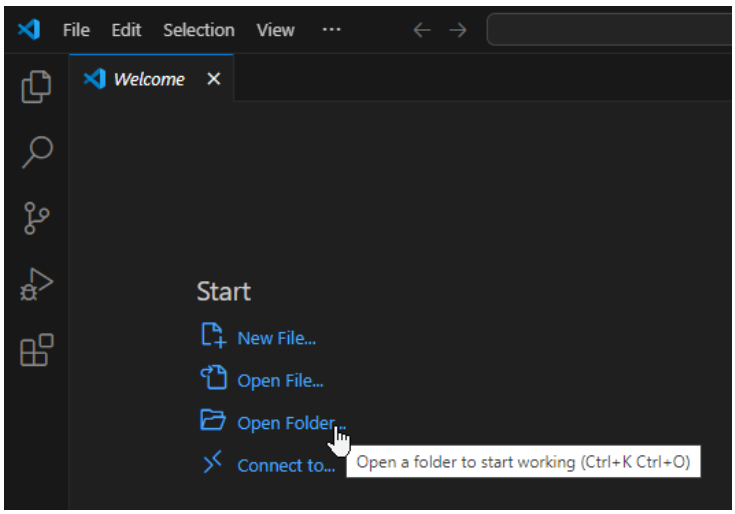
For information on setting up debugging, see *Python GDK* on the previous page.



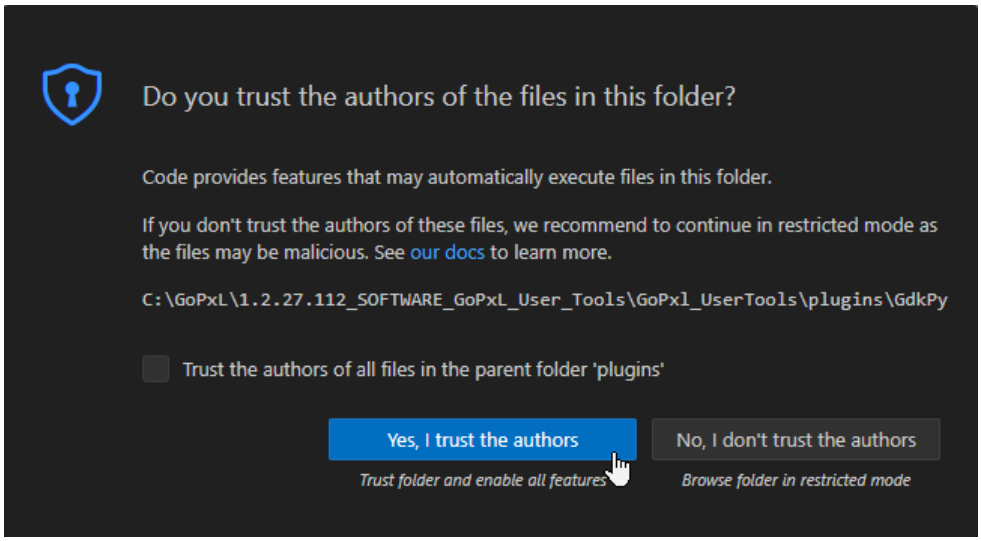
OpenCV is only available on GoPxL running on a PC and on GoMax. To use OpenCV in a tool, you must use `import cv2` in your tool's Python source code. Gocator sensors do not support the OpenCV module in Python tools.

To create a Python GDK tool

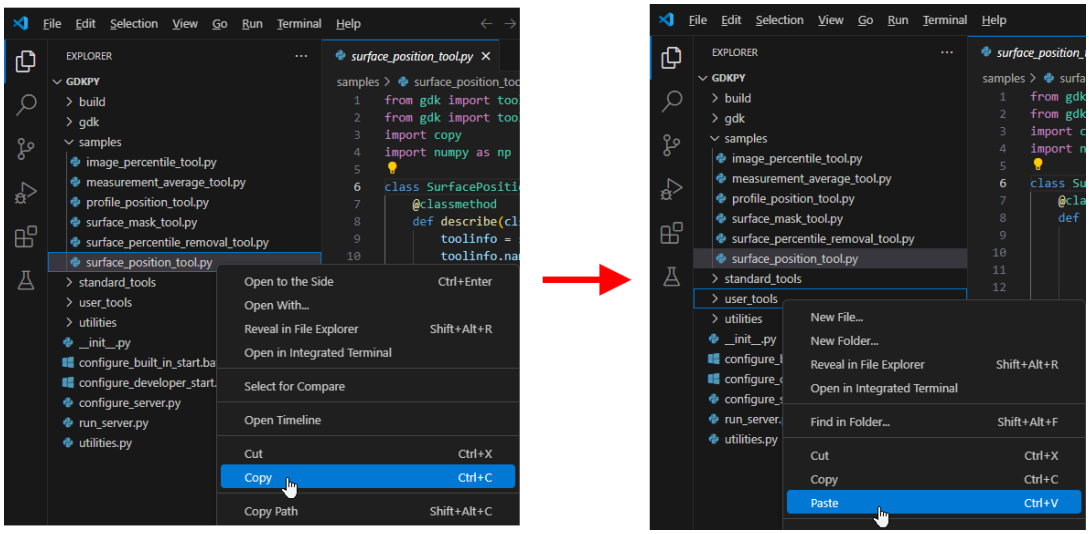
1. If you have not already done so, install Visual Studio Code, start the program, and install the Python Extension.
2. Open the `plugins\GdkPy` folder in the IDE.



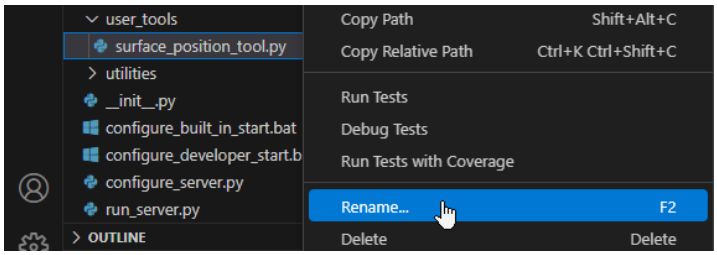
You may need to confirm that you trust the authors of the files in the folder.



3. Copy a file from the `samples` folder into the `user_tools` folder and rename it.

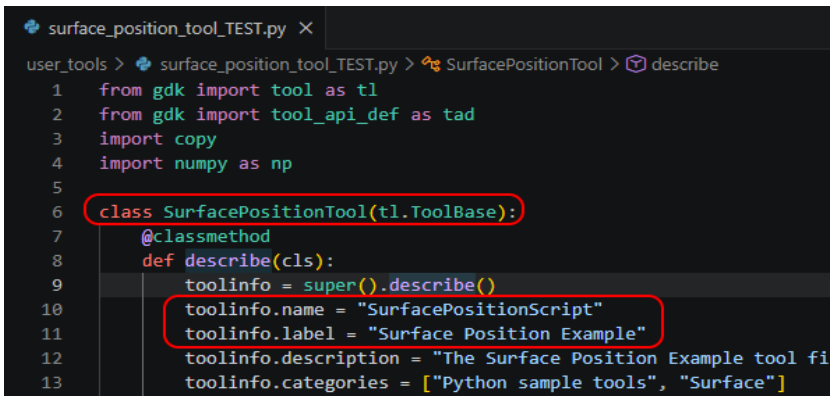


You can use any name for the tool's Python file, but using the main input type as a prefix is a useful convention, for example, "surface_" for Surface tools.



4. Modify the tool's code.

To start (and to avoid conflicts), you should change the name of the class everywhere in the code, as well as the strings assigned to `toolinfo.name` and `toolinfo.label` in the `describe` method.



You also define the tool's inputs, anchors, outputs, and parameters in the `describe` method. Note that the order of the items of a given category (that is, inputs, parameters, outputs, or anchors) in the code is the order in the tool.

```
class SurfacePositionTool(tl.ToolBase):
    @classmethod
    def describe(cls):
        toolinfo = super().describe()
        toolinfo.name = "SurfacePositionScript"
        toolinfo.label = "Surface Position Example"
        toolinfo.description = "The Surface Position Example tool finds the X or Y or Z axis posit
        toolinfo.categories = ["Python sample tools", "Surface"]

        # Inputs
        toolinfo.inputs["Input"] = tl.InputInfo("Surface", tl.DataType.UNIFORM_SURFACE)

        # Anchors
        toolinfo.anchor = tl.AnchorInfo(x=True, y=True, z=True, z_angle=False)

        # Outputs
        toolinfo.outputs["X"] = tl.OutputInfo("X", tl.DataType.MEASUREMENT_VALUE_X)

        toolinfo.outputs["Y"] = tl.OutputInfo("Y", tl.DataType.MEASUREMENT_VALUE_Y)

        toolinfo.outputs["Z"] = tl.OutputInfo("Z", tl.DataType.MEASUREMENT_VALUE_Z)

        toolinfo.outputs["Point"] = tl.OutputInfo("Position", tl.DataType.FEATURE_POINT)

        # Parameters
        toolinfo.parameters["UseRegion"] = tl.ParamInfo("Use Region", tl.ParamType.BOOL, True)
        toolinfo.parameters["Region"] = tl.ParamInfo(
            "Region",
            tl.ParamType.SURFACE_REGION,
            tl.Region3d(0, 0, 0, 5, 5, 5),
        )

        # feature position options
        cls.position_options = {0: "Average", 1: "Median", 2: "Centroid", 3: "Min X", 4: "Max X",
        toolinfo.parameters["Features"] = tl.ParamInfo(
            "Features", tl.ParamType.INT, 0, options=cls.position_options
        )

        return toolinfo
```

You should closely examine the `tool.py` and the `tool_api_def.py` files in the `gdk` folder (which form the GDK's API), as well as the source code of the sample tools, to better understand how to develop your own tools. Observing how Python GDK tools work using a Python debugger can also help you develop your own tools.

5. (Optional) Place any additional modules or libraries your tool needs to import in the `user_tools` folder.

Enabling the Local Python Tool Server

Before you can start debugging a custom tool, or run a custom tool on PC using your own distribution of Python, you must run the `configure_developer_start.bat` batch file. This batch file disables the Python tool server built into GoPxL for PC.

After this, you must run the `run_developer_server.bat` script to start the Python tool server manually. If you want to debug a tool, run `run_server.py` instead.

For information on starting the tool server for debugging purposes, see *Debugging Custom Tools* on the next page.

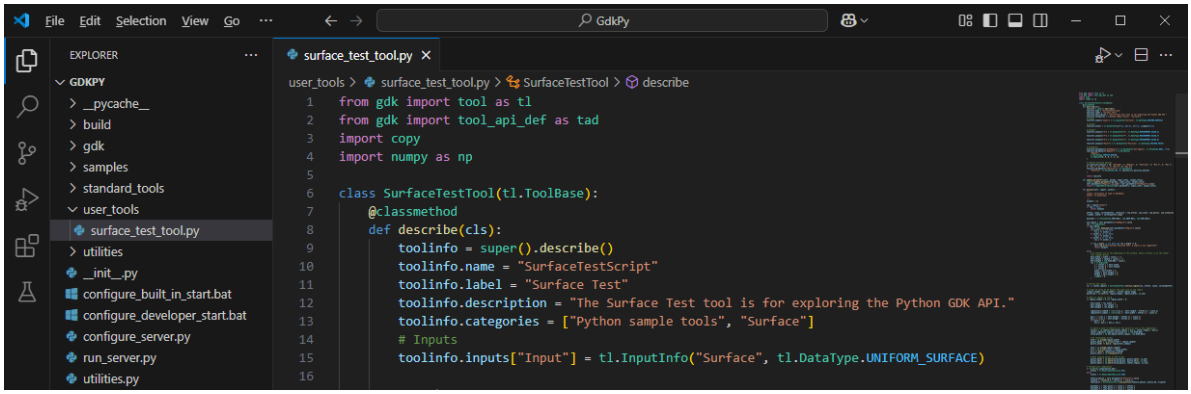
For information on starting the tool server with your own distribution of Python, see *Deploying Your Custom Tools* on page 1254.

Debugging Custom Tools

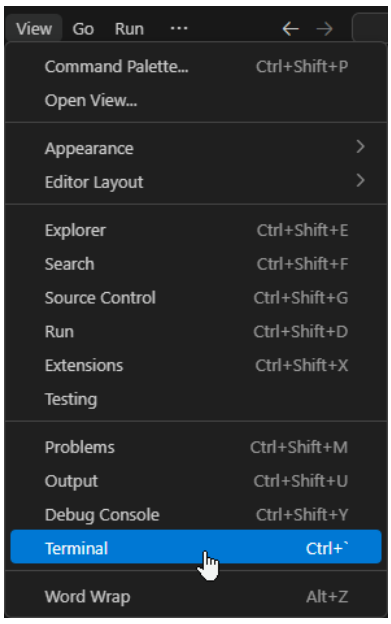
You can debug a Python GDK tool using an IDE or application that supports Python.

To debug your Python tool using Visual Studio Code

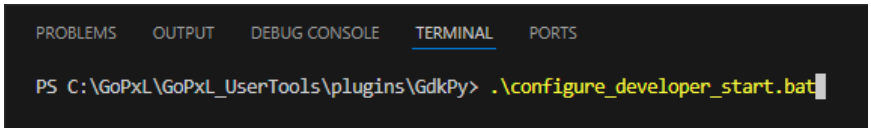
1. Open the `plugins\GdkPy` folder in the IDE.



2. Open the terminal from the **View** menu or by pressing Ctrl+` (Ctrl + back tick).

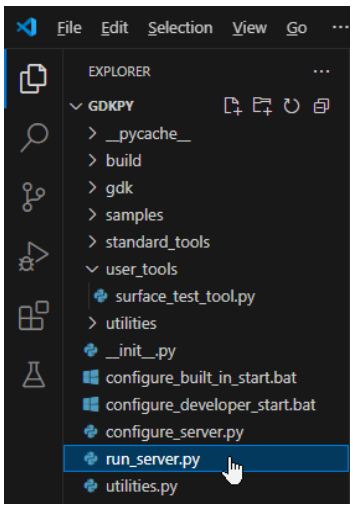


3. At the prompt, type `.\configure_developer_start.bat`.



This disables the Python tool server built into GoPxL for PC. If you do not run this script, you will not be able to debug your tool.

4. From the Explorer view, open the `run_server.py` file.

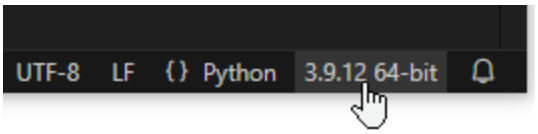


5. (Optional) If Python is not installed on your PC, you can point Visual Studio Code to the Python interpreter used by the internal tool server. To change the interpreter, do the following:

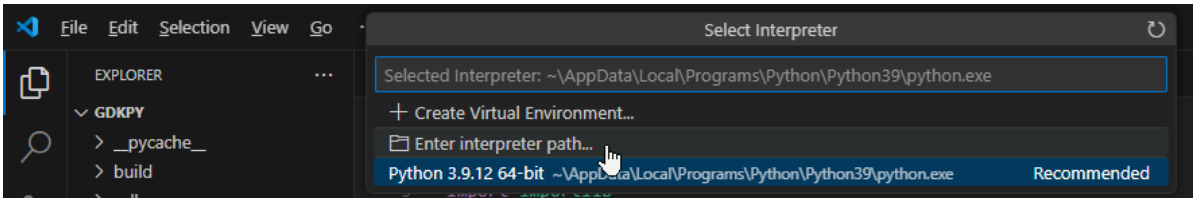
   a. At the lower right, click the Python version in the bottom right corner.



   If you have previously used Python in Visual Studio Code, you may see a version already selected.

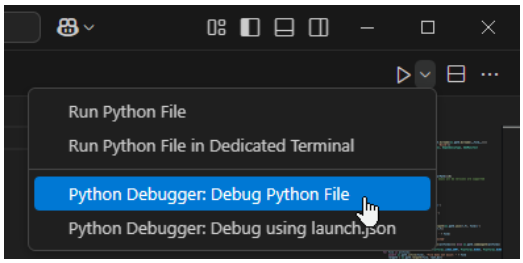


   b. At the top of the editor, click **Enter interpreter path**, and provide the path to the built-in interpreter.

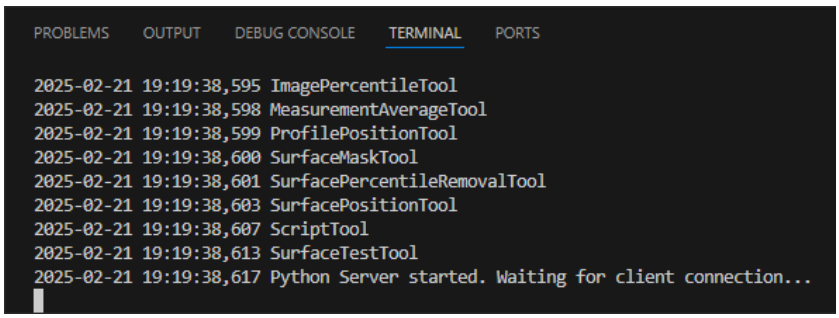


   The built-in Python interpreter is `bin\win64\python\python3.exe`.

6. Start debugging run_server.py.

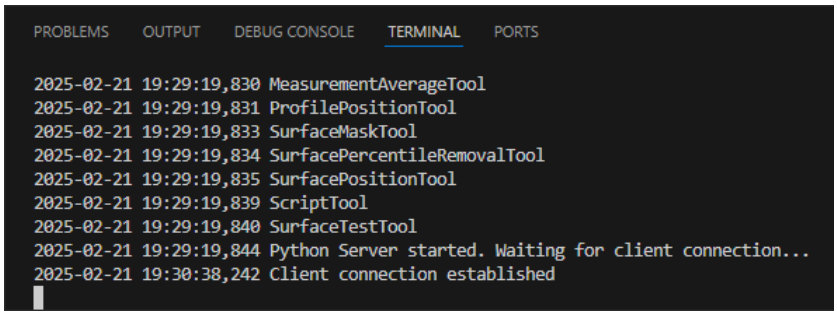


The tool server starts and waits for a connection from a PC instance of GoPxL.



7. Start an instance of GoPxL.

   The instance of GoPxL connects to the tool server.



   For more information, see *Running GoPxL on a Windows PC* on page 1188.

8. Upload some previously scanned data or scan some new data and set the tool's input to this data.

   Your tool's input `must` be set in order to debug it.

9. Set a break point in your tool's Python code.

   The `process` method is good point to start: the tool's main processing occurs here.

Deploying Your Custom Tools

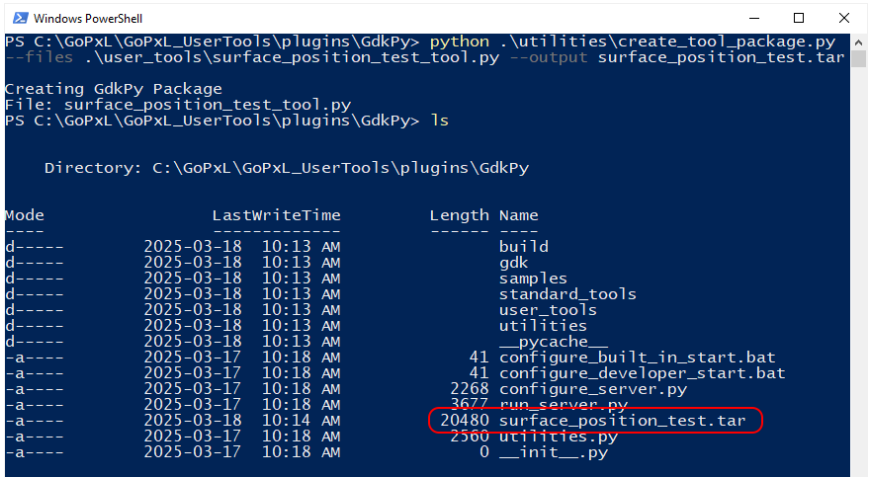
After you have finished developing your tool, you can deploy it as part of either a GoPxL firmware or GoPxL for PC. You use the `GdkPy\utilities\create_tool_package.py` script to create a .tar tool package, which you add to GoPxL using the GoPxL Add-on Tool Manager.

The `create_tool_package.py` script accepts the following arguments:

Arguments

Name	Description
-h, --help	Show the help.
--files	The files to include in the .tar tool package. You can also pass a folder containing multiple Python tools. Files are added recursively.
--output	The name of the tool package with a .tar extension. If you omit this argument, the tool is packaged as `my_tool.tar`.
--ignore	Files or wildcards to ignore. By default, *.pyc files are excluded.

1. Open a command prompt in the folder `plugins\GdkPy` and run `\utilities\create_tool_package.bat` (which calls `create_tool_package.py`), providing the path to your tool.



   The .tar file is created in the folder from which you started the script.

2. Use the GoPxL Add-on Tool Manager utility to add the generated `.tar` tool package to a firmware (`.dat` file) or to GoPxL for PC (in the `GoPxL` folder, where the bin folder is).

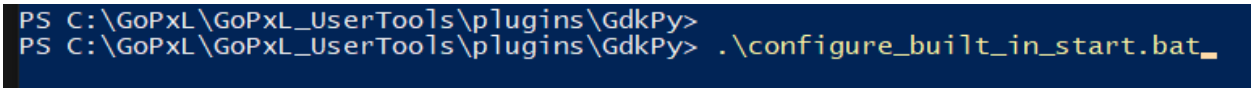
   For more information on using this utility, see *GoPxL Add-on Tool Manager* on page 1300.

3. If you have added the custom tool to GoPxL for PC, open a command prompt and go to the `GdkPy` folder.

   a. Do one of the following:

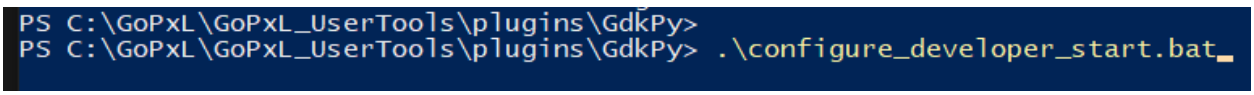
      Using the Python distribution included with GoPxL for PC:

      Run `.\configure_built_in_start.bat`.

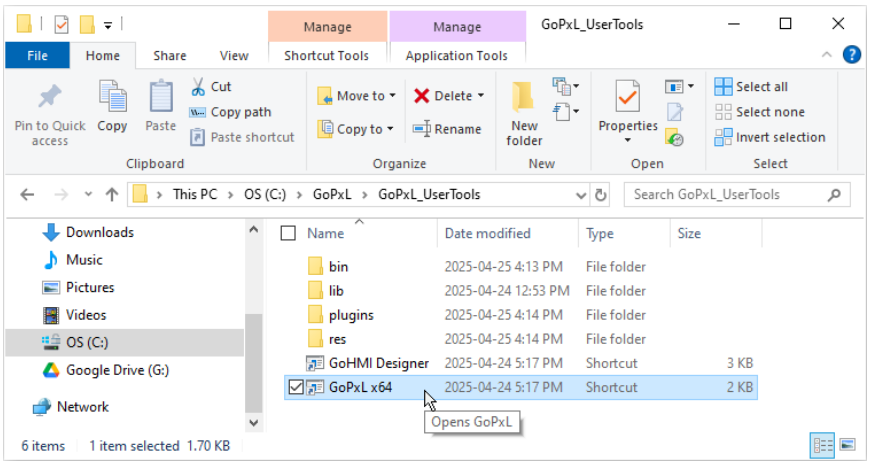


      Using your own Python distribution:

      Run `.\configure_developer_start.bat`.



   b. Go to the main GoPxL_UserTools folder, and start GoPxL.



4. If you have added the custom tool to a firmware, upload it to a device.

   For more information updating a sensor's firmware, see *Software Upgrade* on page 1184.

Integrations

GoPxL currently provides three types of integrations that you configure within GoPxL: GoPxL Data Protocol (for use with the GoPxL SDK and API), industrial protocols, and GoHMI.

For information on the GoPxL data protocol, see *GoPxL Data Protocol* on page 1023.

For information on the protocols supported by GoPxL, see *Protocols (PLCs and other hardware)* below.

For information on GoHMI, see *GoHMI and GoHMI Designer* on page 1210.

Protocols (PLCs and other hardware)

GoPxL supports various protocols for communicating with sensors over Ethernet (TCP/IP).

The following protocols are available:

- *GoPxL Data Protocol* on page 1023
- Modbus
- EtherNet/IP
- PROFINET
- ASCII

> The connection mappings for the industrial protocols (PROFINET, EtherNet/IP, and Modbus) have changed to correct a byte alignment issue. To allow communication between PLCs and GoPxL 1.2, you may need to update PLC logic created for GoPxL 1.1.
>
> For EtherNet/IP, the EDS file has been updated to reflect the changes.

The connection mappings for the industrial protocols (PROFINET, EtherNet/IP, and Modbus) have changed to correct a byte alignment issue. To allow communication between PLCs and GoPxL 1.2, you may need to update PLC logic created for GoPxL 1.1.

For EtherNet/IP, the EDS file has been updated to reflect the changes.

GenICam GenTL Driver

For known issues when using GoPxL with GenTL, see the GoPxL release notes.

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.

Currently, the GenTL driver can't perform an alignment if you haven't saved a job first.

The following sensor scan modes are supported:

- Profile (with **Uniform Spacing** disabled). In this mode, the point cloud 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 and Intensity* on page 139.

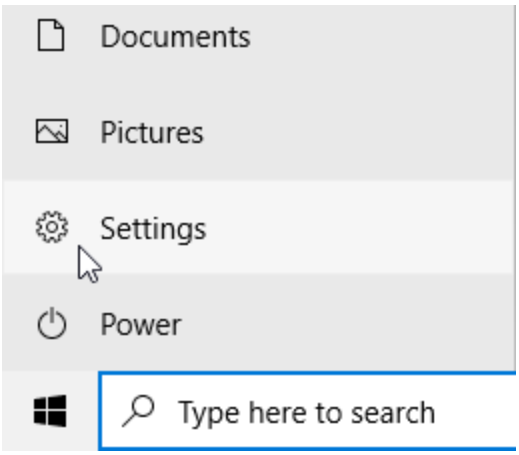
To use these third-party software applications, you *must* configure a system variable so the software can access the GenTL driver. For instructions, see *To configure system variables to use the driver in Windows 10*, below.

To get the utilities package containing the driver (14631-x.x.x.x_GoPxL_Utilities.exe), go to https://lmi3d.com/product-downloads/, choose your product using the drop-down menus, and download the zip package.

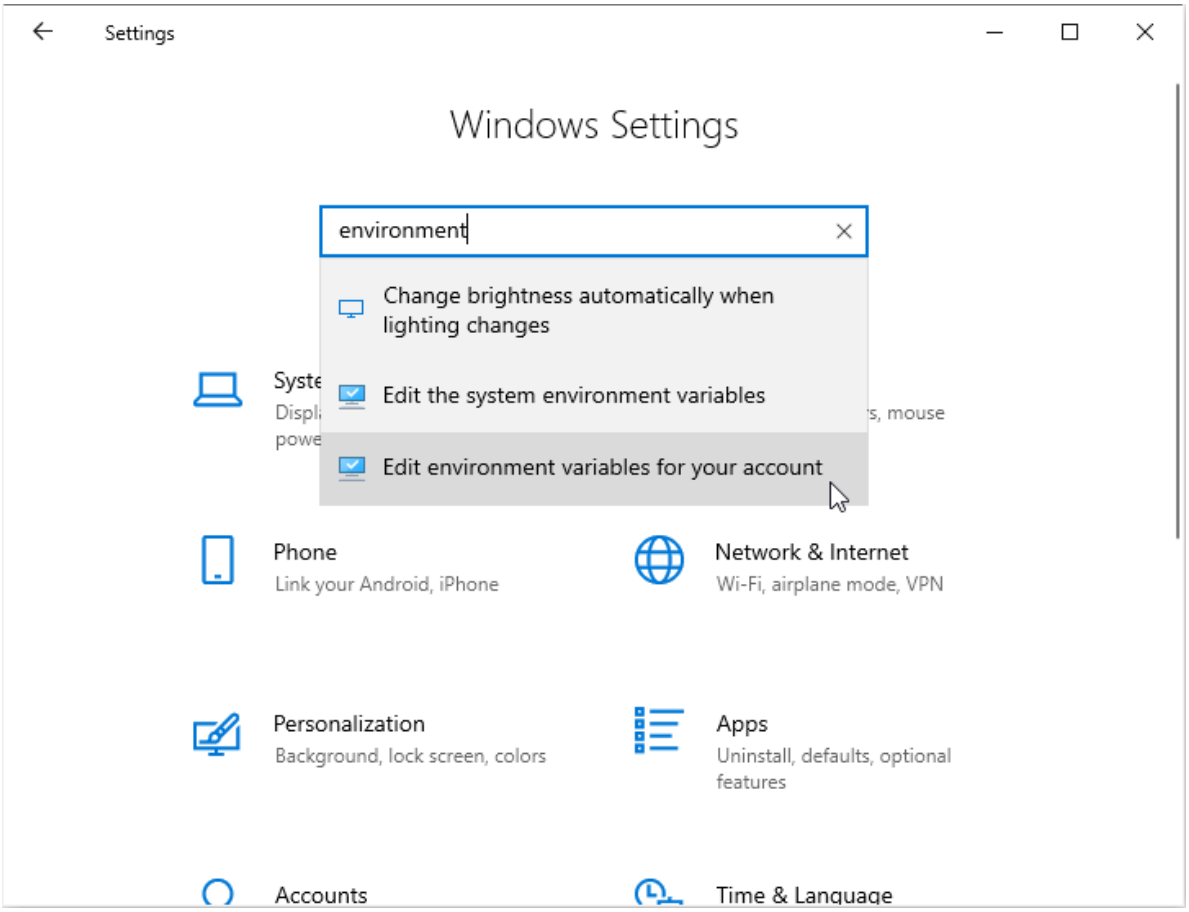
After downloading the package and unzipping the file to a location you will remember, you will find the driver in Integration\GenTL\x64 (you can move the GenTL folder to a more convenient location).

To configure system variables to use the driver in Windows 11

1. Click the Start menu and launch Settings.

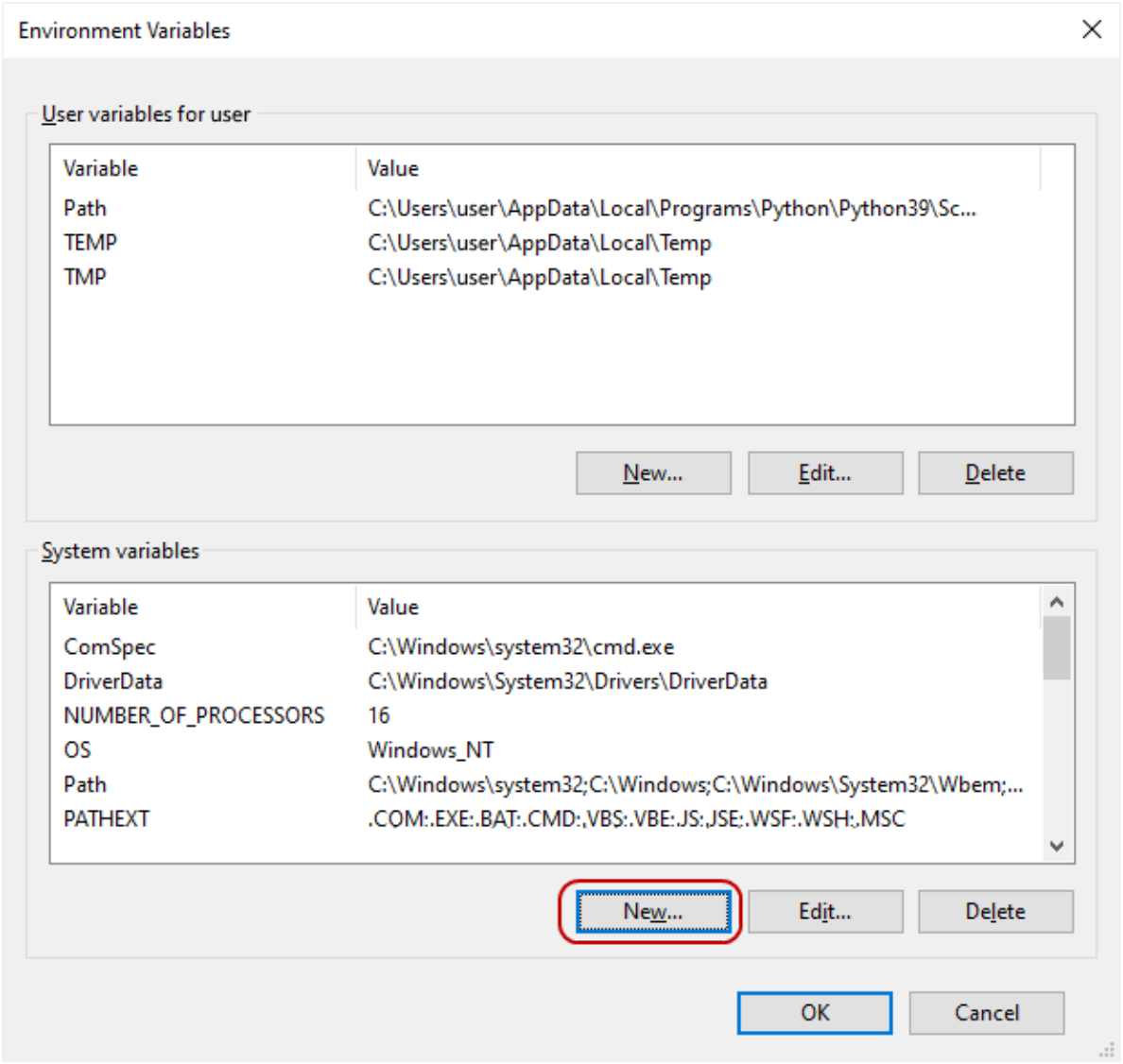


2. In the Settings application search field, type "environment".



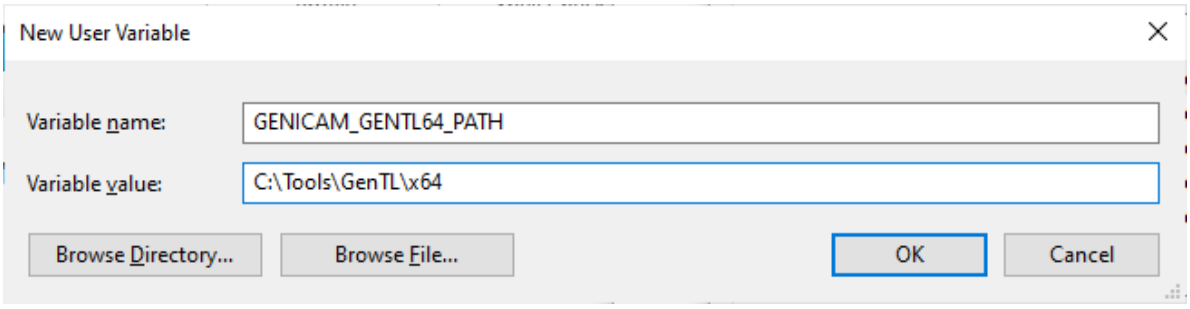
The Environment Variables dialog opens.

3. In the Environment Variables dialog, under the System variables list, click **New**.



4. In the **New System Variable** dialog, enter the following information, depending on your system:

Variable name	Variable value
GENICAM_GENTL64_PATH	The full path to the GenTL\x64 folder.



5. Click OK in the dialogs until they are all closed.

To work with the GenTL driver, you must enable the GoPxL Data Protocol as well as the desired outputs on the **GDP** page. For more information on configuring the protocol, see *GoPxL Data Protocol* on page 1023.

If you need intensity data, check **Acquire Intensity** in the **Scan Mode** panel on the **Acquire** page. For more information on enabling intensity, see *Scan Modes and Intensity* on page 139.

The GenTL driver packs the output and stamps (e.g., time stamp, encoder index, and so on) into either a 16-bit RGB image or a 16-bit grey scale image. The driver also includes intensity if it is enabled. 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.

Channel	Details
	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

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.

Rows	Details
	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.
(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

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

Stamp Index	Column Position	Details
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 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.

Register Address	Name	Read/Write	Length (bytes)	Description
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 ".gpjob". Writing to this register causes the sensor to switch to the 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.
OverrideImageWidth OverrideImageLength	32u	If Halcon is consuming the output of a multi-sensor system or of a tool such as Surface Stitch, that is, where multiple frames are combined, you should set OverrideImageWidth and OverrideImageLength to the maximum expected size. OverrideImageWidth = image width / X resolution OverrideImageLength = image length / Y resolution

Element	Type	Description
		Otherwise, leave the default of 0.
RawPartDetectionHeightThreshold RawPartDetectionMaxLength	Float	If the Gocator sensor is in Profile mode, the GenTL driver converts the profiles the sensor produces into an image that Halcon can process, using these settings. RawPartDetectionHeightThreshold: Detects the start and end of a part when scanning. When data points in the profile go over this threshold, the driver starts concatenating the profiles. As soon as no data points are above this threshold, the driver stops concatenating profiles. RawPartDetectionMaxLength: The maximum length of a part. Either condition (data points going back below the threshold or maximum length reached) will stop profile concatenation for the part.
DataCapacity	64u	The amount of memory, in bytes, used for buffered data. Adjust this depending on the needs of the GenTL consumer.

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 Surface scan data and intensity data into Halcon in real-time. Halcon is supported in GoPxL 1.0 and higher. LMI has tested using Halcon 23.05.

> 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 1258.

This section describes how to configure Halcon to acquire data from the Gocator sensor. You should be familiar with the sensor's Surface mode. Before continuing, make sure Halcon is installed.

> Halcon sets the size of the buffer for acquisition only when the driver is initialized. For this reason, if you increase the size of the sensor output after driver initialization, the scan data Halcon receives will be cropped.

> If Halcon is consuming the output of a multi-sensor system or of a tool such as Surface Stitch, that is, where multiple frames are combined, you should set OverrideImageWidth and OverrideImageLength to the maximum expected size. For more information, see *XML Settings File* on the previous page.

Setting Up Halcon

Before using Halcon with a sensor, you must set up Halcon.

To set up Halcon:

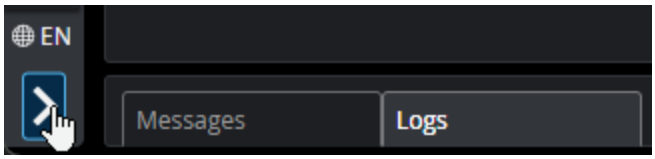
1. Connect a sensor to the PC running Halcon.

2. Connect to the GoPxL IP using a web browser.

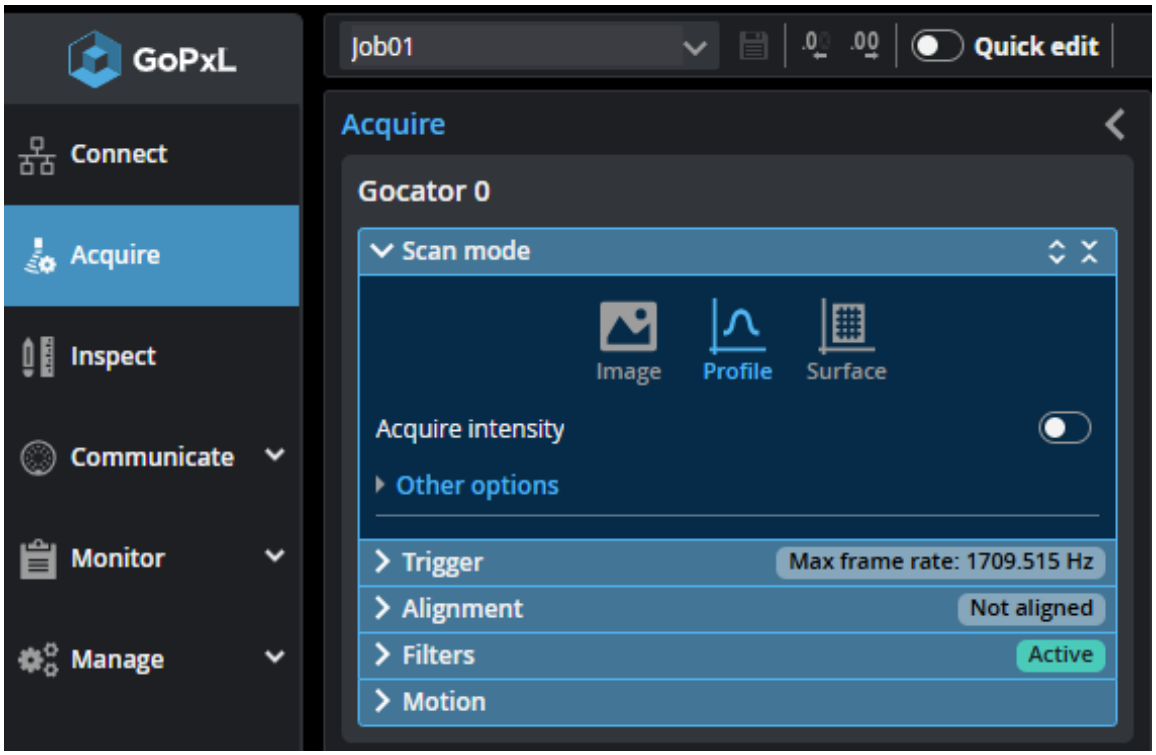
   By default, a sensor's IP is 192.168.1.10. If you have changed the sensor's IP, use that address. If you can't find your sensor, see *GoPxL Discovery Tool* on page 1290.

   For more information on running GoPxL on a PC, see *Running GoPxL on a Windows PC* on page 1188.

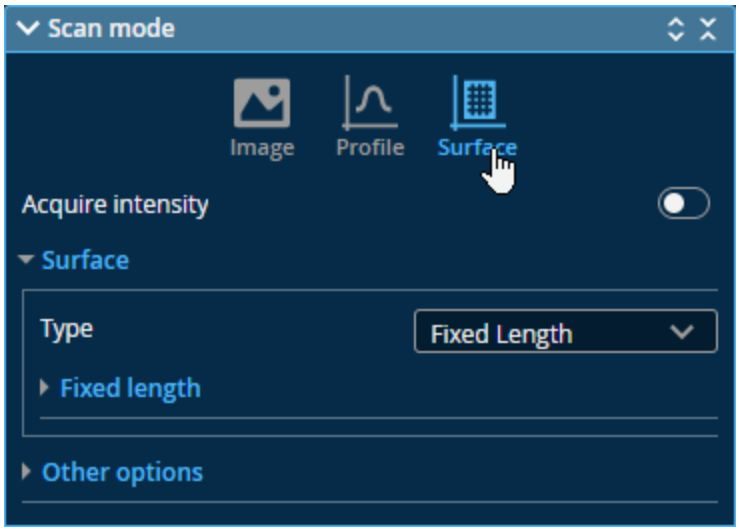
3. In the GoPxL interface, expand the navigation bar on the left side of the interface.



4. Go to the **Acquire** page.

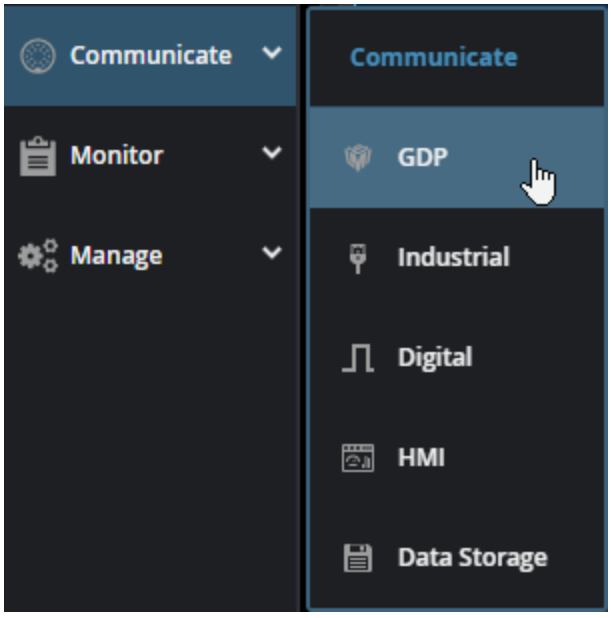


5. In the **Scan mode** section, click the **Surface** icon to switch to Surface mode.



6. (Optional) If you need intensity data, enable **Acquire intensity**.
7. Configure the sensor to produce the desired surface data.

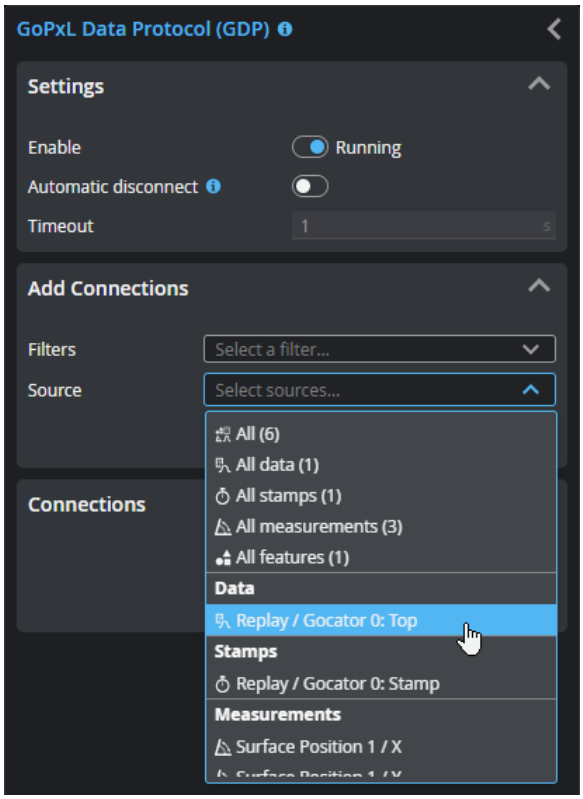
   For more information on configuring sensors, see *Acquire - Alignment and Acquisition* on page 122.
8. Click the **GDP** page icon in the navigation bar.



9. On the **GDP** page, enable the protocol.

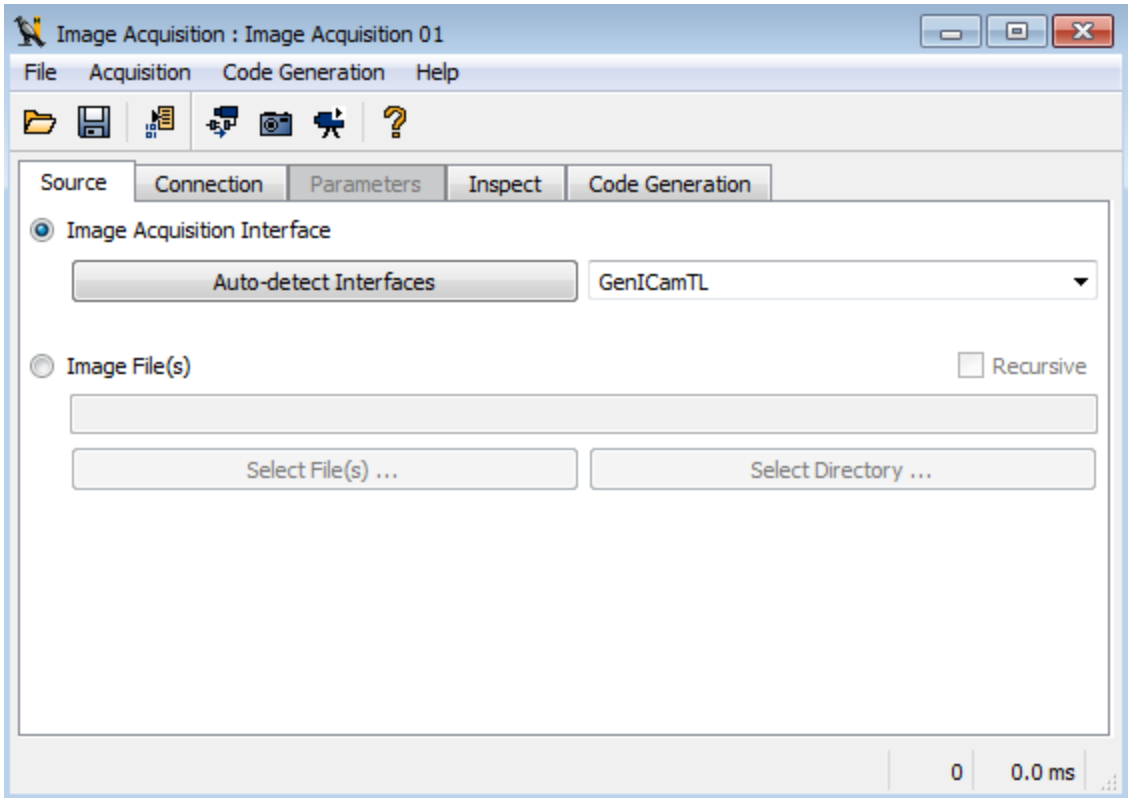


10. In the **Add Connections** section, choose the surface you want to send to Halcon.



For more information on configuring Gocator output, *GoPxL Data Protocol* on page 1023.

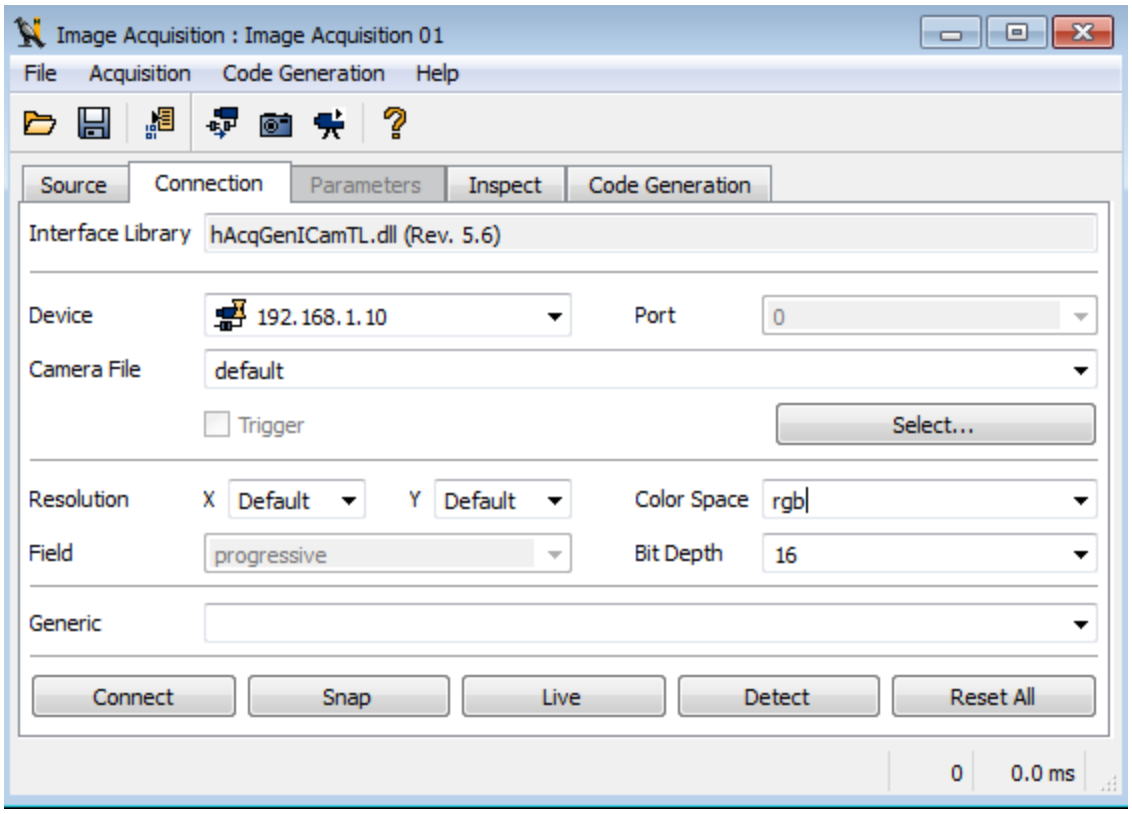
11. Make sure the sensor is running.
12. On the PC, launch Halcon.
13. In Halcon, in the **Assistants** menu, click **Open New Image Acquisition**.
14. In the dialog that opens, in the **Source** tab, check the **Image Acquisition Interface** option and choose `GenICamTL` in the drop-down.



> 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.

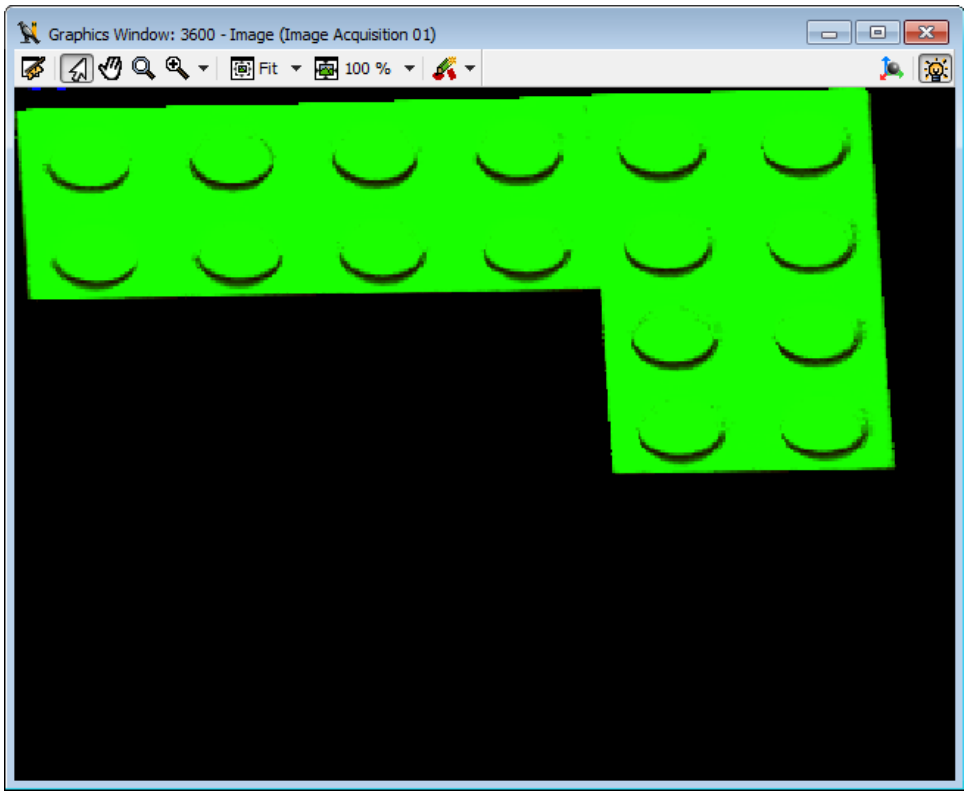
15. Switch to the **Connection** tab.

    If Halcon detects a sensor, the sensor's IP will be listed next to **Device**.



16. In the **Connection** tab, set **Color Space** to `RGB` and **Bit Depth** to `16`.
17. 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

In Halcon, you can use various internal procedures (functions) 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 following table describes each of these procedures.

Halcon Procedures

Procedures	Description
set_ framegrabber_ param	Generic Halcon function to set parameters on the sensor. Can be used to set sensor-specific settings and execute specific commands. For a complete list of settings that can be changed, see below. 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 sensor. **Value**: The parameter value to set on the sensor.

Procedures	Description
	For full details on using this function, see *XmlSetting and XmlCommand Parameters* on page 1277.
get_framegrabber_param	Generic Halcon function to get parameters on the sensor. Used in conjunction with set_framegrabber_param to set the path to the setting, after which the setting's value is retrieved with get_framegrabber_param. In the generic form: `get_framegrabber_param( AcqHandle, 'Name', 'Value')` *Parameters* **AcqHandle** (input): Acquisition handle created by `open_framegrabber`. **Name** (input): The name of the parameter to set on the sensor. **Value** (output): The variable where parameter's value will be stored. For full details on using this function, see *XmlSetting and XmlCommand Parameters* on page 1277.
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 1258. The function accepts the image acquired from `grab_image_async`, and returns the height map, intensity and stamps. ***Parameters*** **Image** (input): RGB Image acquired by using `grab_image_async`. **HeightMap** (output): The height map image. **Intensity** (output): The intensity image. **Stamps** (output): The stamps structure. **Index** (input): The index of the scan to extract. **NumScans** (output): The number of scans in the acquired image. Each output is returned as decimal value. ***Example*** `Go2GenTL_ParseData(Image, HeightMap, Intensity, Stamps, 0, NumScans)`
Go2GenTL_ParseStamps	The stamp information is encoded in the image and can be retrieved using the Go2GenTL_ParseData function. The function Go2GenTL_ParseStamps is used to extract the stamp information from the stamp structure. For details on how the information is packed in the data, see the sections under *GenICam GenTL Driver* on page 1258. The function accepts the stamp structure acquired from the Go2GenTL_ParseData function, and returns the various values encoded in the stamps.

Procedures	Description
	Parameters (Input) **Stamps**: The stamps structure acquired by using Go2GenTL_ParseData. ***Parameters (Output)*** **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 resolution in millimeters. **zOffset**: The Z offset in millimeters. **zResolution**: 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. **NumScans**: The number of scans in the acquired image. **SyncTimestamp**: The PTP time stamp. **MeasurementIds**: Array of every measurement's ID. **MeasurementValues**: Array of every measurement's value. **MeasurementDecisions**: Array of every measurement's decision. **MeasurementDecisionCodes**: Array of every measurement's code. Each output is returned as decimal value. ***Example*** `Go2GenTL_ParseStamps(Stamps, FrameCount, Timestamp,`

Procedures	Description
	`EncoderPosition, EncoderIndex, Inputs, xOffset, xResolution, yOffset, yResolution, zOffset, zResolution, Width, Length, HasIntensity, NumScans, SyncTimestamp, MeasurementIds, MeasurementValues, MeasurementDecisions, MeasurementDecisionCodes)`
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`. *Parameters (Output)* **ConfigFile**: The name of the job file. The file name includes the extension .gpjob. *Example* `Go2GenTL_ConfigFileName (AcqHandle, ConfigFile)`
Go2GenTL_ SetConfigFileNam e	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 .gpjob. *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.

Procedures	Description
	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. **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.

Procedures	Description
	Example `Go2GenTL_SetExposure(AcqHandle, exposure)`

XmlSetting and XmlCommand Parameters

SENSOR SETTINGS

You use the `set_framegrabber_param` and `get_framegrabber_param` functions to set and get the values of advanced parameters on sensors, and additionally modify special settings and send commands at the driver level. (Examples are shown below and in *Sample Program for Various GenTL Settings* on page 1280.)

When you set the second parameter of `set_framegrabber_param` to "XmlSetting", you can use the function to do the following:

- Set the path to a setting you wish to set *or* get.
- Set the value of a setting.
- Set an attribute of a setting.

When you set the second parameter of `set_framegrabber_param` to "XmlCommand", you can send commands and change driver-level settings at the driver level.

Finally, when you retrieve the value of a sensor setting, you set the second parameter of `get_framegrabber_param` to "XmlSettingReturn".

For example, to get the value for a setting, use a series of calls to `set_framegrabber_param` to set the path to the setting, by chunks of up to 32 characters, and retrieve the value into the variable passed in a call to `get_framegrabber_param`. The following gets the value of the Smoothing filter's X window size:

```
set_framegrabber_param(AcqHandle, 'XmlSetting', '/scan/engines')
set_framegrabber_param(AcqHandle, 'XmlSetting', '/LMILaserLineProfiler')
set_framegrabber_param(AcqHandle, 'XmlSetting', '/scanners/scanner-0')
set_framegrabber_param(AcqHandle, 'XmlSetting', '?parameters')
set_framegrabber_param(AcqHandle, 'XmlSetting', '/Smoothing/winX')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', var)
```

To set a value, use `set_framegrabber_param` to set the path as above and add "=<value>". Note that when setting a value with XmlSetting, you must end the set operation with either an empty string or "\n". The following sets the Smoothing filter's X window to 2.5:

```
set_framegrabber_param(AcqHandle, 'XmlSetting', '/scan/engines')
set_framegrabber_param(AcqHandle, 'XmlSetting', '/LMILaserLineProfiler')
set_framegrabber_param(AcqHandle, 'XmlSetting', '/scanners/scanner-0')
set_framegrabber_param(AcqHandle, 'XmlSetting', '?parameters')
set_framegrabber_param(AcqHandle, 'XmlSetting', '/Smoothing/winX=2.5')
```

```
set_framegrabber_param(AcqHandle, 'XmlSetting', '')
```

Using "\n":

```
set_framegrabber_param(AcqHandle, 'XmlSetting', '/Smoothing/winX=2.5\n')
```

For a set or a get, if an index or a parameter is required, use parentheses:

```
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/ResetGocator(1)')
set_framegrabber_param(AcqHandle, 'XmlSetting', '')
```

GENTL DRIVER SETTINGS AND COMMANDS

You can change settings or send commands at the driver level. These options always start with GenTL/, in contrast with the Setup/ node for configuring the sensor.

For example, to stop a sensor programmatically, you can do the following:

```
set_framegrabber_param(AcqHandle, 'XmlCommand', 'GenTL/StopGocator')
set_framegrabber_param(AcqHandle, 'XmlCommand', '')
```

You can also optionally add the "\n" character to the end of the last line of the set operation:

```
set_framegrabber_param(AcqHandle, 'XmlCommand', 'GenTL/StopGocator\n')
```

The following driver-level settings are available:

Driver-level settings

Setting	Comment
"GenTL/Buffers"	Read-Write. Gets/Sets the number of buffers to use in the acquisition pipeline.
"GenTL/Encoder"	Read-only. Gets the current encoder value.
"GenTL/IsResponsive"	Read-only. Gets the responsiveness of the sensor.
"GenTL/HasChanges"	Read-only. Indicates whether the sensor needs a GenTL/Refresh command (see *Driver-level commands* on page 1280).
"GenTL/Version"	Read-only. Returns the version of the GenTL driver.
"GenTL/StartStopControl"	Read-Write. Enables or disables the user defined control. Setting this value to 1 allows the user to start and stop manually the sensor, using the "GenTL/StartGocator" and "GenTL/StopGocator" commands.
"GenTL/State" "GenTL/State(id)"	Read-only. By default, gets the state of the controlling sensor (formerly "main"). If an ID

Setting	Comment
	is provided, the state of the corresponding sensor is returned. Use ID=0 for the controlling. Any other grouped sensors (formerly "buddy") start at ID=1. For a system with three grouped sensors, the index can be from 0 to 2.
"GenTL/BuddyState" "GenTL/BuddyState(id)"	Read-only. By default, gets the state of the first controlled (formerly "buddy") sensor in a sensor group, excluding the controlling sensor (formerly "main"). If an ID is provided, the state of the corresponding sensor is returned. Use ID=0 for the first controlled sensor. For a system with three grouped sensors, the index can be 0 or 1.
"GenTL/BuddyAt" "GenTL/BuddyAt(id)"	Read-only. By default, gets the serial number of the first controlled (formerly "buddy" sensor), excluding the controlling sensor. If an ID is provided, the serial number of the corresponding sensor is returned. Use ID=0 for the first controlled sensor. For a system with three grouped sensors, the index can be 0 or 1.
"GenTL/BuddyCount"	Read-only. Returns the number of controlled sensors (formerly "buddy" sensors), excluding the sensor that is controlling the other sensors. For a system with three grouped sensors, the function will return 2.
"GenTL/FlushOldBuffers"	Flushes the GenTL pipeline buffers. Can be used to ensure only a new surface captured, instead of getting the previously captured surfaces.
"GenTL/Main"	Read-only. Returns the serial number of the main sensor.
"GenTL/AlignResult"	Read-only. Returns the alignment result of the last "GenTL/Align" command (see *Driver-level commands* on the next page).
"GenTL/LastResult"	Read-only. Returns the last result for any GenTL XmlCommand.
"GenTL/TransformX(id)" "GenTL/TransformY(id)" "GenTL/TransformZ(id)" "GenTL/TransformXAngle(id)" "GenTL/TransformYAngle(id)" "GenTL/TransformZAngle(id)"	Read-Write. Gets or sets the transformation value for a given transform of the specified sensor ID.
"GenTL/TransformEncoderResolution"	Read-Write. Gets/Sets the encoder resolution for a sensor.
"GenTL/DataSourceIds[ids]"	Read-Write. Gets/Sets the indices of one or more surfaces to transfer. By default, this is set to the following, which returns only the first surface: 'GenTL/DataSourceIds=[0]\n' To specify more than one, use the following syntax. which sets the data source to the first, third, and fourth surfaces:

Setting	Comment
	'GenTL/DataSourceIds=[0,2,3]\n'

The following driver-level commands are available:

Driver-level commands

Setting	Comment
"GenTL/FlushGocatorBuffers"	Flushes the Gocator pipeline buffers. Can be used to ensure only a new surface captured, instead of getting the previously captured surfaces.
"GenTL/StartGocator"	In an enabled StartStopControl mode, starts the Gocator programmatically.
"GenTL/StopGocator"	In an enabled StartStopControl mode, stops the Gocator programmatically.
"GenTL/ResetGocator" "GenTL/ResetGocator(wait)"	By default, this call resets the Gocator without waiting. If the wait parameter is set to 1, the command will wait for the Gocator to be reset, and reconnects to it automatically if possible.
"GenTL/Align(timeout)"	Performs an alignment on the sensor, waiting for the given timeout in seconds.
"GenTL/ClearAlignment"	Clears the current alignment on the sensor.
"GenTL/ResetEncoder"	Resets the encoder. Note that this only works if the encoder is directly connected to the sensor, and may not work on encoders connected to a Master.
"GenTL/Reconnect"	Attempts to reconnect to the sensor, after it was disconnected or reset.
"GenTL/Refresh"	Updates the client state to reflect any changes that have occurred in the sensor network.
"GenTL/ClearData"	Clears the data on message queue of the Gocator.
"GenTL/SoftwareTrigger"	Triggers the Gocator for an acquisition, if it is setup to use Software Triggering.

Sample Program for Various GenTL Settings

```
* Connect to the sensor. This code is auto generated by the Image Acquisition dialog box.
User can manually override the directory path by editing the line below.
* Set image type to 'rgb' if the format is 16-bit RGB. 'gray' if the format is 16-bit mono
open_framegrabber ('GenICamTL', 0, 0, 0, 0, 0, 0, 'progressive', 16, 'rgb', -
1, 'false', 'default', '127.0.0.1:3190', 0, -1, AcqHandle)

* Example calls for configuring the sensor
* Go2GenTL_ResampleMode (AcqHandle, ResampleMode)
* Go2GenTL_ImageSize (AcqHandle, ImageWidth, ImageHeight)
* Go2GenTL_Encoder (AcqHandle, EncoderValue)
Go2GenTL_Exposure (AcqHandle, currentExposure)
Go2GenTL_SetExposure (AcqHandle, currentExposure + 20)
set_framegrabber_param (AcqHandle, 'grab_timeout', 100000)
* set_framegrabber_param (AcqHandle, 'XmlCommand', 'GenTL/ClearData\n')

* Note: Configuration name has to be less than 16 characters
* Go2GenTL_SetConfigFileName (AcqHandle, 'newExposure.gpjob')
```

```
* Go2GenTL_ConfigFileName (AcqHandle, ConfigFile1)

* Other examples
* *Get fixed length
*  set_framegrabber_param(AcqHandle, 'XmlSetting', 'scan/engines')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/LMILaserLineProfiler')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/scanners')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/scanner-0')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '?parameters')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/scanModeSettings')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/surface')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/fixedLength')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/length')
*  get_framegrabber_param(AcqHandle, 'XmlSettingReturn', length)
*  *Set fixed length</c>
*  set_framegrabber_param(AcqHandle, 'XmlSetting', 'scan/engines')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/LMILaserLineProfiler')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/scanners')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/scanner-0')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '?parameters')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/scanModeSettings')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/surface')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/fixedLength')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '/length')
*  set_framegrabber_param(AcqHandle, 'XmlSetting', '=50.00\n')

*  set_framegrabber_param (AcqHandle, 'XmlSetting', 'GenTL/DataSourceIds=[0,1,2]\n')

* Version
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/Version')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', version)

* Transform
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/TransformX(1)')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', xOffset)
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/TransformX(1)=0.01\n')
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/TransformX(1)')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', xOffset)

set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/TransformYAngle(1)')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', yAngle)
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/TransformYAngle(1)=0.01\n')
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/TransformYAngle(1)')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', yAngle)

* Encoder Resolution
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/TransformEncoderResolution')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', encoderRes)
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/TransformEncoder')
set_framegrabber_param(AcqHandle, 'XmlSetting', 'Resolution=0.002\n')
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/TransformEncoderResolution')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', encoderRes)

* Buddy List
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/BuddyCount')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', buddyCount)
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/BuddyAt(0)')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', buddy0)
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/BuddyAt(1)')
```

```
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', buddy1)

* State
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/Refresh\n')

set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/State')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', state0)
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/State(1)')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', state1)

set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/BuddyState')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', bstate0)
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/BuddyState(1)')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', bstate1)

* Alignment
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/ClearAlignment')
set_framegrabber_param(AcqHandle, 'XmlSetting', '')

set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/AlignResult')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', alignResult)

set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/Align(5)')
set_framegrabber_param(AcqHandle, 'XmlSetting', '')

set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/AlignResult')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', alignResult)

* Reset
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/ResetGocator(1)\n')
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/LastResult')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', lastResult)

* Responsiveness
set_framegrabber_param(AcqHandle, 'XmlSetting', 'GenTL/')
set_framegrabber_param(AcqHandle, 'XmlSetting', 'IsResponsive')
get_framegrabber_param(AcqHandle, 'XmlSettingReturn', isResponsive)

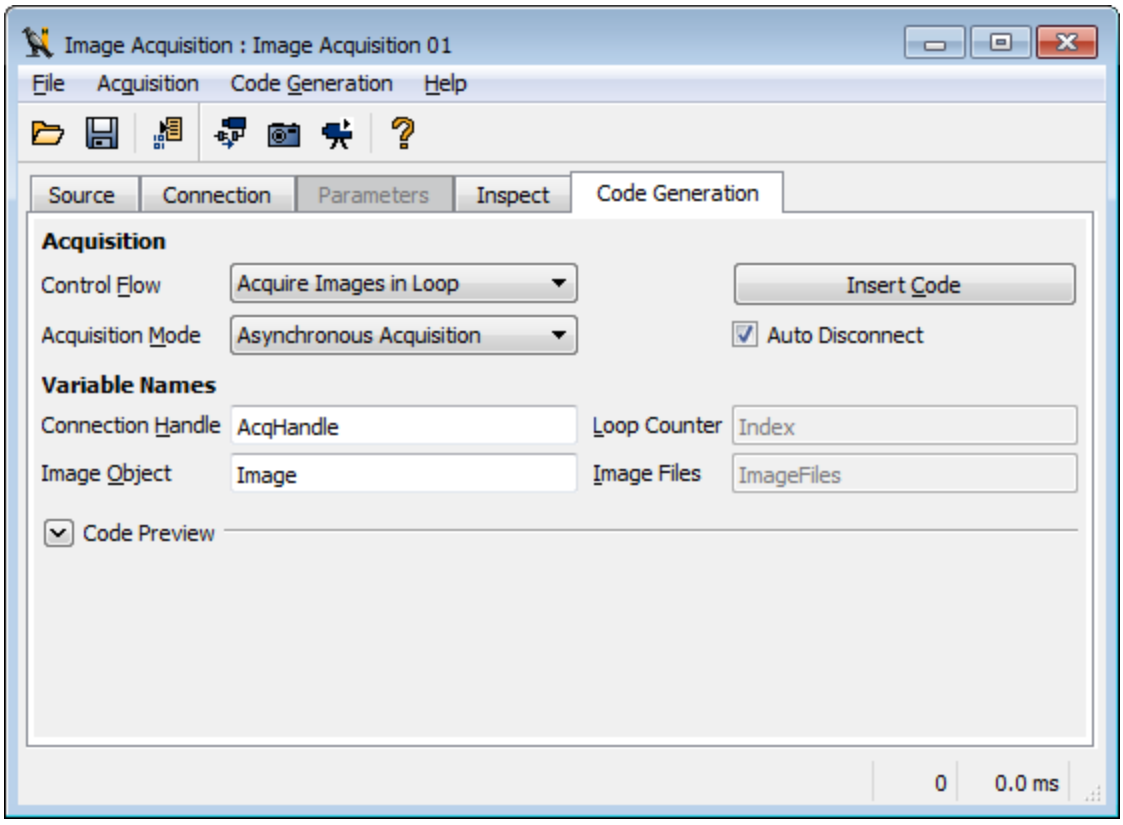

close_framegrabber (AcqHandle)
```

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.

GoAIK Cognex VisionPro Driver

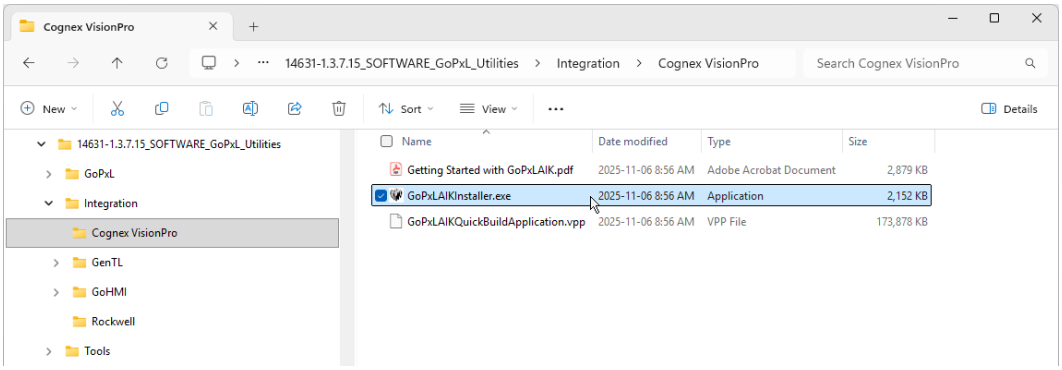
You use LMI's GoAIK driver to connect to Gocator sensors from Cognex VisionPro. The GoAIK driver only supports Surface mode with **Uniform Spacing** enabled. You must enable the required output in the Ethernet panel in the Output page. Check **Acquire Intensity** in the **Scan mode** panel on the **Acquire** page if you need intensity data.

The GoAIK driver packs the output, intensity, and stamps (that is, time stamp, encoder index, and so on) into a 16-bit grey scale image. You can select the format in the GoAIK.xml setting file.

The width and height of the 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.

You can find the GoAIK driver installer, as well as an example tool for Cognex VisionPro and Ether-Inspect and Edit Tool Block, in the Utilities package (14631-x.x.x.x_GO_Utilities.exe), available on LMI's website (https://lmi3d.com/product-downloads/). Select Gocator as the brand, Software as the category, and the desired GoPxL release.

Once you have downloaded and extracted the Utilities package, you can find the installer and example tool in the `Integration > Cognex VisionPro` folder.



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

Rows	Details
	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.
(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. Refer to the stamps on how to retrieve the offset and resolution values.

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 GoAIK 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)
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)

Stamp Index	Column Position	Details
13	52..55	Height map length (in pixels)
14	56..59	Specify if intensity is enabled or not

XML Settings File

The settings file, GoAIK.xml, resides in the same directory as the GoAIK driver.

Element	Type	Description
DataCapacity	64u	The amount of memory, in bytes, used for buffered data. By default, 500000.
Sensors/Sensor/LockIpAddress		The IP address of a sensor you want to connect to. You can add multiple Sensor elements (containing a LockIpAddress element) under the Sensors element. The default value of LockIpAddress is 192.168.1.10.

Interfacing with VisionPro and Ether-Inspect 11

The following sections assume that a Gocator sensor is already installed and accessible over a network.

Setting Up VisionPro

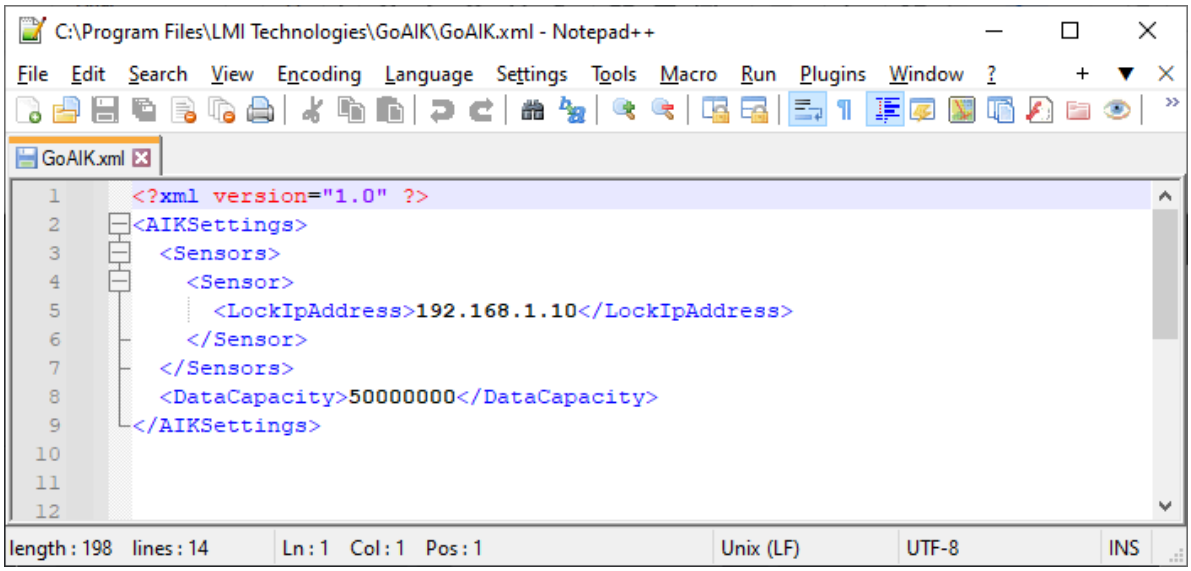
After you have installed the GoAIK driver, you need to configure the GoAIK.xml settings file and run Cognex VisionPro or Ether-Inspect 11.

To set up VisionPro or Ether-Inspect 11

1. Go to the folder in which you installed the GoAIK driver.

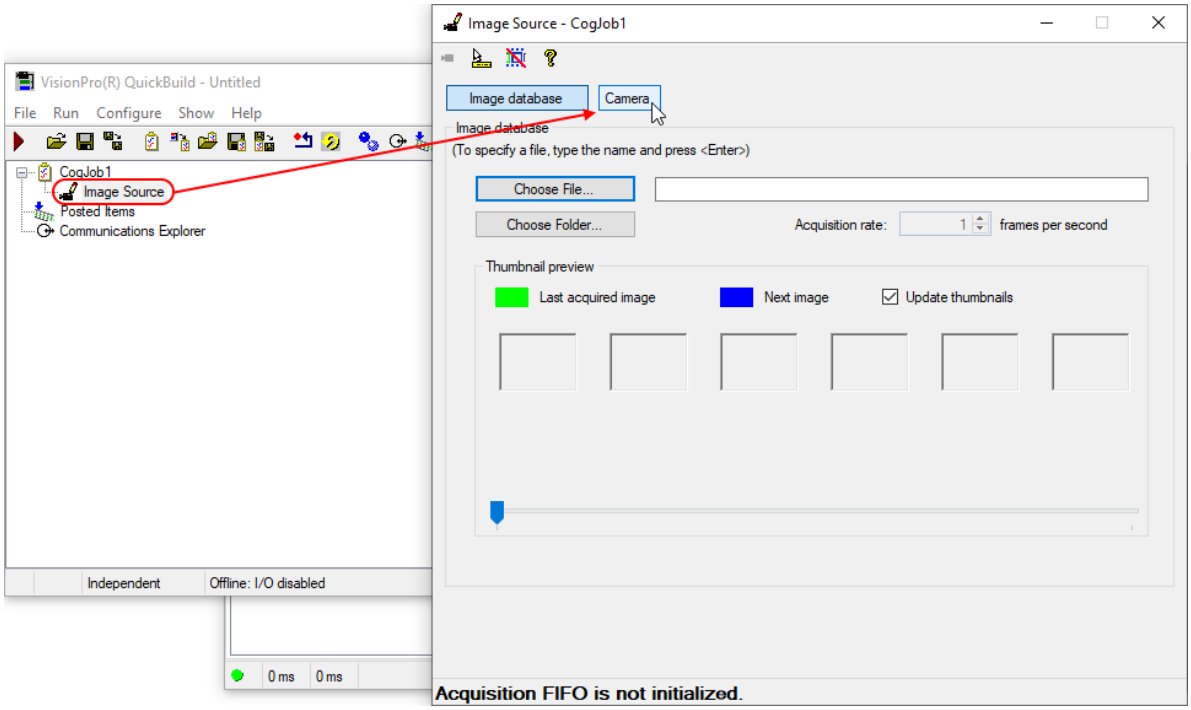
   By default, the installation location is `C:\Program Files\LMI Technologies\GoAIK`. For information on where to get the installer, see *GoAIK Cognex VisionPro Driver* on page 1284.

2. If necessary, edit the value of `LockIpAddress` element in GoAIK.xml to specify the IP address of the sensor you want to connect to.

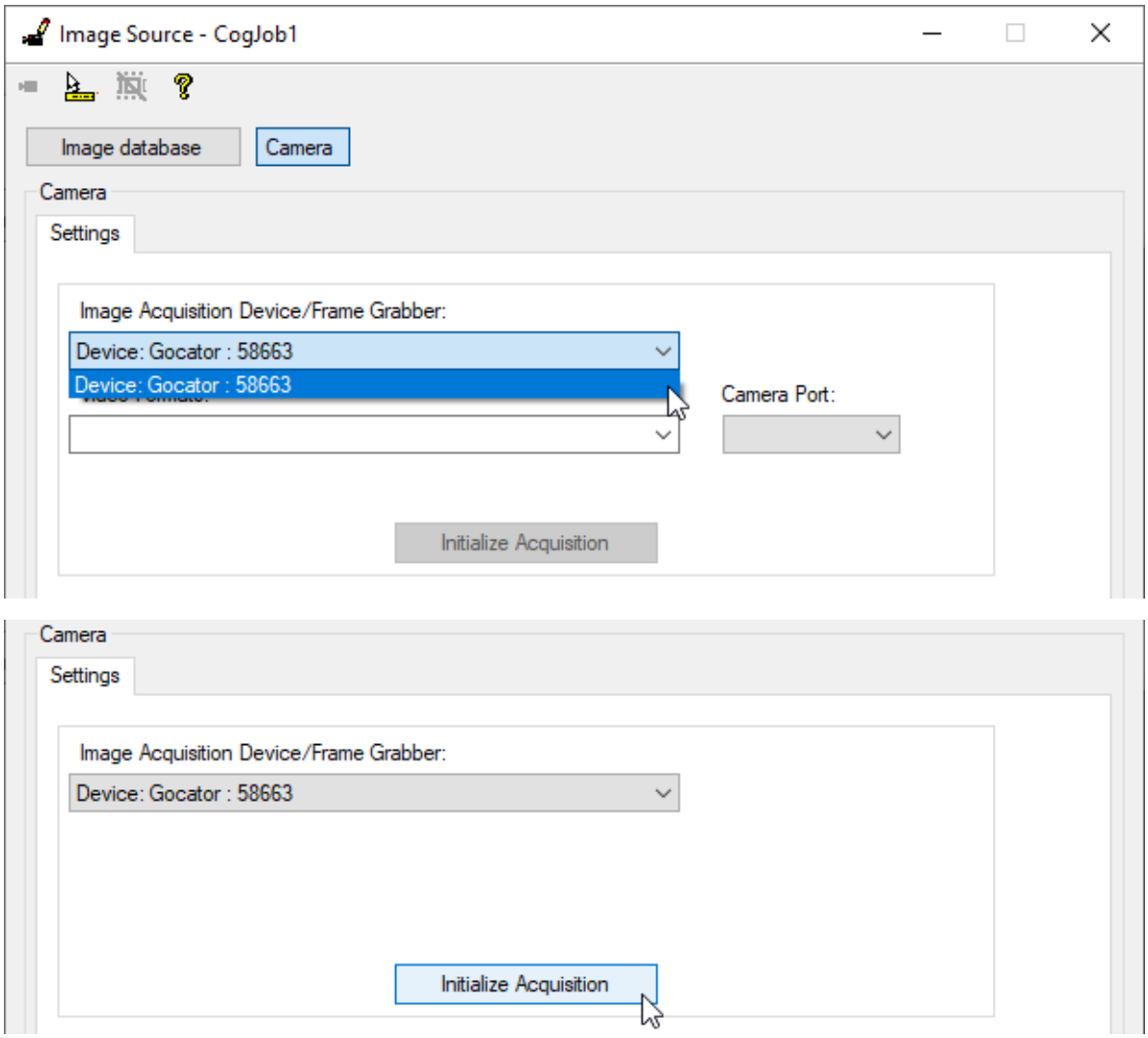


The default value of `LockIpAddress` is 192.168.1.10. (This is the default address of Gocator sensors.)

3. Open Cognex VisionPro or Ether-Inspect.
4. In VisionPro, double-click **Image Source** and in the dialog that opens, click **Camera**.



5. In the **Settings** tab, select the sensor in **Image Acquisition Device/Frame Grabber**, and then click **Initialize Acquisition**.



Additional tabs display. Close the Image Source dialog.

6. If you have previously configured the sensor and saved the configuration as a job, go to step 13.
7. Connect to the sensor and set the scan mode to Surface.
8. (Optional) If you need intensity data in your output, on the **Acquire** page, enable **Acquire intensity**.
9. Configure the sensor's Surface acquisition in the **Acquire** page in the GoPxL interface.

   For more information on setting up acquisition, see *Acquire - Alignment and Acquisition* on page 122.
10. On the **Communicate** > **GDP** page, enable GDP.
11. Add connections for the measurements, stamps, or geometric features you want to send over GDP in the Add Connections panel.

    For information on enabling and configuring GoPxL Data Protocol, see *GoPxL Data Protocol* on page 1023.
12. Go to step 14.
13. (Optional) If you have configured the sensor and saved the configuration to a job on the sensor, you can use the Job command in VisionPro, to switch to that job.

For more information on the available commands, see *Commands* below.

14. In the GoPxL interface, press Start, and in VisionPro, observe the Surface data that the sensor outputs.

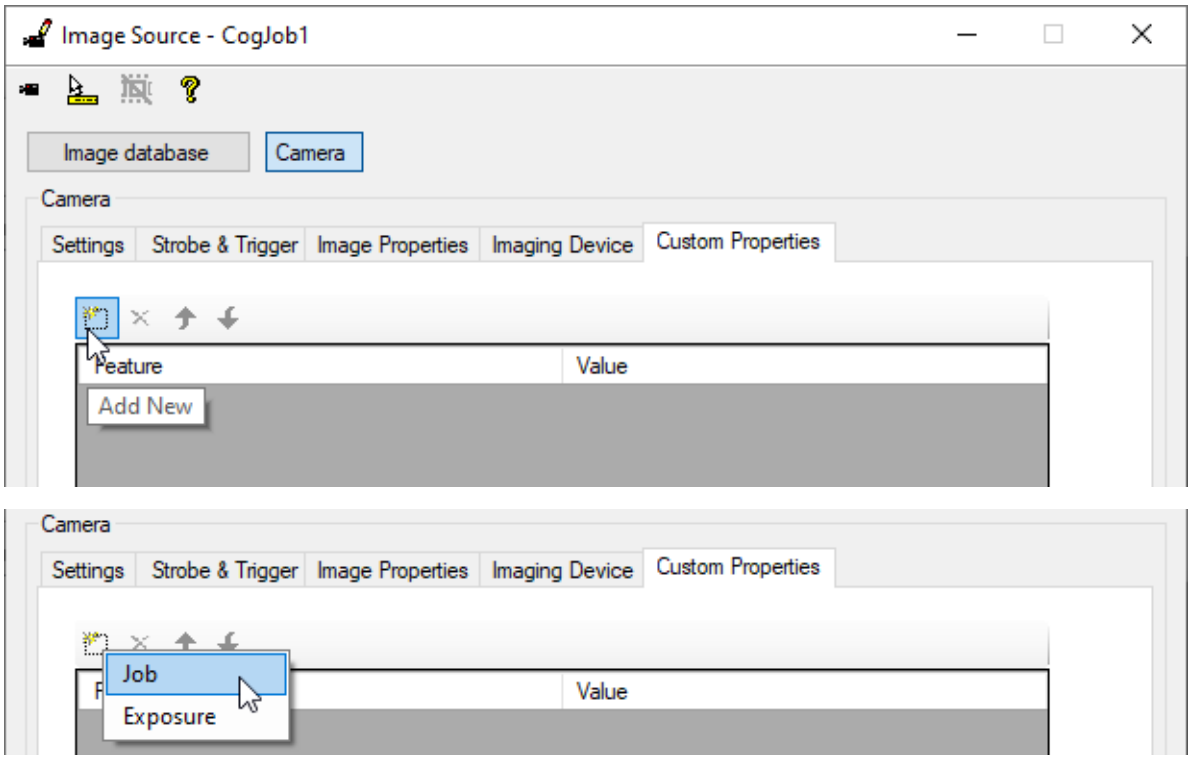
Commands

The GoAIK driver provides two commands:

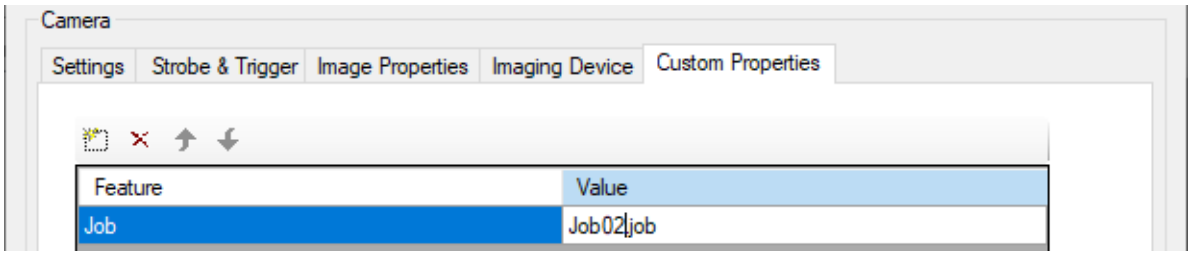
- job: Lets you set the job on the sensor
- exposure: Lets you set the exposure on the sensor.

To add a command

1. Open Image Source, and in the **Custom Properties** tab, click **Add New** and select a command.



2. Edit the value of the command.



Note that when you add a command and set its value, the command is sent as soon as you move the focus from the value field

Utilities

The following sections describe some of the utilities 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 1257.

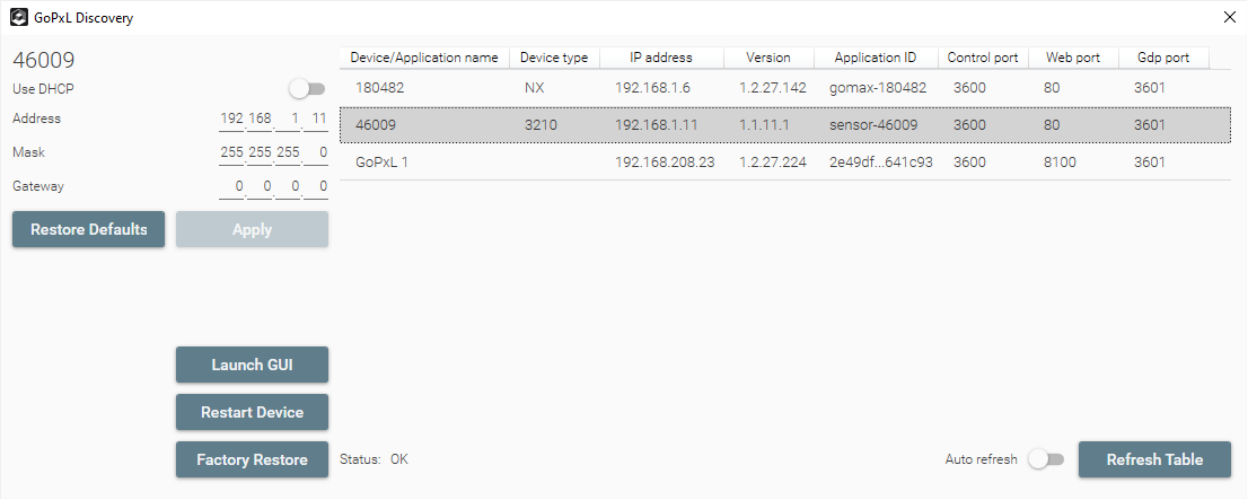
You can get the tools in the utilities package (`14631-1.2.30.5_GoPxL_Utilities.exe`) from the downloads area of the LMI Technologies website: https://lmi3d.com/product-downloads/. Run the installer, and provide a location where the content of the package will be unzipped.

The following tools are available:

- Alignment CAD (G2): CAD files for pyramid plate and double-sided pyramid alignment targets. For use with Surface Align Wide and Surface Align Ring on multi-sensor G2 systems, respectively. You should only use these if your application requires a high degree of accuracy.
- Discovery tool: Used to find devices on a network. See *GoPxL Discovery Tool* below.
- Pattern Editor: Used to edit patterns created in the pattern matching tool. For more information, see *Pattern Editor* on page 1307.
- Replay Utils tool: Lets you convert between .gprec files and .srf/.sur/.pcd/.pro files. It also lets you convert .gprec files to CSV files. For more information, see *Replay Utility Tool* on page 1294.
- Track Editor: Used with the Surface Track tool. For more information, see *Surface Track* on page 980.
- Zebra Aurora Design Assistant; for more information, see the instructions in `Integration\Zebra\Design Assistant\GoPxL Zebra Design Assistant Integration.pdf` in the Utilities package. Note that G3 sensors do not support Zebra Assistant.

GoPxL Discovery Tool

The GoPxL Discovery tool lets you perform various operations with sensors running GoPxL, PC instances of GoPxL, and GoMax units. The tool also lets you see sensors running Gocator Classic firmware, and launch their user interface in a browser, for example, to make it easier to upgrade a sensor from Classic to GoPxL.



You can do the following with a sensor running GoPxL:

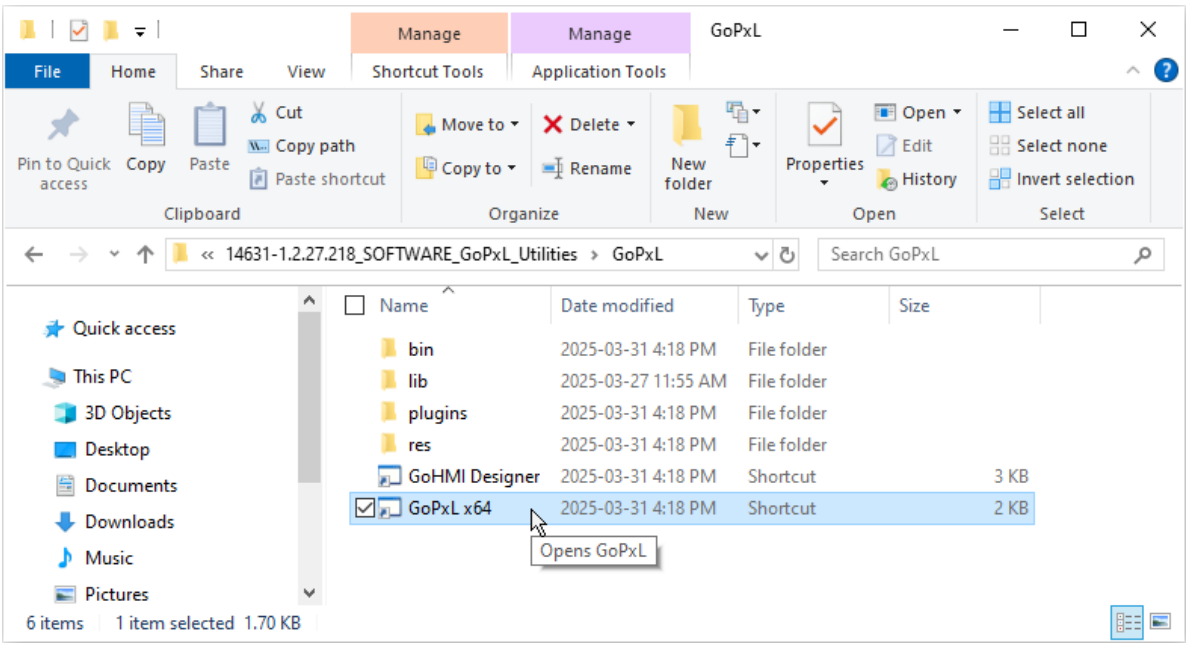
- Find a sensor if you have forgotten its IP address.
- View port information.
- Set the network configuration (IP address, mask, and so on) of a sensor, and also restore these values to the default.
- Launch the GUI of a sensor in your browser.
- Restart the sensor.
- Perform a factory restore on a sensor.

You can do the following with a PC instance of GoPxL or a GoMax:

- Find its IP address.
- View port information.
- Launch the GoPxL of the PC instance or GoMax in your browser.

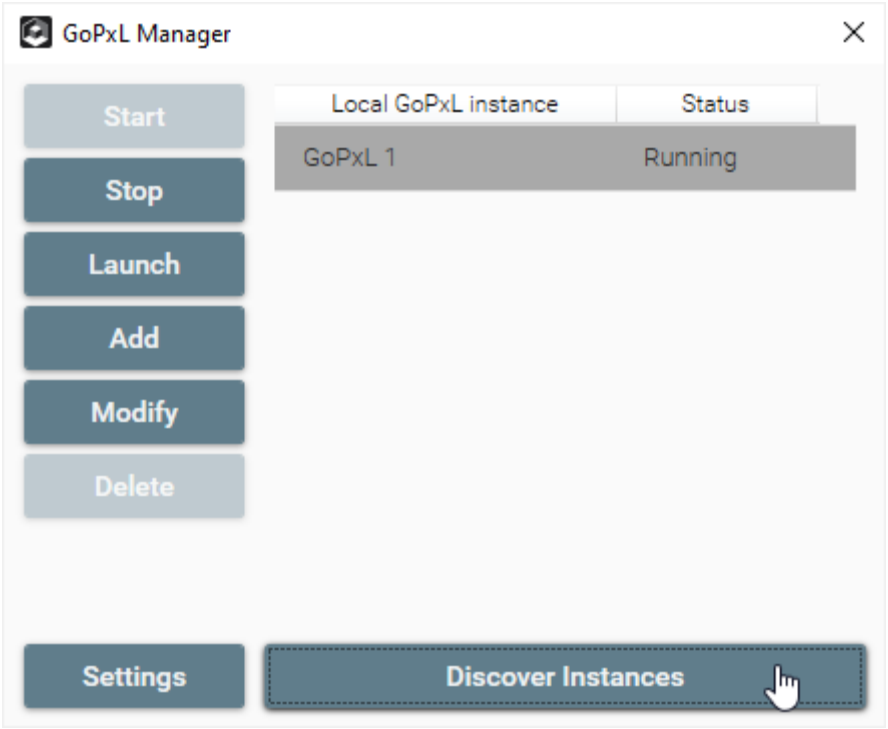
To run the GoPxL Discovery tool

1. If it isn't already running, start the Windows version of GoPxL.

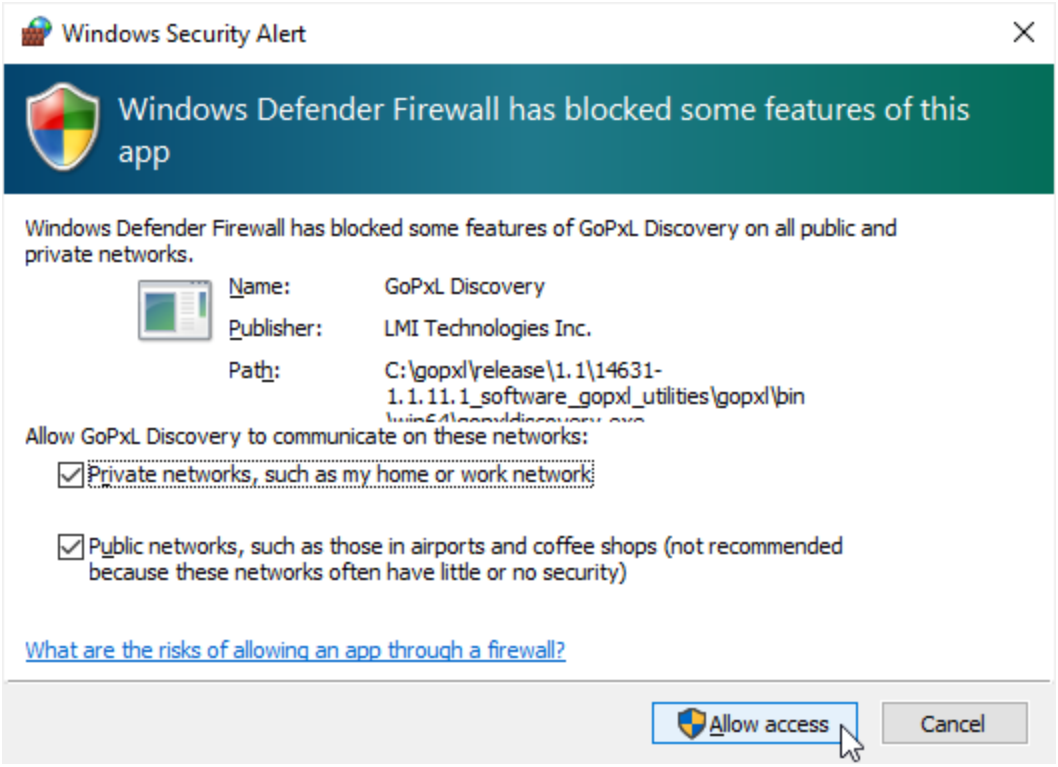


GoPxL Manager opens.

2. In the manager tool, click **Discover Instances**.



Your firewall may ask if you want to allow the GoPxL Service Application and Python to communicate over your network. To avoid issues, LMI recommends allowing communication over both Public and Private networks. Because the notifications appear one on top of the other, make sure to check the appropriate boxes and carefully click **Allow access** for both alerts.



The GoPxL Discovery tool opens.

When the GoPxL Discovery tool opens, it lists the sensors running GoPxL and the PC instances of GoPxL that are currently visible on the network. Sensors are identified in the **Device/Application Name** column by their serial number, and in the **Device Type** column by their model. If a sensor is running through a PC instance of GoPxL, its status (at the bottom of the tool) is "Device is remotely connected", and you can't perform any operations on it from the tool. Otherwise, the sensor's status is "OK".

GoPxL instances are identified in the **Device/Application Name** column by their name.

The tool provides important network-related information, namely, IP addresses and Web, Control, and GDP ports. The Web port is the port used to connect to the GoPxL interface in a browser. For sensors, the default port is 80; you don't need to specify this in the URL when connecting directly to a sensor. The Control port and the GDP ports are used by the SDK and the REST API. (For an introduction to these, see *GoPxL SDK and REST API* on page 1234.)

If a sensor or GoPxL instance is missing, click the **Refresh Table** button. Optionally, you can also enable the **Auto refresh** toggle; the table refreshes .

To perform operations on a sensor or a GoPxL instance, click its row in the tool and then perform the operation.

Replay Utility Tool

The Replay Utility tool (ReplayUtils.exe) is a console application that lets you do the following conversions with scan data. You can run it from a command line using Windows Command Prompt or Windows PowerShell.

- .gprec <=> .srf/.sur/.pcd (valid for surface recording data)
- .gprec <=> .pro (valid for profile recording data)
- .gprec <=> .csv (valid for profile/surface recording data)
- folder of .gprec files <=> .gprec file (valid for surface/profile/image/video data)

Intensity information is included when converting to and from .pcd and .srf files. Intensity is not supported when converting to and from the .sur format.

You can get the tool in the utilities package (`14631-x.x.x.x_GoPxL_Utilities.exe`) from the downloads area of the LMI Technologies website: https://lmi3d.com/product-downloads/.

After downloading the utilities package, unzip the file. Then, open a command prompt or PowerShell instance in the folder containing the utility. To open a command line quickly, in File Explorer go to the folder containing the tool, hold Shift, right-click in empty area, and choose "Open command window here" or "Open PowerShell window here."

Run ReplayUtils.exe (under Tools\ReplayUtils). You must either copy the .gprec file into the folder containing the utility, or include its entire path in the call to the utility, for example, `ReplayUtils.exe -i \path\to\file\myData.gprec -f 0 -o myOutput.sur`.

The tool provides the following parameters:

Parameters

Name	Description
--help	Show the help.
-i --input	Specify the file name of the input file or the input directory. Required.
-f --frame	Specify the *zero-based index* of the frame to be exported in the recording data. For example, to export the first frame in the scan data, use "-i 0"; for the second frame, use "-i 1". (Default is 0.)
-w --width	Specify the width of the exported surface. (Only valid for importing .pcd files.)
-h --height	Specify the height of the exported surface. (Only valid for importing .pcd files.)
-sxz --swapXZ	Whether to swap the X and Z coordinates. (Only valid for importing .pcd files.)
-syz	Whether to swap the Y and Z coordinates. (Only valid for importing .pcd files.)

Name	Description
--swapYZ	
-z --zoom	Specify the zooming factor of the exported surface. (Only valid for importing .pcd files.)
-r --remove	Remove the specific point. (Only valid for importing .pcd files.)
-a --all	Export all the frames if the import is recording data.
-o --output	Specify the file name of the output file. If the imported file isn't recording data and the exported file name isn't specified, the utility uses the imported file name as the exported file name.

CSV File Format

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.



The "Part" section is unrelated to part detection. It represents Surface data when **Enable uniform spacing** is checked.

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

Field	Description
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.

Device and firmware information (the ID, Model, and Version fields) is not currently output.

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
Exposure Min	Dynamic exposure min
Exposure Max	Dynamic exposure max
FOV X	Active area X

Field	Description
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

Recording filters are not currently supported.

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

Field	Description
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 the sub-header is a resampled X position Each column in the 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)

Field	Description
(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 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:

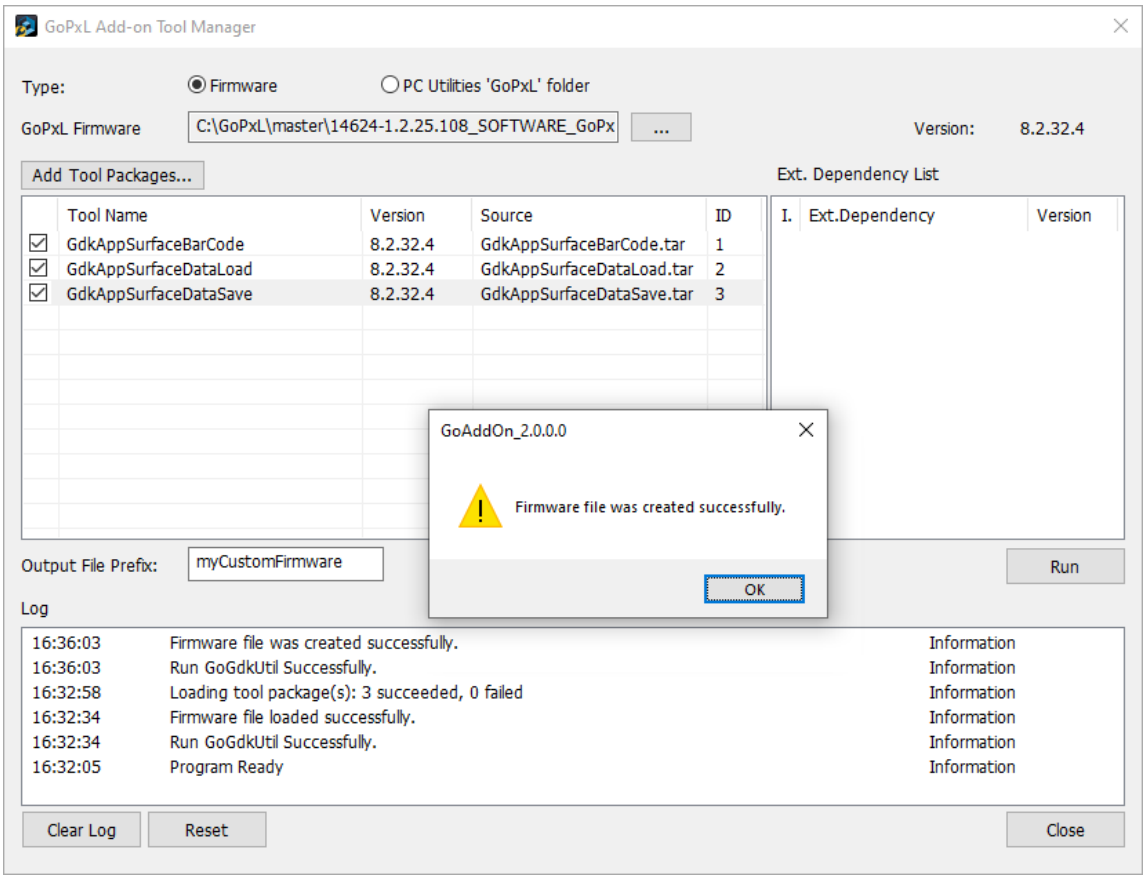
```
p0x, p0y, p0z, p1x, p1y, p1z, ..., pnx, pny, pnz
p(n+1)x, p(n+1)y, p(n+1)z, ...
...
```

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.

GoPxL Add-on Tool Manager

The GoPxL Add-on Tool Manager lets you quickly and easily add LMI-provided beta tools to GoPxL. You can also use the Manager add Python GDK tools to a firmware. After you add tools to a firmware file, upload the firmware to a compatible sensor; for more information, see *Software Upgrade* on page 1184.



The tool manager is available in the 14632-x.x.x.x_GoPxL_Tools_AddOn_Beta.zip package, available on the LMI Product Downloads page (https://lmi3d.com/product-downloads/), under the *Beta Software Releases* section.

To get the package:

1. Go to https://lmi3d.com/product-downloads/ and log in to your account.
2. Select Gocator in the brand drop-down.
3. Choose the package that corresponds to the firmware 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 GoPxL Add-on Tool Manager is as follows:

1. Load a .dat GoPxL firmware file (or point the tool manager to the "GoPxL" folder in an unzipped Utilities package).
2. Load one or more beta tool packages (in .tar archives).
3. Enable the tools you want to add to the loaded firmware (in the checkbox column).
4. Run the tool manager on the firmware and tool packages.

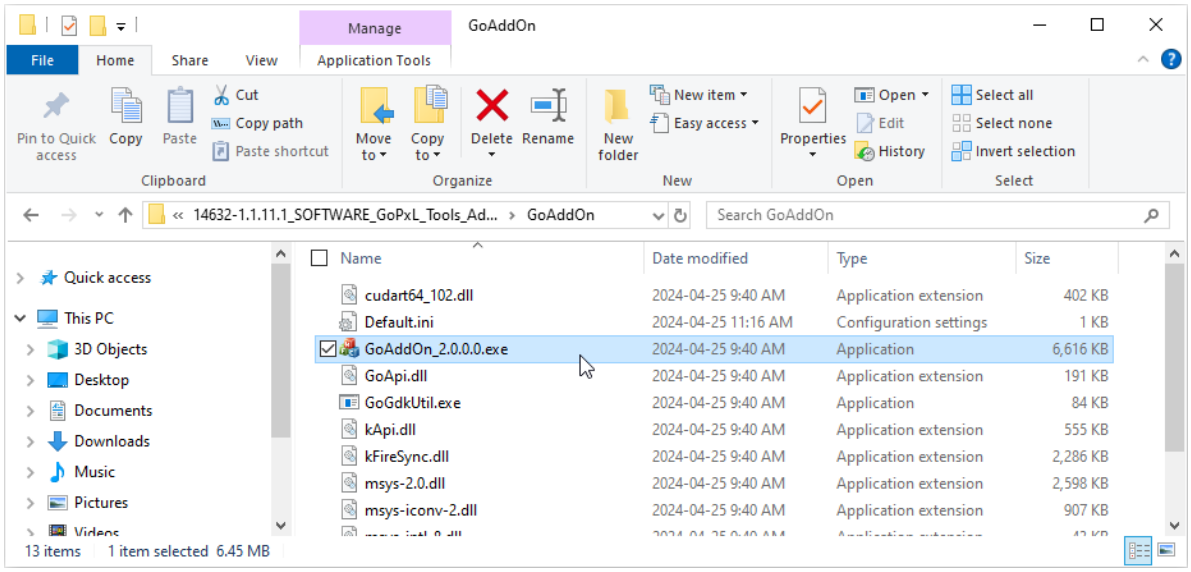
The result is a modified firmware file or GoPxL for PC that contains the tools you selected.

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 add beta tools to a firmware or to GoPxL for PC:

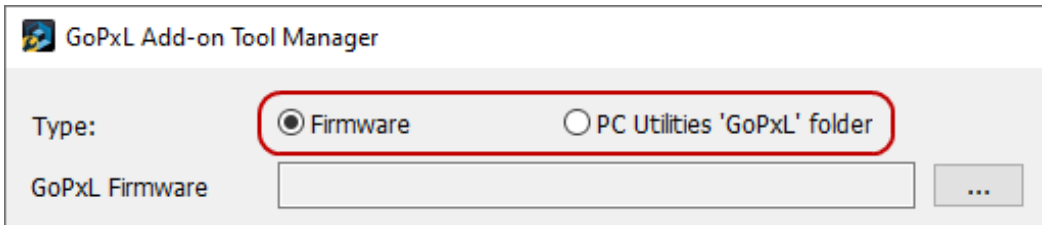
1. If you haven't already done so, download and unzip the package containing the GoPxL Add-on Tool Manager to a convenient location on your computer.
2. Launch the tool manager (GoAddOn_x.x.x.x.exe) from the \GoAddOn subfolder.



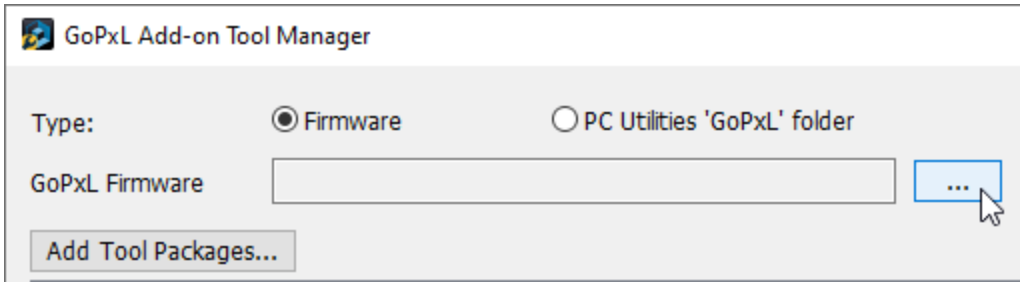
3. In the tool manager, set the type.

   To create a custom firmware you will upload to a sensor, set the type to **Firmware**.

   To create a custom Windows version of GoPxL, set the type to **Accelerator/Emu**.

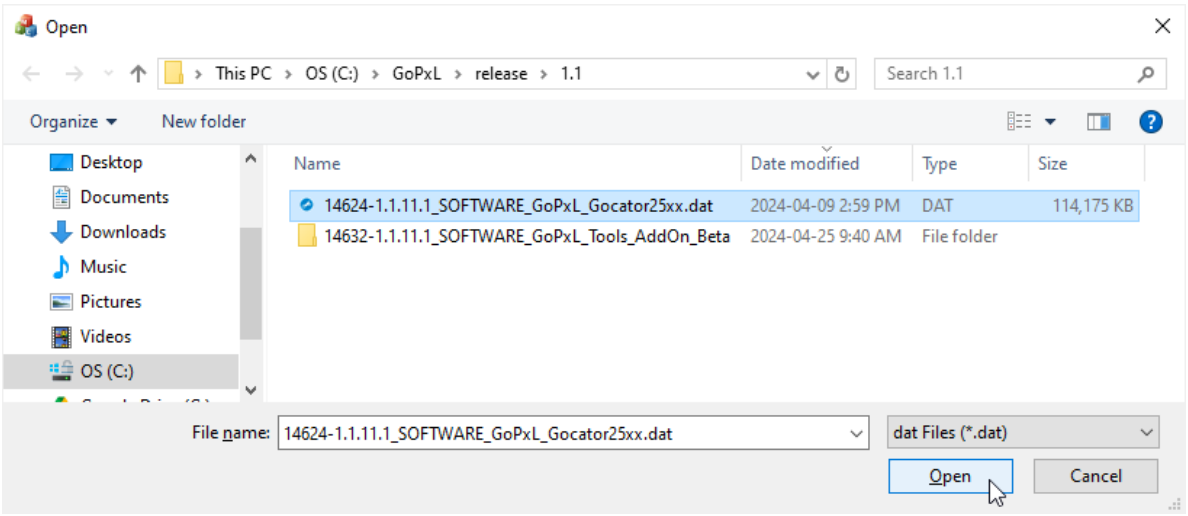


4. Below the type, click the ... button to choose the firmware or GoPxL folder.

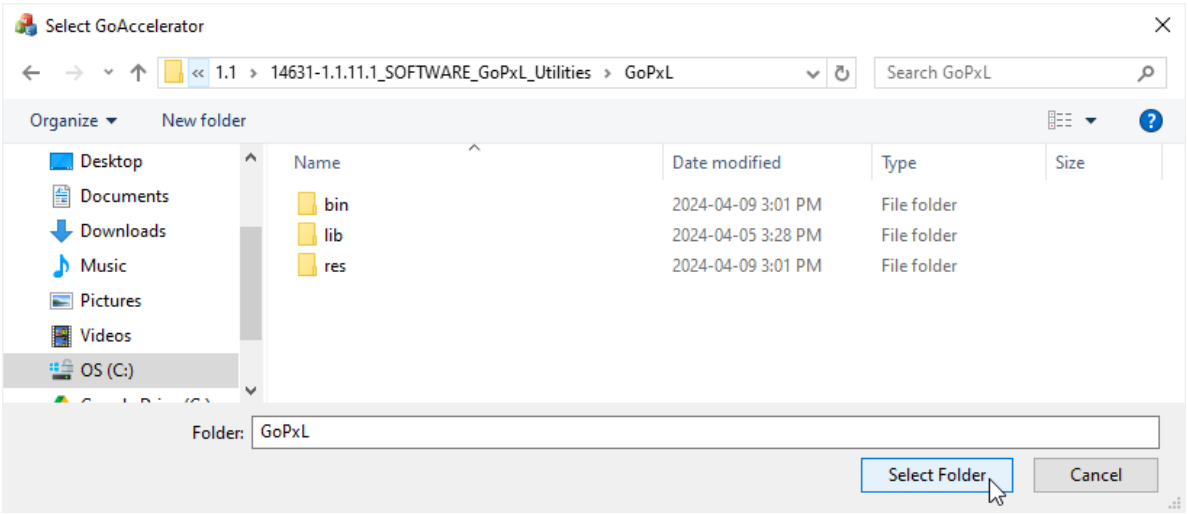


5. In the Open dialog that displays, navigate to the location of the .dat firmware file or the folder you want to use.

   Choosing a firmware:

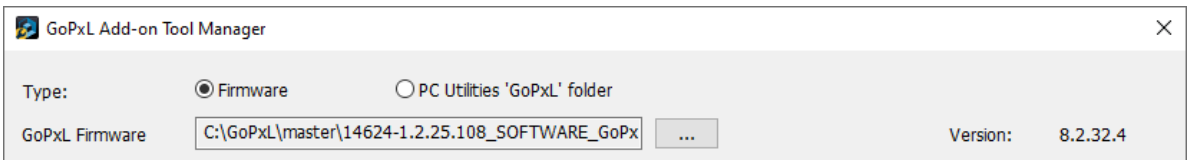


   Choosing a folder containing GoPxL for Windows:



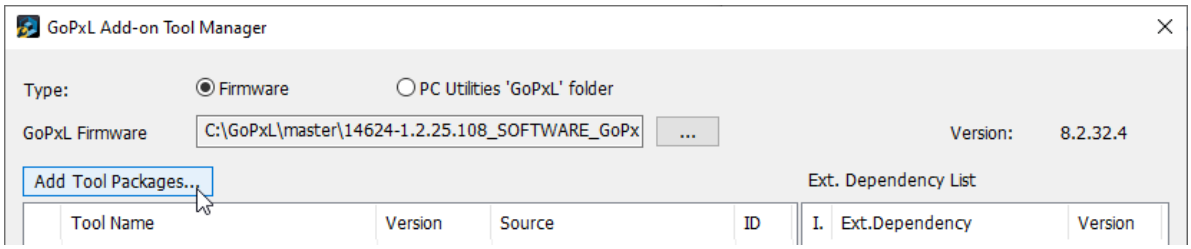
   Note that you choose the `GoPxL` folder.

   The tool manager loads GoPxL.

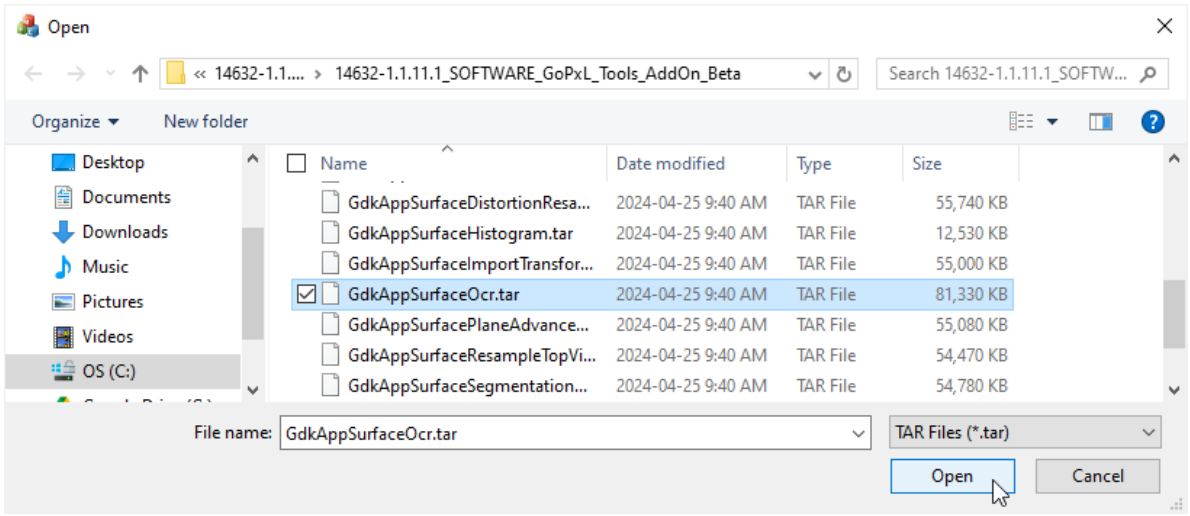


   If the version of GoPxL was previously modified using the tool manager, the tools will be listed in the tool manager. For information on removing tools, see *Removing Beta Tools* on page 1305.

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**.

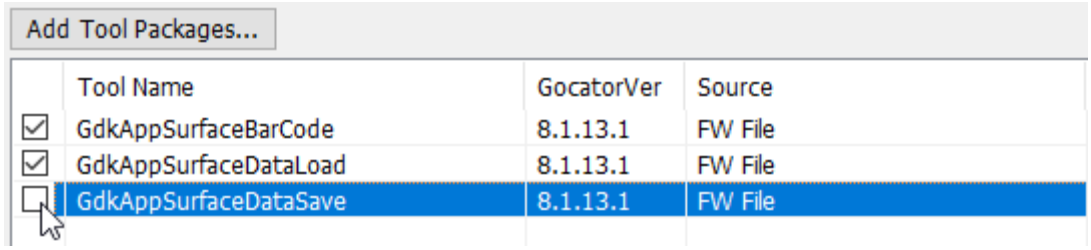


The tool manager adds the package to the tool list.

Note that 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 or folder. 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 or folder.



> Be sure to check at least one tool. If a tool is not checked, it will not be added to GoPxL.

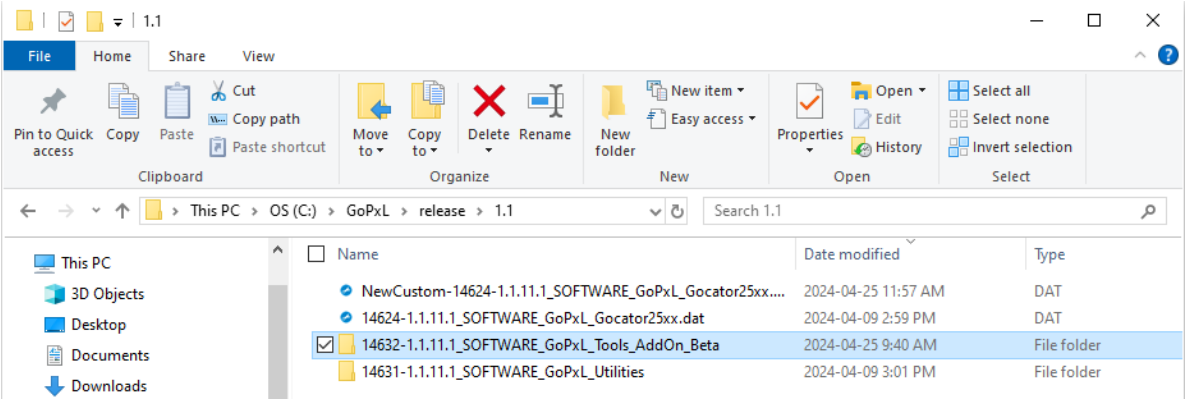
9. (Optional) If you are adding tools to a firmware, in the **Output File Prefix** field, change the default "NewCustom" to something that will help you remember what the new firmware is for.



This field is hidden when you are adding tools to GoPxL for Windows folder.

10. Click **Run**.

    If you are adding tools to a firmware, the tool manager creates a new firmware file 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.

    If you are adding tools to a GoPxL for PC folder, the tool manager updates it with the added tools.

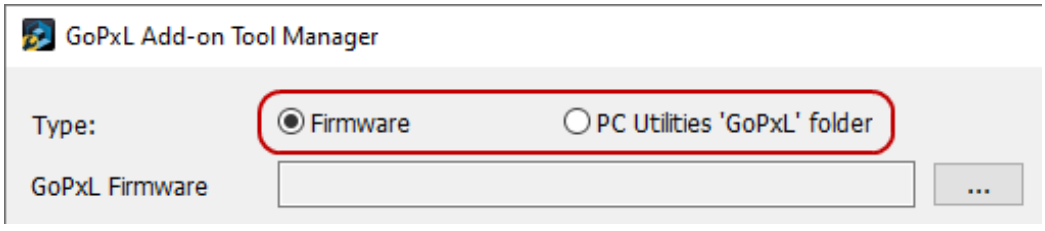
After you have successfully created the new firmware or PC version of GoPxL, you can upload it to any compatible sensor (for more information, see *Software Upgrade* on page 1184) or run it (for more information, see *Running GoPxL on a Windows PC* on page 1188).

Removing Beta Tools

To remove a tool you previously added to a firmware or from GoPxL for PC:

1. Follow steps 1 to 5 in *To add beta tools to a firmware or to GoPxL for PC:* on page 1302

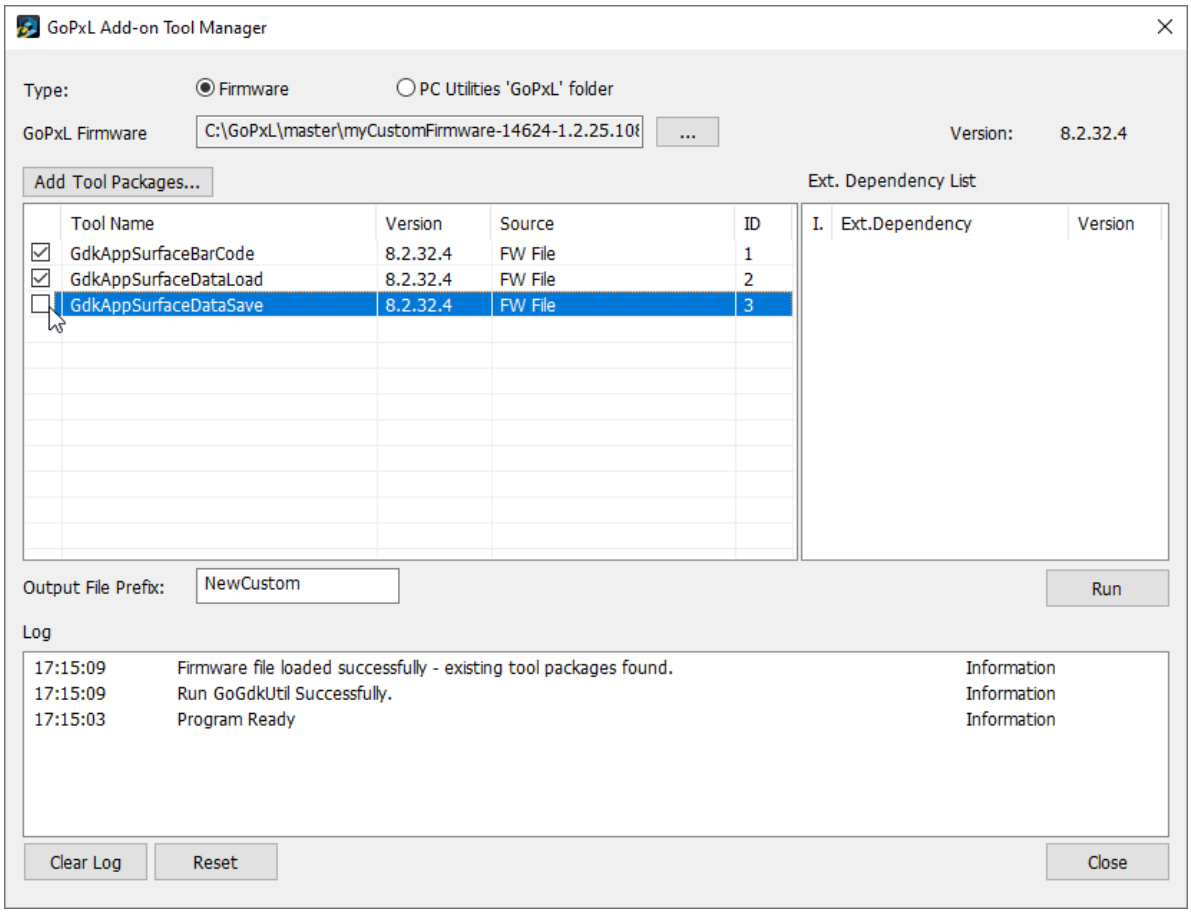
   The type should match the type used when the tool was added.



   You must navigate to the location of the previously modified GoPxL (firmware or PC version).

   After loading GoPxL, the tool list is automatically populated.

2. In the tool list, deselect the tools you want to remove.



3. Click **Run**.

   The GoPxL Add-on Tool Manager saves a new version of GoPxL with the deselected tools removed.

   > The tool manager overwrites existing firmware files with no warning.

After you have successfully created the new firmware or PC version of GoPxL, you can upload it to any compatible sensor (for more information, see *Software Upgrade* on page 1184) or run it (for more information, see *Running GoPxL on a Windows PC* on page 1188).

Pattern Editor

The pattern editor lets you modify patterns created in the Surface Pattern Matching tool (for more information on the tool, see *Surface Pattern Matching* on page 913). Although the patterns created in the Surface Pattern Matching tool will often result in good matches with your targets, you can use the pattern editor to improve the models, specifically by doing the following:

- Remove unwanted contours the Surface Pattern Matching tool has detected on edges in the scan data.
- Re-detect contours from the scan data using higher or lower levels of input image resolution (taken from the scan data) or contrast levels, compared to what the Surface Pattern Matching tool does internally.
- Identify certain contours as being required for a match to occur.
- Identify certain contours as being used to determine the position of a matched instance.

> The Pattern Editor contains a **Pattern** setting (Surface or Image). Leave this set to Surface. The Image setting is reserved for future use.

The editor uses two folders in `C:\GoTools\`, on the PC it is running on.

`GoPxLSurfacePatternMatchingSensor`

When you choose a sensor or PC instance of GoPxL in the **Source** drop-down and click **Connect**, the editor empties this folder and copies any model files on the source into the folder. For this reason, if you have previously worked on model from this folder, make sure you have uploaded any changed models to the sensor or PC instance. The models in this folder are listed in the Model Files until you switch **Source** to PC Work Directory, or empty it by clicking **Clear Sensor Work Directory**.

The folder is referred to as the "Sensor Work Directory" in the editor.

`GoPxLSurfacePatternMatching`

This is called the "PC Work Directory" in the editor. You typically use this folder to transfer model files between sensors or PC instances of GoPxL. Also, because it is only emptied if you manually do it by clicking **Clear PC Work Directory**, it may be better to work from this folder. To get models in this folder, you first connect to a source and then click **Copy all files to the PC work directory**.

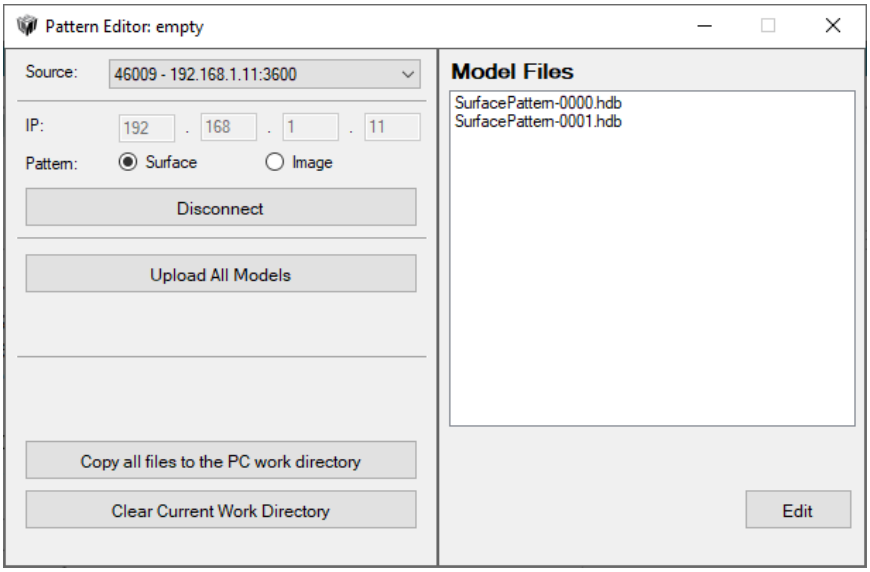
To see the files in this folder, switch **Source** to "PC Work Directory".

The pattern editor is available in the GoPxL Utilities package (`14631-x.x.x.x_GoPxL_Utilities.exe`, in the `Tools\Pattern Editor` folder. You can find the package on LMI's Product Downloads page (https://lmi3d.com/product-downloads/).

Launching the Pattern Editor

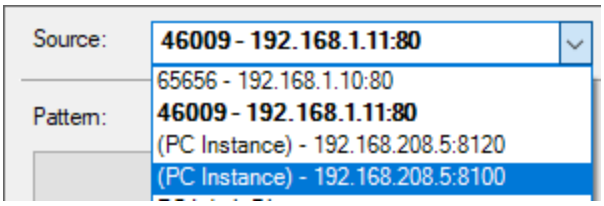
The first time you run the pattern editor, it may take longer to launch, as it will register certain DLLs required by the application.

The pattern editor can work with model files that come from an unaccelerated sensor or from a PC instance of GoPxL (which may or may not be accelerating a sensor). In all cases, the editor works with copies of the files in working folders in the local PC file system.



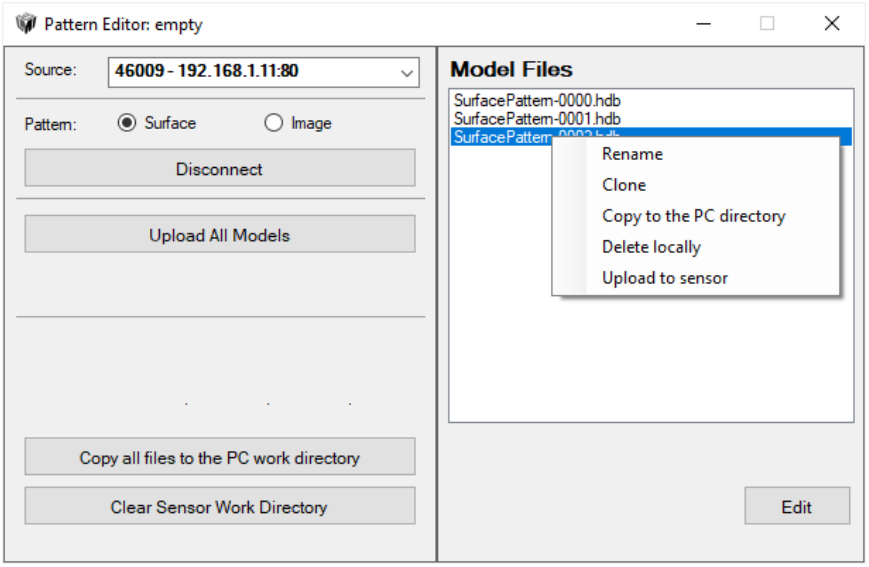
The Pattern Editor application

Parameters

Name	Description
Source	The source from which the editor fetches model files.  The currently connected sensor is displayed in bold in the **Source** list. The list contains the following types of entries:

Name	Description
	Sensor Sensors are listed with their serial number followed by their IP address. **PC Instances** Instances of GoPxL running on the PC are identified by "(PC Instance)", followed by their IP address. **PC Work Directory** This is a local directory (`C:\GoTools\GoPxLSurfacePatternMatching`) that you typically use to transfer model files between sensors and PC instances of GoPxL.
Pattern	Leave this set to Surface.
Connect / Disconnect	Connects to the source in **Source**. Note that connecting to a sensor deletes all models in the sensor working folder (`C:\GoTools\SurfacePatternMatchingSensor`) and then downloads the models currently on the sensor to that folder. For this reason, if you have modified models with the Pattern Editor, make sure to upload them before disconnecting and reconnecting. To fetch any models you have created on-sensor after connecting, disconnect using the **Disconnect** button, and then reconnect.
Upload All Models	Uploads the models from the `GoPxLSurfacePatternMatchingSensor` local folder to the connected source (to a sensor or a PC instance of GoPxL).
Edit	Opens the selected model in the model editor; you can also double-click a model in the list to edit it. For more information, see *Overview of the Editor* on the next page.
Copy all files to the PC work directory Copy all files to the sensor work directory	Copies files between the PC Work Directory and the Sensor Work Directory. The name of this button depends on whether **Source** is set to **PC Work Directory**, or to a sensor or PC instance of GoPxL.
Clear Sensor Work Directory Clear PC Work Directory	Removes all models from the local working folder. The name of this button depends on whether **Source** is set to **PC Work Directory**, or to a sensor or PC instance of GoPxL. When you clear the sensor work directory, you are only deleting the files locally, not on the sensor or PC instance.

By right-clicking on a model in the list of models, you can perform the operations listed below.



Rename: Renames the model.

Clone: Makes a copy of the selected model using the name you provide.

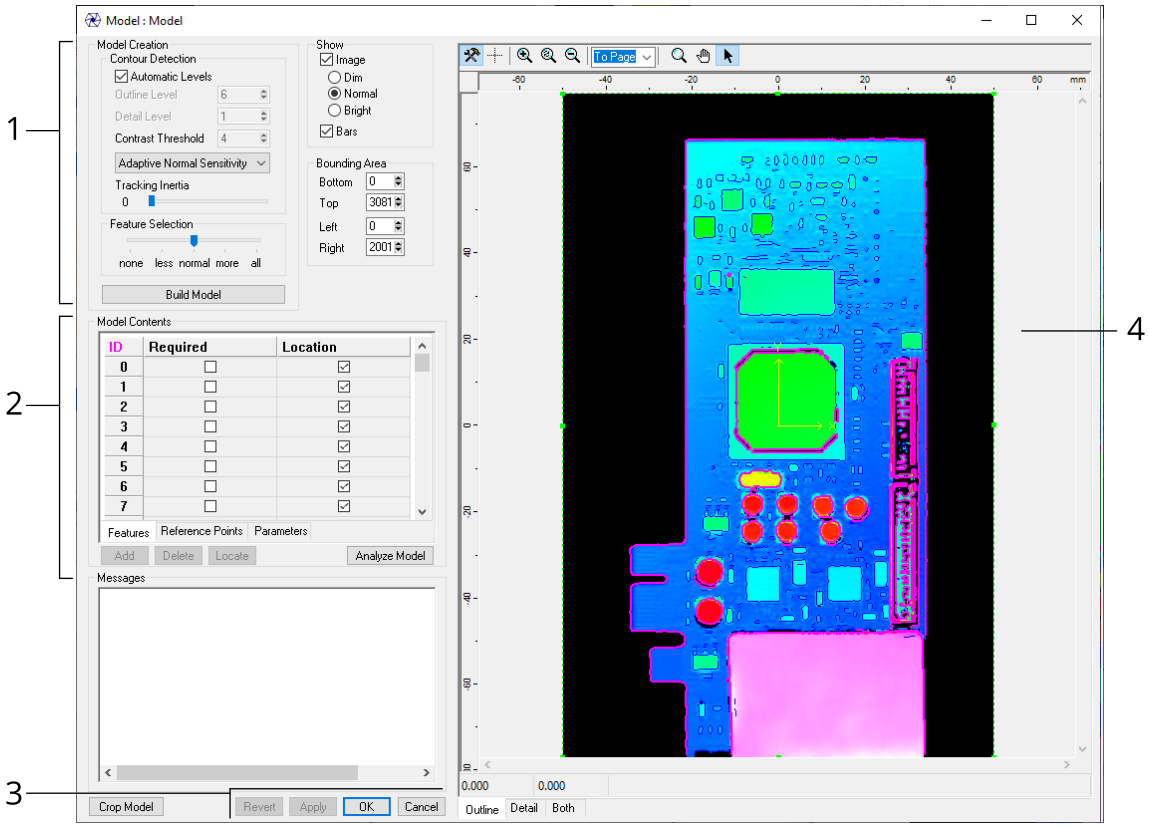
Copy to the PC directory / **Copy to the sensor directory**: Copies the model to a local directory.

Delete locally: Deletes the model from the `GoPxLSurfacePatternMatchingSensor` folder. Not displayed when **Source** is set to "PC Work Directory".

Upload to sensor: Uploads the model to the connected sensor.

Overview of the Editor

After clicking Edit in the pattern editor helper application, the selected model opens in the editor window.



	Element	Description
1	Model Creation pane	Settings related to contour detection and feature selection. After configuring these settings, or resizing the model's bounding box (green dotted line), you must rebuild the model using the **Build Model** button.
2	Model Contents	The list of the features in a model (contours used in recognition and location of an instance). Note that some model contents (reference points and some settings in the **Parameters** tab) are not currently supported by the Surface Pattern Matching tool.
3	Save and discard buttons	Used to apply changes to a model, revert to the model's original state when it was loaded, and so on.
4	Outline, Detail, and Both tabs	The editor tabs that show the Outline and Detail levels of the model. The **Both** tab shows both levels together, but you can't edit models on this tab.

Models

Models are made up of features selected from the source contours detected either by the Surface Pattern Matching tool or by the model editor itself (if you rebuild the model using the **Build Model** button). The features are used to identify and locate instances in the scan data.

Contours, and the features selected from the contours for use in recognizing and locating an instance, work on two “coarseness” levels: the Outline level and the Detail level.

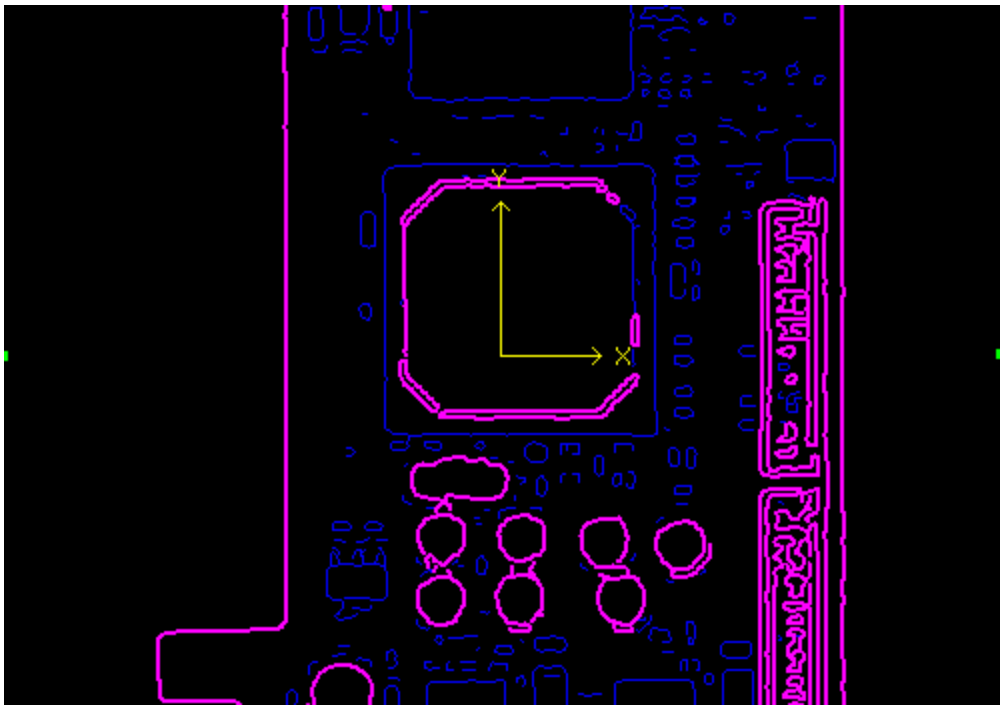
Outline: Used to quickly identify potential instances of a pattern in scan data. The Outline level is the "coarse" level of contours / features. The features at this level can be less stable, as they are not used to calculate the location of the instance. For example, a label whose position might change from frame to frame or a hole whose size might change from frame to frame could be kept at the Outline level.

Detail: Used to confirm whether an instance is in fact valid and to refine its location. The Detail level is the "fine" level of contours / features. The features at this level must be more stable and rigid with respect to one another. For this reason, given Surface scan data, include features that are all on the same plane to ensure that their positions will not be unstable due to parallax or other scanning issues. Furthermore, features on a part that might vary in size from frame to frame, or change position (such as a label), should be excluded.

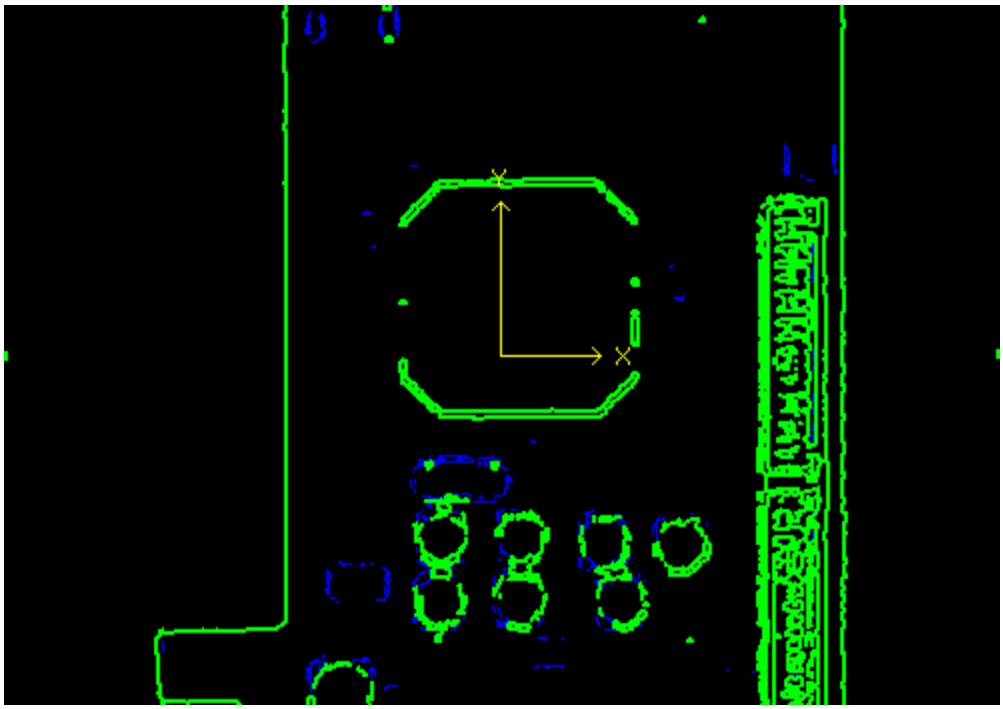
That said, the Outline and Detail levels will often be similar in terms of which features are included.

You can edit (add and remove) features at these levels separately, in the Outline and Detail panels in the main editor window. For more information on adding and removing features, see *Adding and Removing Features Manually* on the next page.

In the editor, “unused” contours (those not selected to take part in instance recognition or locating) are indicated with dark blue paths. Features (contours selected to take part in instance recognition or locating) are indicated with either magenta paths (at the Outline level) or with green paths (at the Detail level); the features in a model are listed in the **Model Contents** pane.



Dark blue unused contours and magenta features at the Outline level.



Dark blue unused contours and green features at the Detail level.

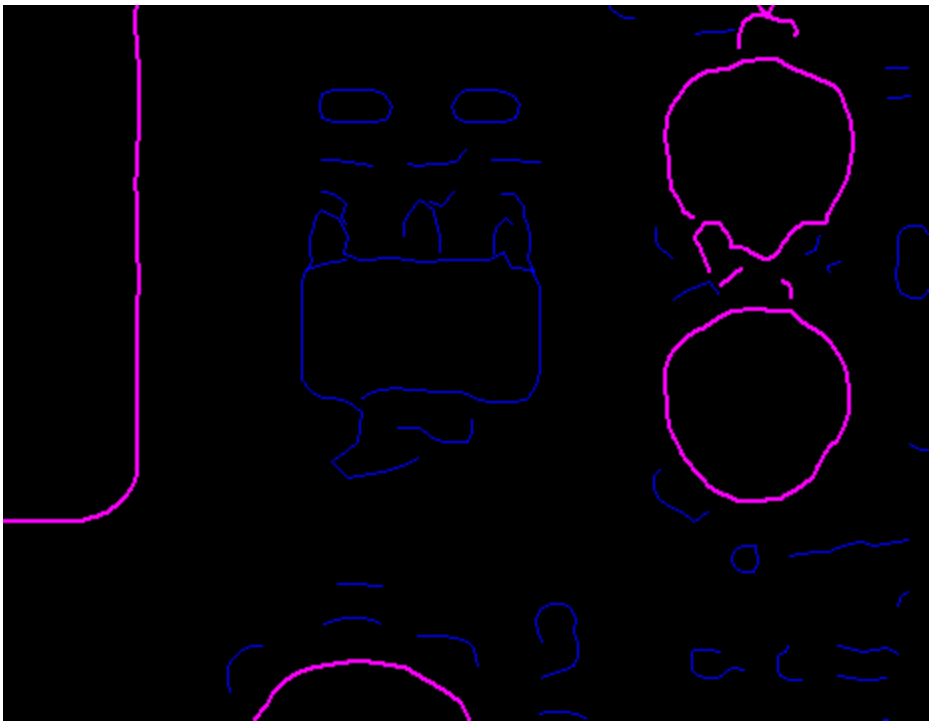
> In the Surface Pattern Matching tool, only the Detail level of features is displayed.

Adding and Removing Features Manually

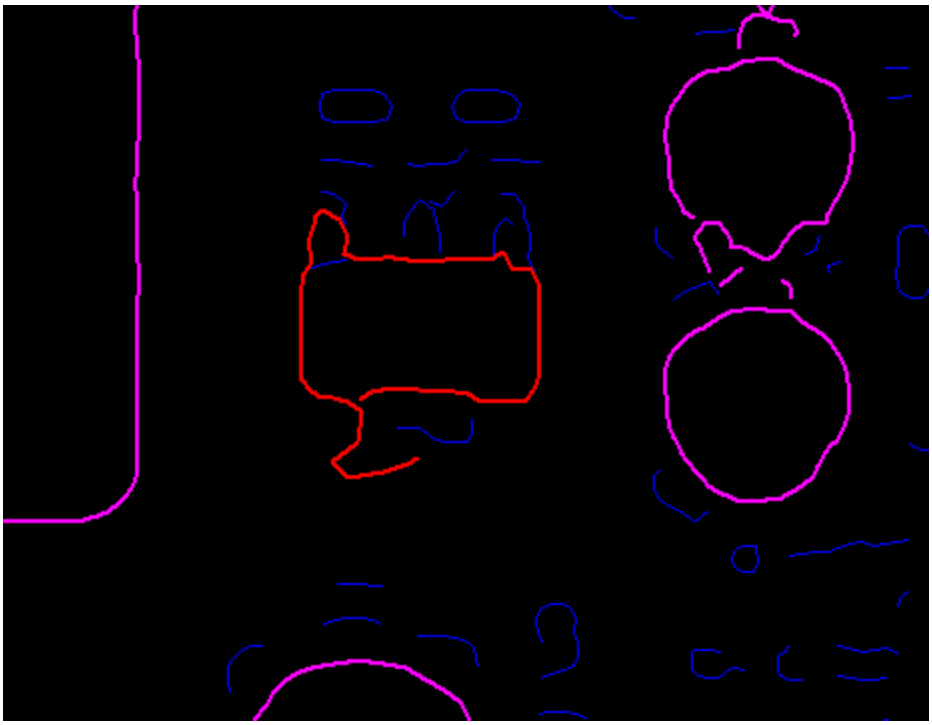
You can manually add features to a model from the source contours, or remove features currently in a model, at both the Outline and Detail levels. This can be useful if the model produced by the Surface Pattern Matching tool includes features related to parts of targets that could change or be present/absent from frame to frame. You should only include features that are constant from frame to frame.

> Adding and removing features works in the same way in the Outline and Detail tabs.

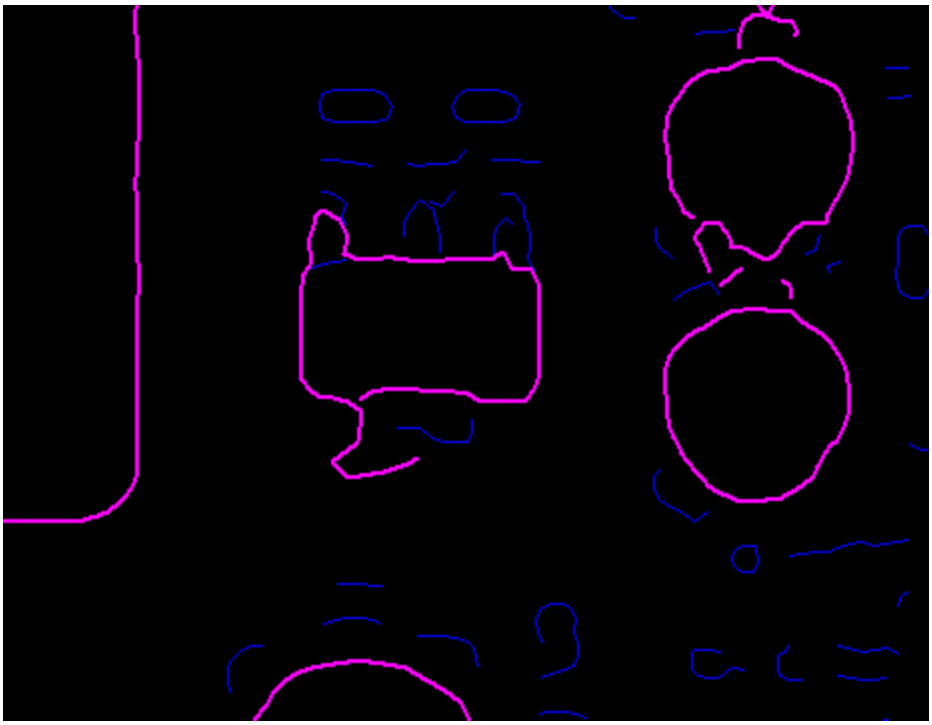
To add a feature from a source contour, double-click a dark blue contour in either the Outline or Detail tab and click Add or press the Insert key on your keyboard.



Dark blue unused contour (contours already added as features in the model are magenta).

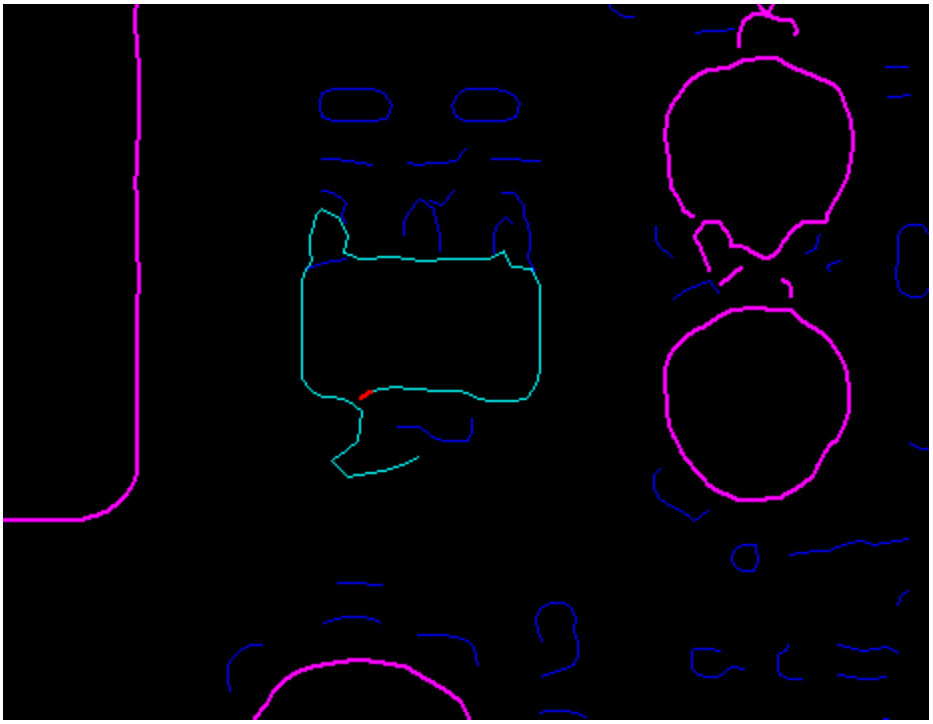


Contour selected by double-clicking it.



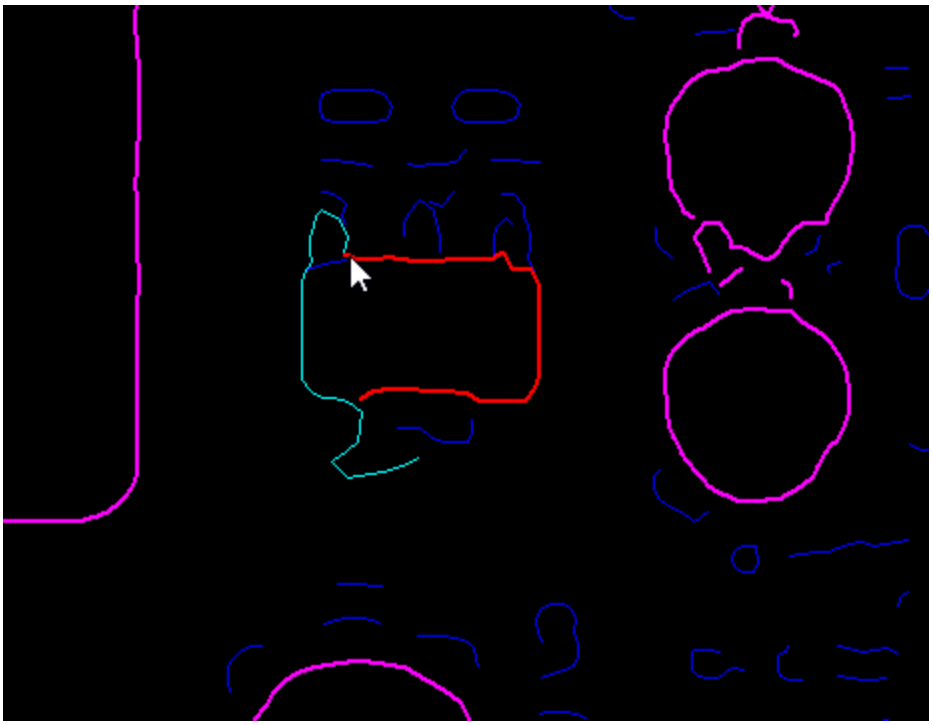
Contour added as a feature in the model (magenta).

If you single-click a dark blue unused contour, it turns cyan and lets you select segments of the contour.

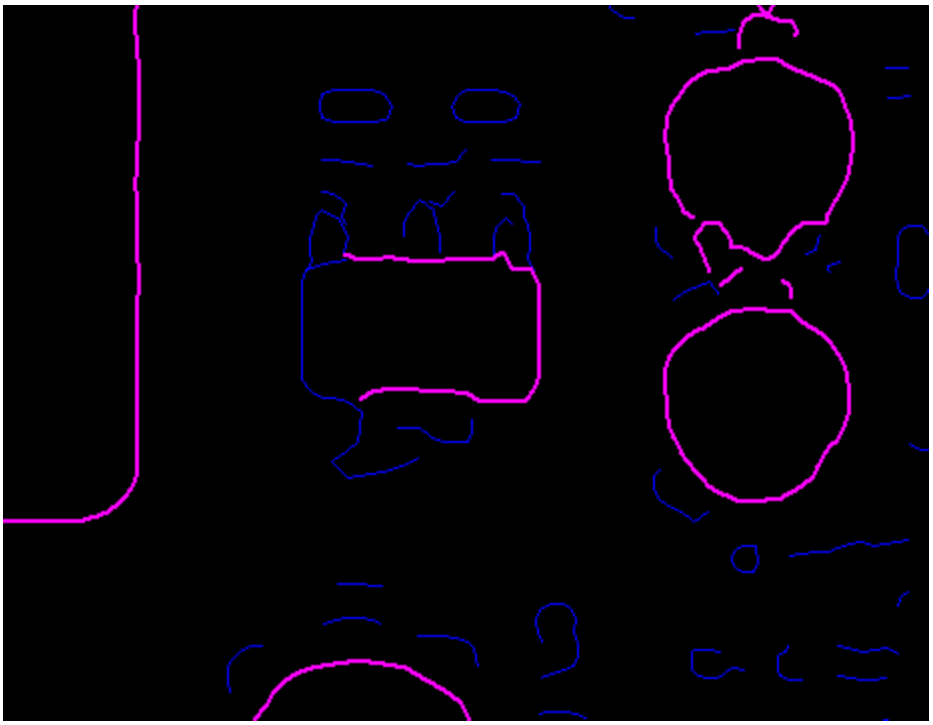


Contour highlighted in cyan with a selected segment (red).

Pressing the Ctrl key on your keyboard and clicking another segment selects a portion of the cyan path.



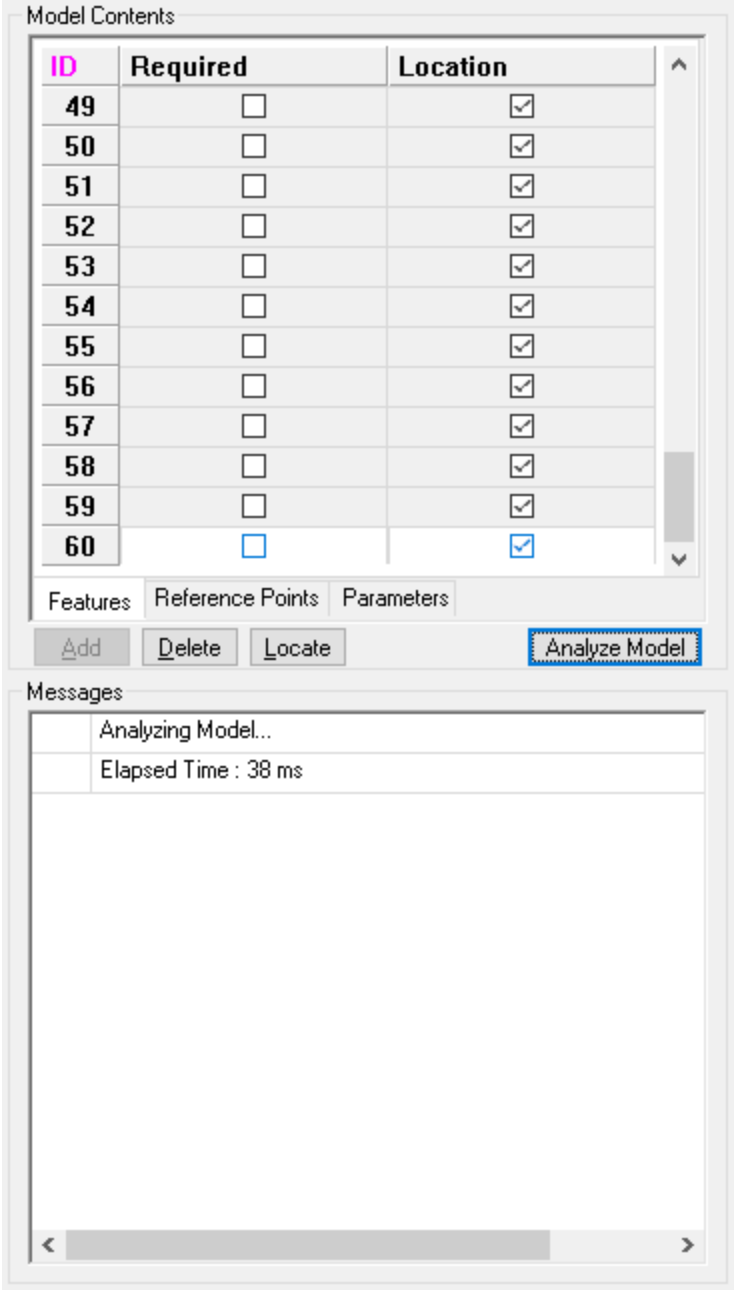
Clicking **Add** or pressing the Insert key on your keyboard adds the segment of the contour as a feature to the model.



After adding a feature, it is added to the list of features on the **Feature** tab in the **Model Contents** panel. Do not press the build button after adding or removing features, or you will lose the modifications that were just made. You must however save the changes; for more information, see *Saving and Discarding Changes* on page 1323.

To remove a feature, click a magenta or green path in the editor to select it and click **Delete** or press the Delete key on your keyboard. After removing a feature, it is removed from the list of features on the Feature tab in the Model Contents panel. You do not need to build the model, but must save the changes; for more information, see *Saving and Discarding Changes* on page 1323.

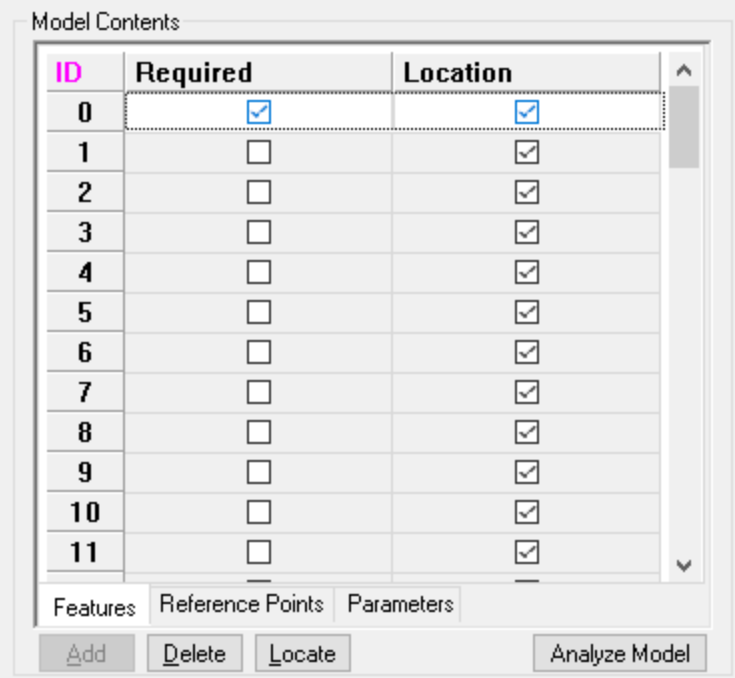
After adding a feature to a model or removing a feature from a model, you should analyze the model by clicking **Analyze Model**. Make note of any errors in the **Messages** panel.



No error messages after clicking Analyze Model.

Setting Required and Locating Features

In the list of features in the **Model Contents** pane, you can indicate that a feature is “required” or that it is used to calculate the location of an instance by checking the appropriate checkbox next to the feature.



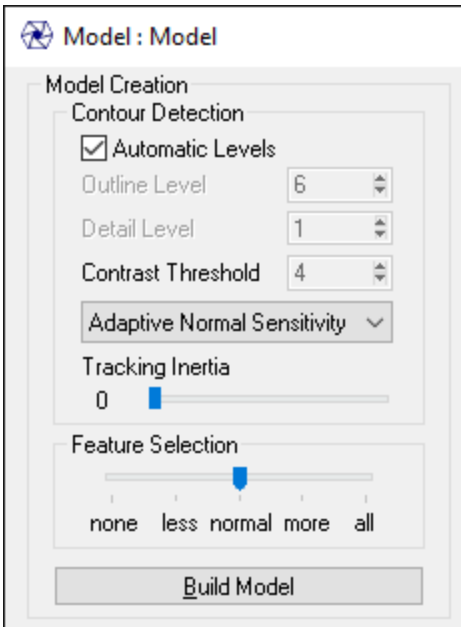
When Required is checked for a feature, it *must* be found by the Surface Pattern Matching tool in order for an instance to be identified.

When Location is checked for a feature, the Surface Pattern Matching tool uses the feature to calculate the location of instances. If a feature's location is not checked, it is only used for instance recognition. An example of the latter is a tag or label glued to an object. Although the label's contours (it's shape or what is written on it) might be unique enough to help recognize an instance, it's position *on* the object (that is, relative to the other features) might vary in its position from frame to frame. For this reason, it might be useful for instance recognition, but not for determining the location of the object.

Model Creation Settings and Rebuilding

The Surface Pattern Matching tool uses internally fixed settings to detect contours in the scan data and then select features from those source contours. In the pattern editor, you can increase or decrease the contour detection levels, change the contrast threshold, and so on, and then rebuild the model. This can be useful if the Surface Pattern Matching tool's internal settings do no produce the right amount of source contours and subsequently features.

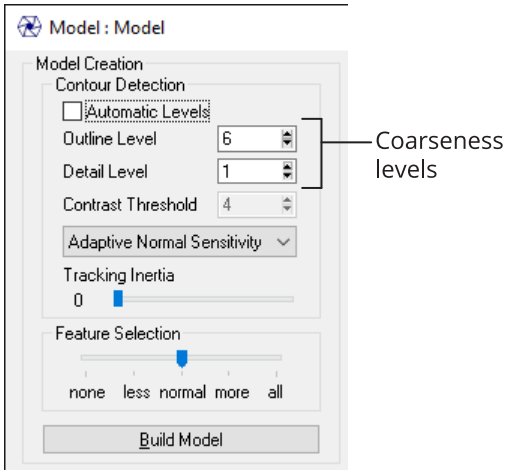
The settings described here are found in the **Model Creation** section of the model editor.



After making changes to any of these settings, you must rebuild the model by clicking **Build Model**, and then save the changes. You should also click **Analyze Model** after rebuilding a model. Pay special attention to messages in the **Messages** pane at the bottom of the editor to make sure there are no errors. For more information on saving changes, see *Saving and Discarding Changes* on page 1323.

Coarseness Levels

By default, the pattern editor uses automatically determined contour coarseness values (at both the Outline and Detail levels) to detect contours in the scan data image. If you uncheck **Automatic Levels**, you can change the **Outline Level** and **Detail Level** values to generate more or fewer contours, from which you can then select features that more reliably represent your target.

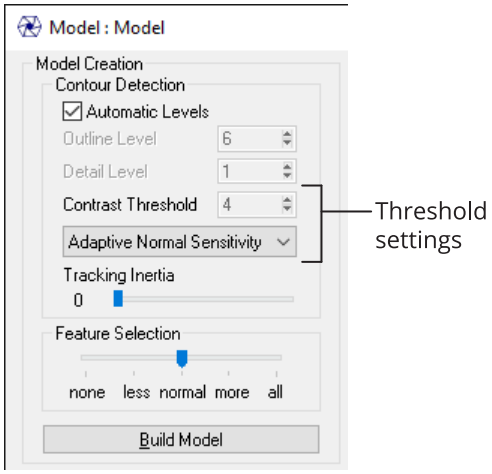


The **Outline Level** and **Detail Level** values range from 1 to 16. At the lowest value, contours are detected in a full-resolution version of the image based on the scan data, which results in more

contours from which to choose features. At higher values, contours are detected in a reduced-resolution version of the image based on the scan data: the resolution is reduced by the setting's value, which results in fewer contours being detected. Note that **Detail Level** must be less than or equal to **Outline Level**.

Thresholds

You can adjust the level of sensitivity the pattern editor uses to detect contours in the scan data image.



By default, the sensitivity is set to Adaptive Normal Sensitivity, but you can set it to one of the following adaptive sensitivity levels, or to a fixed threshold value (see below).

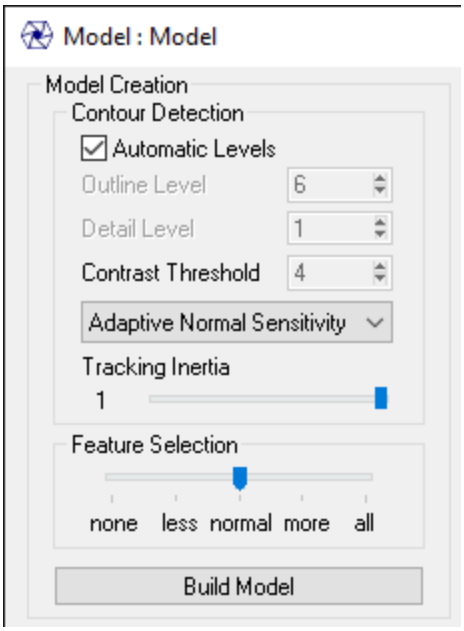
Adaptive High Sensitivity: Results in more low-contrast contours, but also noise.

Adaptive Low Sensitivity: Results in strongly defined contours and eliminates noise, but may miss important contour segments.

If you set the dropdown to Fixed Value, you can then set a fixed threshold in **Contrast Threshold**. The **Contrast Threshold** value corresponds to the minimum step required to detect corners. A lower value generates more contours when you rebuild the model, but may also result in noise.

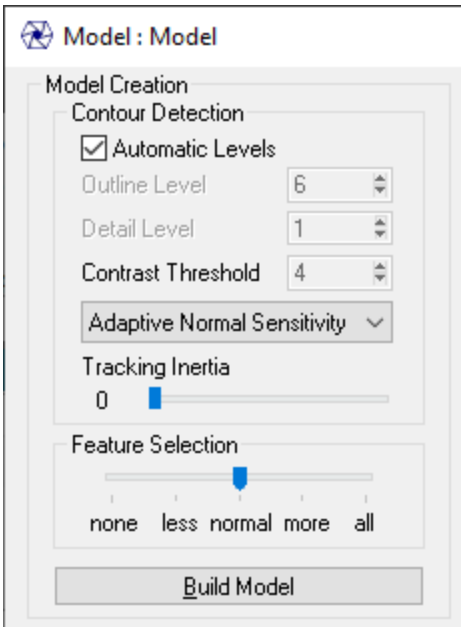
Tracking Inertia

Setting the Tracking Inertia slider to 1 closes small gaps in the source contours, connecting contours that might otherwise be broken into smaller sections.



Feature Selection

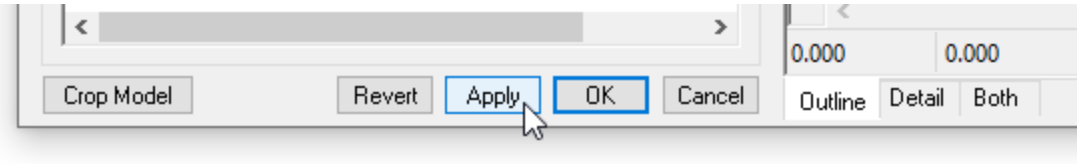
This setting ranges from **none** to **all**, which determines which features the pattern editor selects from the detected contours and adds to the model when you rebuild it. You should use **none** (which adds no features to the model) if you want to manually add features to the model from the detected contours. The **normal** setting tries to add the most appropriate features to the model; use this setting with simple to moderately complex parts. The **all** setting adds all detected contours as features to the model; only use this with very complex parts, such as electronic parts.



Saving and Discarding Changes

After making changes to a model (either adding or removing features, or re-detecting contours by clicking **Build Model**), you *must* do the following:

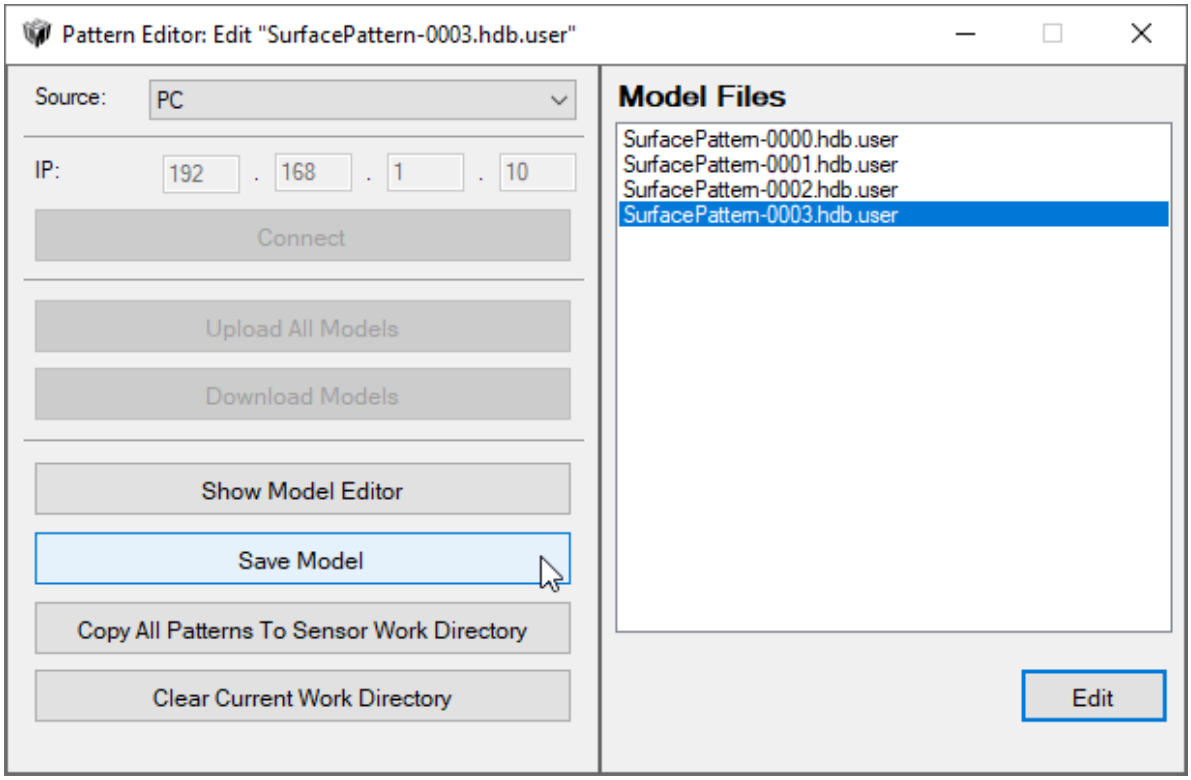
1. In the model editor, at the bottom of the window, click **Apply** or **OK**.



   Clicking **Apply** leaves the model editor open. Do this if you want to continue working on a model (for example, if you want to test the model in Gocator before closing the model).

   Clicking **OK** closes the model editor.

2. In the pattern editor helper application, click the model you were working on, and click **Save Model**.



   The model is saved to the working folder.

> After making changes to a model, you must re-load the pattern in the instance of Surface Pattern Matching to see the changes.

Before saving changes (either by clicking **Apply** or **OK**), you can revert the model to its initial state by clicking **Revert**.

Miscellaneous

Reference points, which you can create in the pattern editor on the **Reference Points** tab in the **Model Contents** panel, are not currently supported by the Surface Pattern Matching tool.

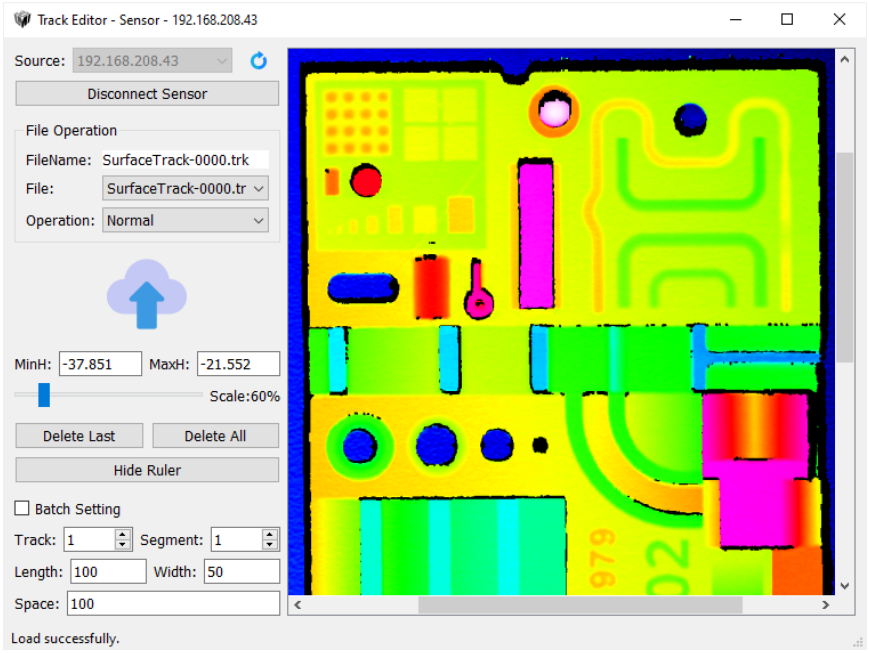
The Surface Pattern Matching tool does not currently support the custom shading area (on the **Parameters** tab in the **Model Contents** panel).

You can change the region of the model by resizing the green bounding box (or setting its dimensions in the **Bounding Area** section in the model editor) and rebuilding the model. After resizing and rebuilding the model to the desired region, you can use the **Crop Model** button to reduce the size of the final model surface, and reduce the pattern file size.

In the Show section in the model editor, you can hide the scan data to see only the contours and features by unchecking **Image**. Note that the Dim / Normal / Bright options below **Image** only apply to intensity data.

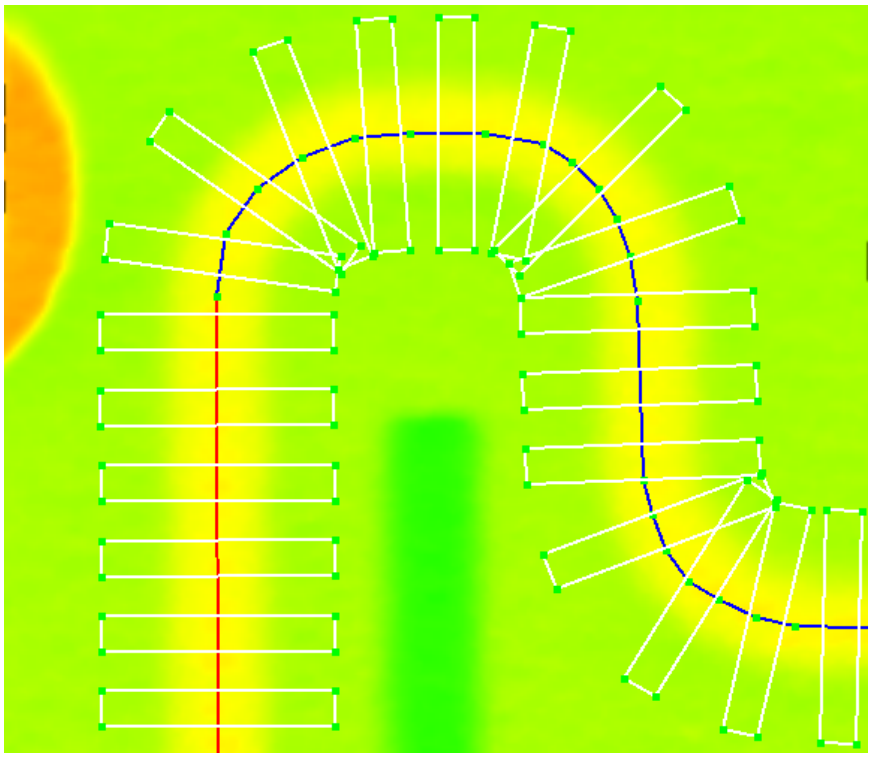
Track Editor

You use the track editor in conjunction with the Surface Track tool. The track editor lets you configure "path" and "ruler" information on a frame of scan data from a sensor or GoMax device, a PC instance of GoPxL, or a GoPxL backup file. The Surface Track tool uses this information to inspect targets along the defined path. For more information on paths, rulers, and other concepts related to the Surface Track tool, see *Surface Track* on page 980.



The track editor

In the track editor, you define one or more paths, and configure rulers along these paths.

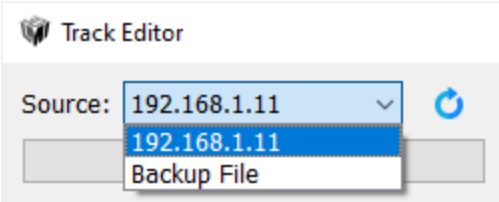


Closeup of the track editor window, showing a track of material on a surface (yellow on green), path points (green dots), a path (blue segments; red segment for the currently selected segment), and rulers (white rectangles).

> The following assumes that you have already scanned a representative target and created a track file from within the Track tool. For more information, see the first steps of *To configure the Track tool:* on page 984.

To load and work with scan/track data:

1. In the track editor, in the **Source** drop-down, choose one of the following:



- **An IP address**: Choose this option to edit a track file on a sensor or a GoMax device, or in a PC instance of GoPxL.
- **Backup File**: Choose this option to edit a track file in a backup file you have created from a sensor or a PC instance of GoPxL and saved in the PC's file system. For more information on backup files, see *Backup and Restore* on page 1184.

> If you only see "Backup File" but expect to see an IP address, make sure the device or PC instance is running and is connected to your network, and then click the Reset button ( ).

2. If you have chosen an IP address, click **Connect Sensor**.

   The track editor loads the data from the device or PC instance.

3. If you have chosen a backup file, click **Load Backup**, navigate to the location of the backup file, select it, and click **Open**.

   The track editor loads the data.

4. Under **File Operation**, in **File**, select the track file you have created in the Surface Track tool, and under **Operation**, choose Load.

> If you only see "External" under **File**, it means no track file has been defined on the device / PC instance or in the backup file. You can use "External" to load a .trk file from the PC's file system. Otherwise, create a track file in the Surface Track tool, and then disconnect and reconnect the source.

   The editor loads the specified track file.

   For a complete list of the file operations, see *File operations* below.

5. 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. This does not change the height map data.

File operations

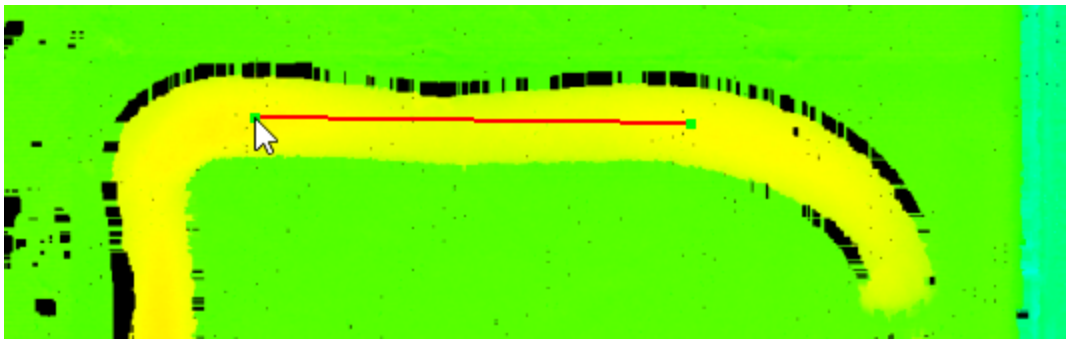
Operation	Description
Normal	The utility defaults to this option after performing another operation.
Load	Loads the track file selected in the **File** list. If you choose "External" in the **File** list, choosing Load displays a dialog that lets you load a file on the PC.
Save	Saves the currently loaded track file (as shown in **FileName**).
Save As	Saves the currently loaded track file (as shown in **FileName**) as a new file.
Delete	Deletes the file selected in the **File** list.
Refresh	Refreshes the **File** list.

After you have loaded the data, you must add a path and configure its rulers.

To add and configure a path and its rulers:

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.

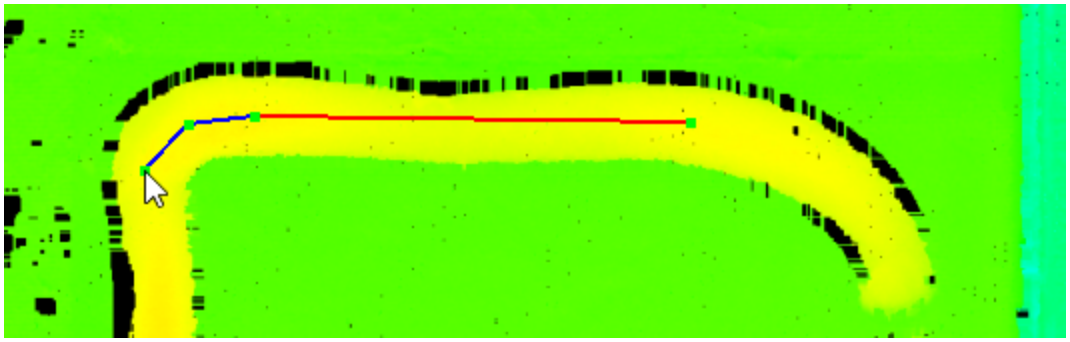
   A red segment between the first two green path points appears in the editor window.



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.

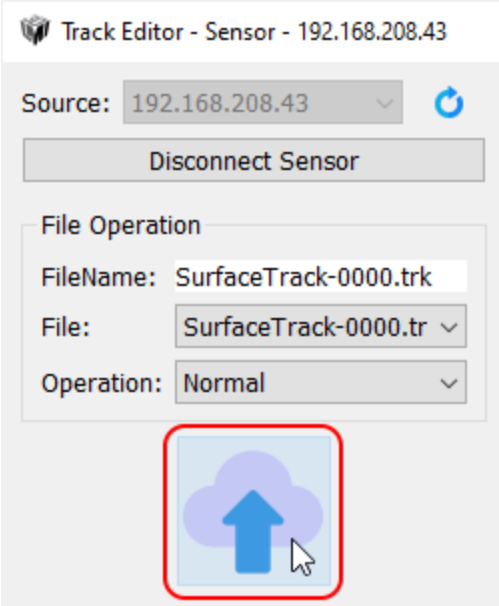
   When adding points on corners, add more points to follow the track more precisely.



3. Continue clicking until you complete the path along the track.
4. Modify the ruler settings.

   For more information, see *Ruler Settings* on the next page.
5. Under **Operation**, click **Save** to save the path information to the data.

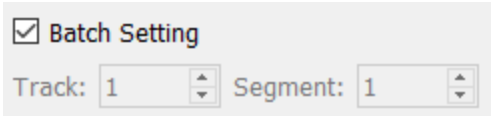
   For a complete list of the file operations, see *File operations* on the previous page.
6. Click the Sync button.



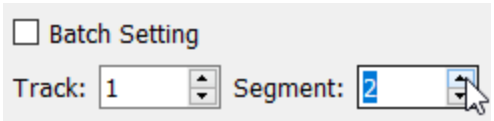
The changes are synchronized to the source (device or backup file).

Ruler Settings

After you have finished adding a path, you can configure the rulers on the path (the dimensions and the spacing of the rulers). You can choose to apply dimensions/spacing to rulers in all segments at the same time by checking **Batch Setting**.



Otherwise, you must move through the individual path segments by clicking the spinner control in the **Segment** field and setting the ruler dimensions for each segment.



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.

Setting	Description
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.

Specifications

The following sections describe the specifications of Gocator sensors and connectors, as well as Master hubs.

Sensors **1332**
Gocator 3210 Sensor 1333
Gocator 3210 1335
Gocator 3500 Series 1338
Gocator 3506 1340
Gocator 3520 1343
Sensor Connectors **1346**
Gocator Power/LAN Connector 1346
Grounding Shield 1347
Power 1347
Safety Input 1347
Gocator I/O Connector 1348
Grounding Shield 1349
Digital Outputs 1349
Inverting Outputs 1349
Digital Input 1350
Encoder Input 1350
Serial Output 1351
Analog Output 1351
Master Network Controllers **1353**
Master 100 1353
Master 100 Dimensions 1354
Master 400/800 1355
Power and Safety 1355
Encoder 1356
Input 1356
Master 400/800 Electrical Specifications 1357
Master 400/800 Dimensions 1358

Master 810/2410 1359
Power and Safety 1361
Encoder 1362
Input 1362
Electrical Specifications 1363
Encoder 1364
Input 1366
Master 810 Dimensions 1368
Master 2410 Dimensions 1369
Master 1200/2400 1370
Power and Safety 1370
Encoder 1371
Input 1371
Master 1200/2400 Electrical Specifications 1372
Master 1200/2400 Dimensions 1372

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.

For maximum system reliability, use Phoenix Contact QUINT power supplies.

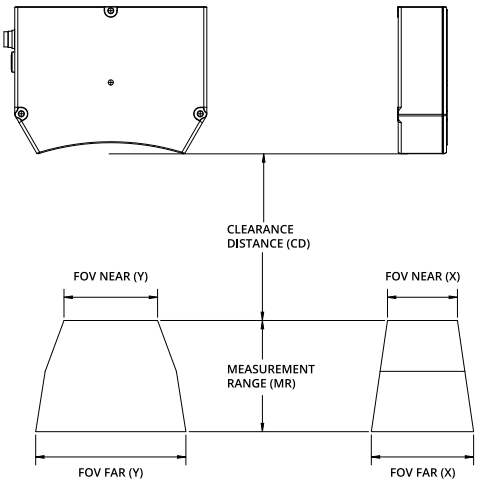
For Gocator 3506 and 3210, operate sensors at 48 VDC for optimal performance.

Linearity Z values and Resolution Z are typical values.

Field of View and Resolution XY are specified as [X] x [Y], near to far.

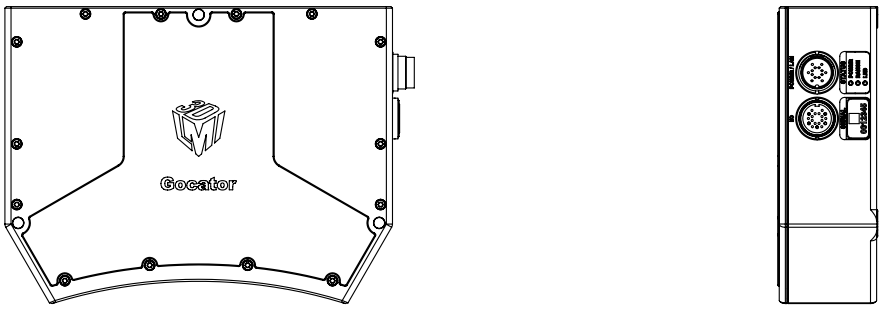
Differential encoder requires the use of Master 400/800/810/1200/2400/2410.

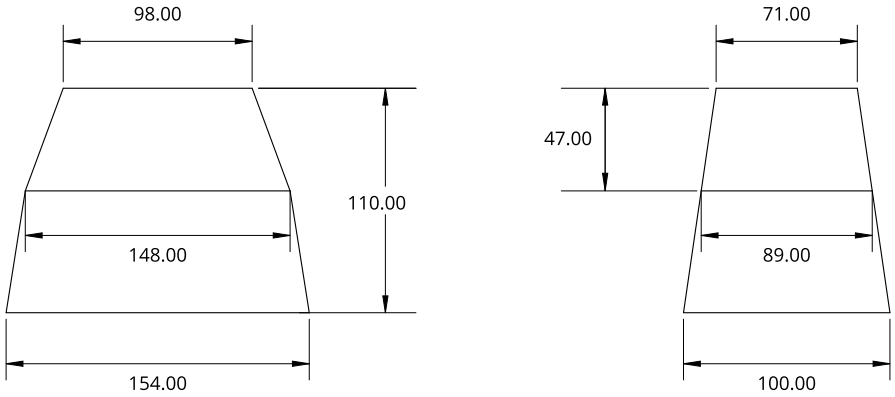
The following diagram illustrates some of the terms used in the table above.



Mechanical dimensions, CD/FOV/MR, and the envelope for each sensor model are illustrated on the following pages.

Field of View / Measurement Range / Coordinate System Orientation

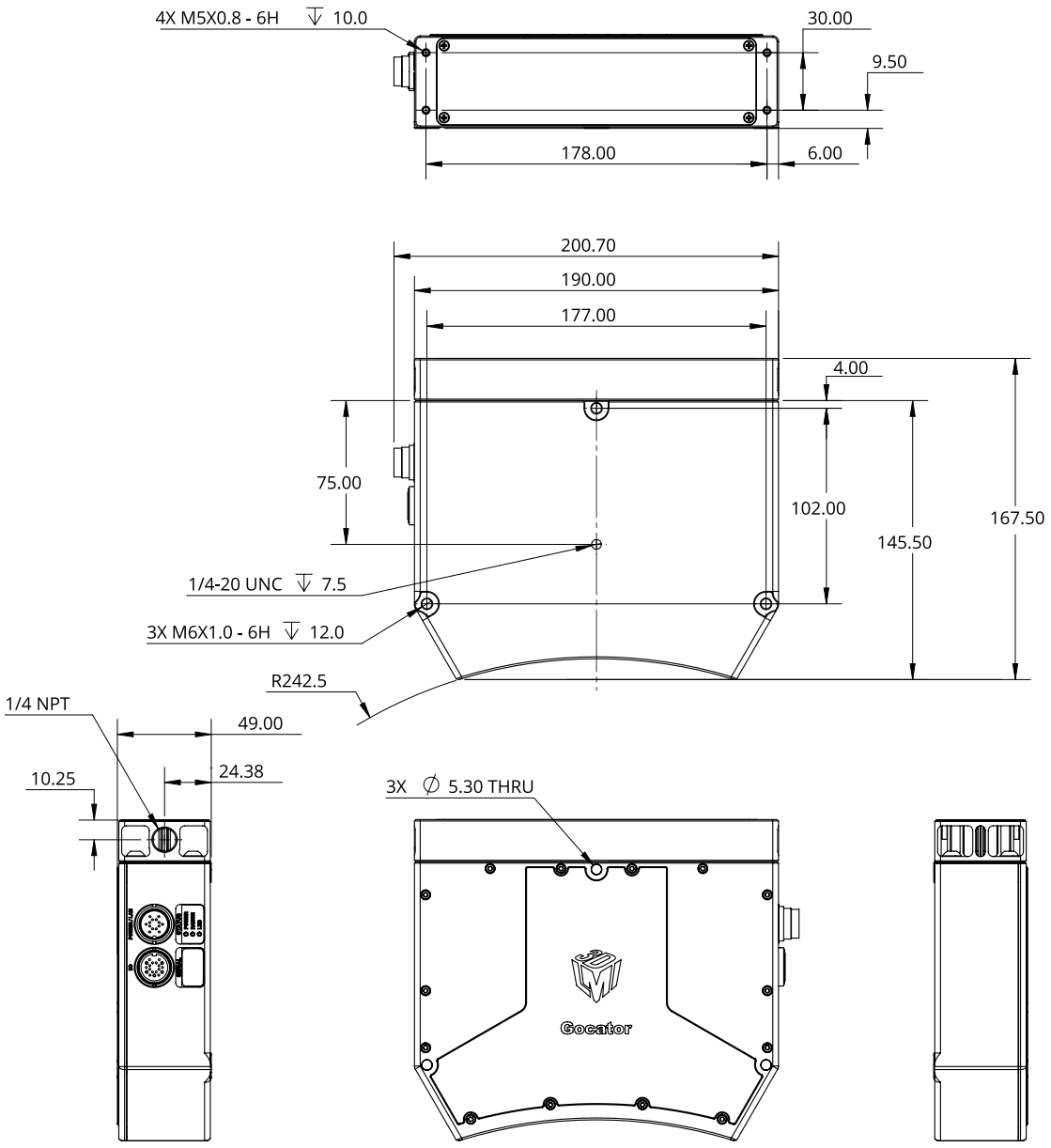




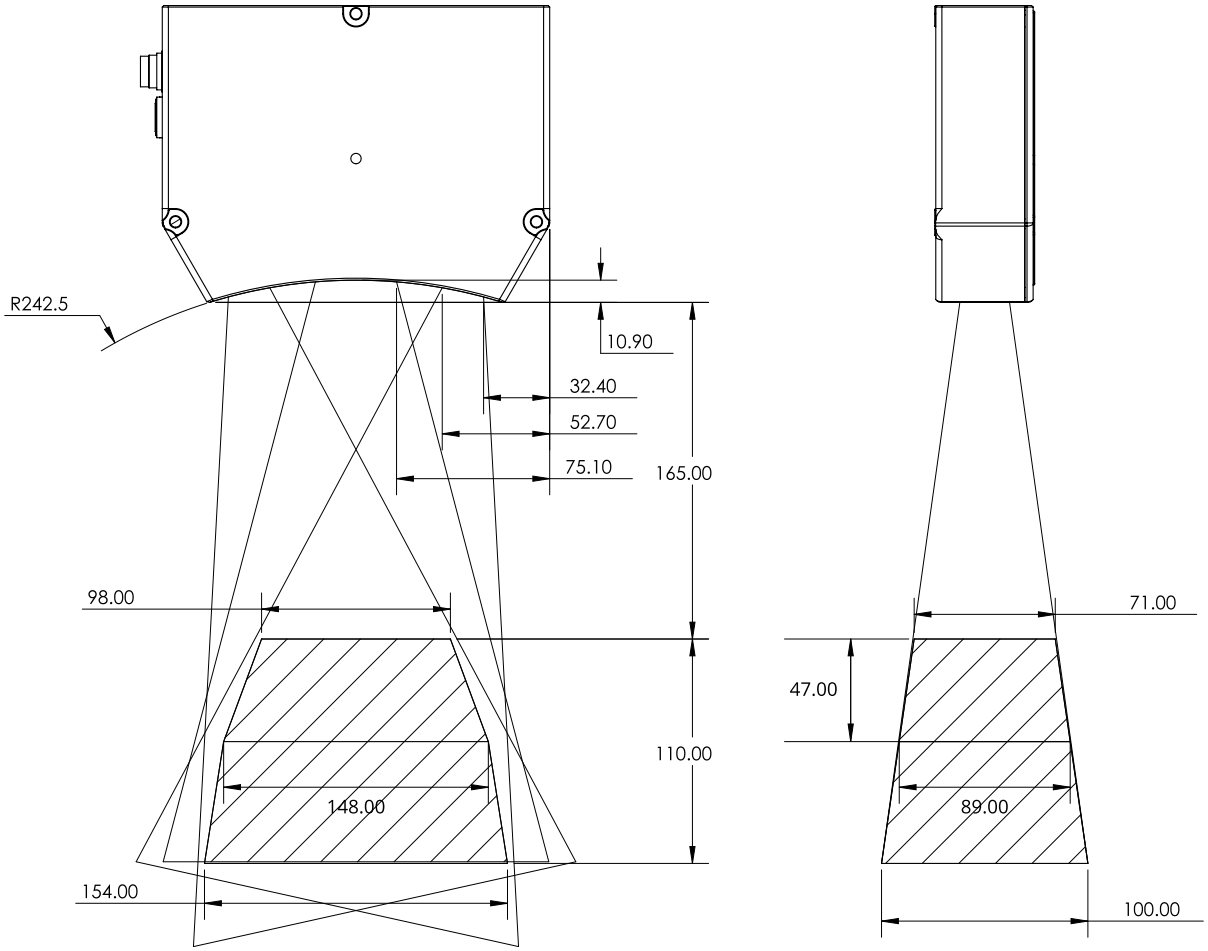


Dimensions

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	3520
Scan Rate (Hz)	6	6	3
Imagers (megapixels)	5	5	5
Clearance Distance (CD) (mm)	52.5	87.0	203
Measurement Range (MR) (mm)	7	25.0	150
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)	179 x 115 (near) 282 x 175 (far)
Repeatability Z (µm)	0.2	2.0	4.6
Resolution XY (µm)	6.7 - 7.1	20 - 25	74 - 121
VDE Accuracy (mm)	n/a	0.012 mm[1]	0.090 mm[2] - 0.200 mm[3]
Accuracy XYZ (µm)[4]	6	n/a	n/a
Input Voltage (Power)	+24 to +48 VDC (25 Watts); Ripple +/- 10%		48 VDC (50 Watts); Ripple +/- 10%
Operating Temp.	0 to 50° C		0 to 40° C
Storage Temp.	-30 to 70° C		
Dimensions (mm)	49x152x177.5	49 x 136 x 170	55 x 167 x 260
Weight (kg)	1.77	1.52	2.6

[1]Based on 2634, Part 2.

[2]VDE within central 100 mm measurement range.

[3]VDE within full 150 mm measurement range.

[4]Based on sphere-fitting at various positions in the scan volume.



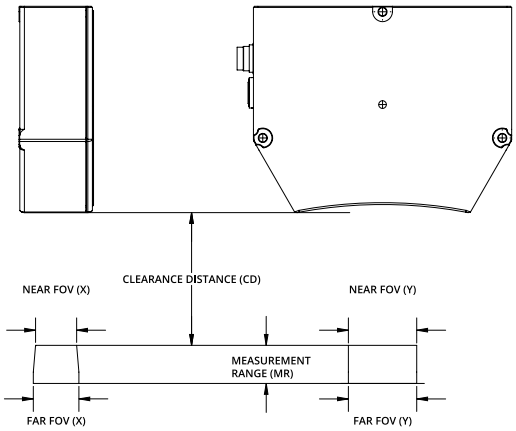
For maximum system reliability, use Phoenix Contact QUINT power supplies.



For Gocator 3506 and 3210, operate sensors at 48 VDC for optimal performance.

Field of View and Resolution XY are specified as [X] x [Y], near to far.

The following diagram illustrates some of the terms used in the table above.

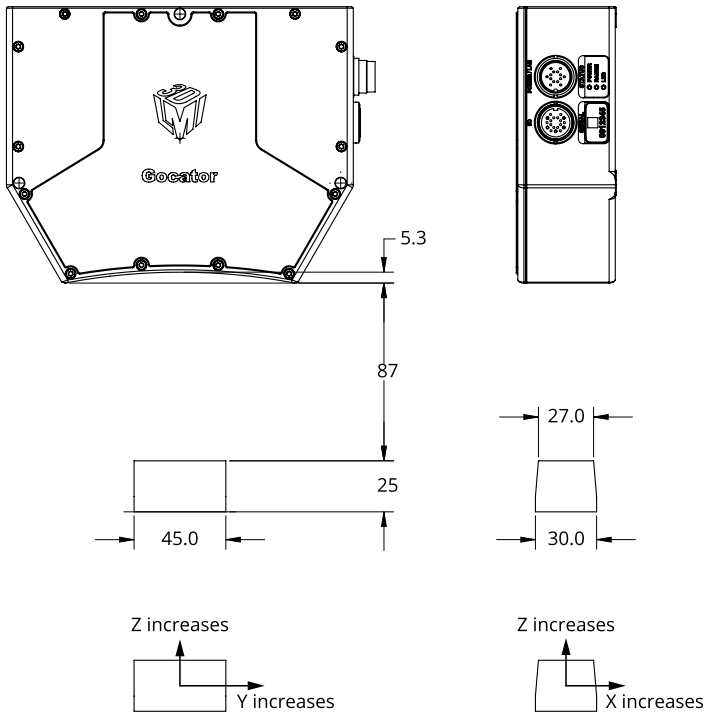


ALL 3500 SERIES MODELS	
Light Source	Blue LED (465 nm)
Interface	Gigabit Ethernet
Inputs	Differential Encoder, Trigger
Outputs	2x Digital output, RS-485 Serial (115 kBaud) Analog Output (4 – 20 mA)
Housing	Gasketed aluminum enclosure, IP67
Vibration Resistance	10 to 55 Hz, 1.5 mm double amplitude in 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

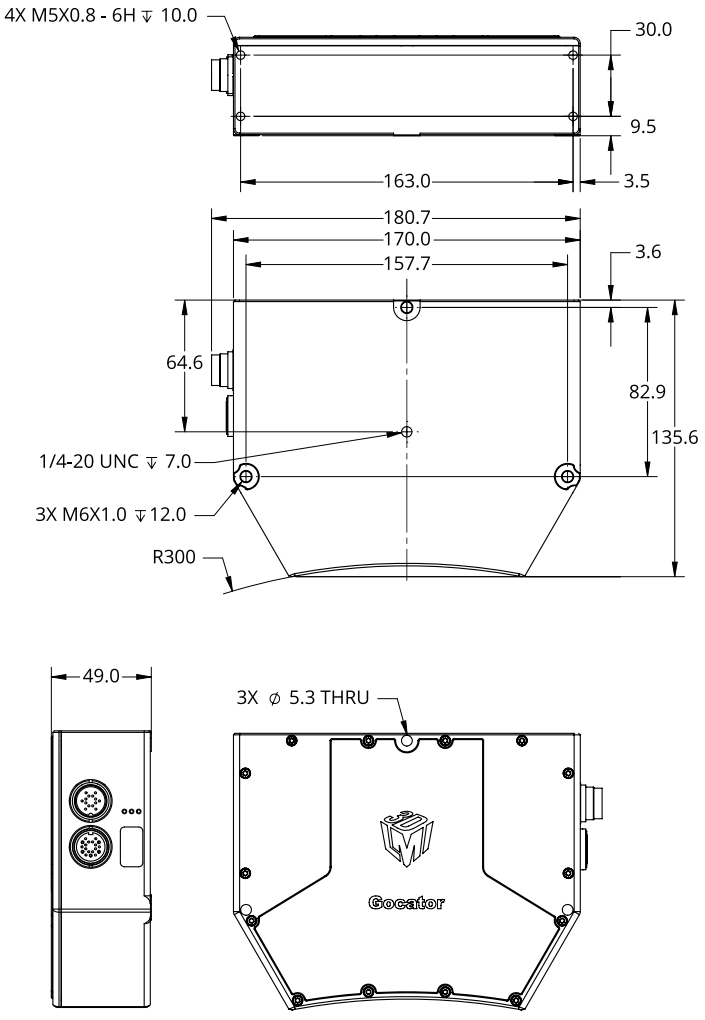
Mechanical dimensions, CD/FOV/MR, and the envelope for each sensor model are illustrated on the following pages.

Gocator 3506

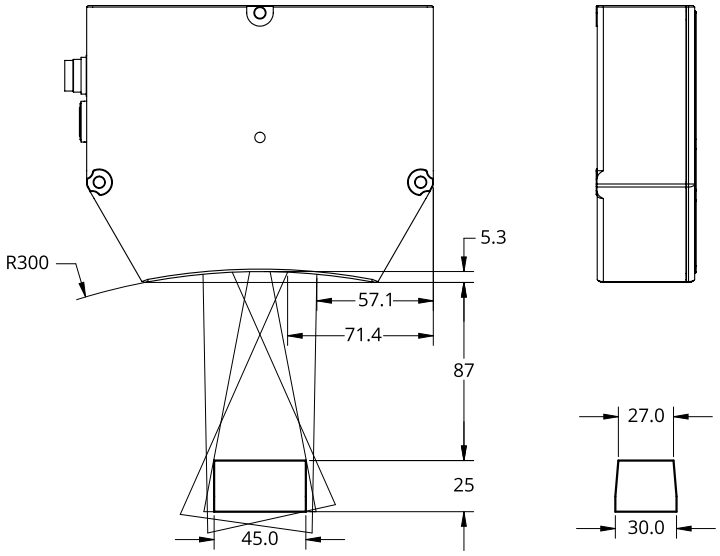
Field of View / Measurement Range / Coordinate System Orientation



Dimensions

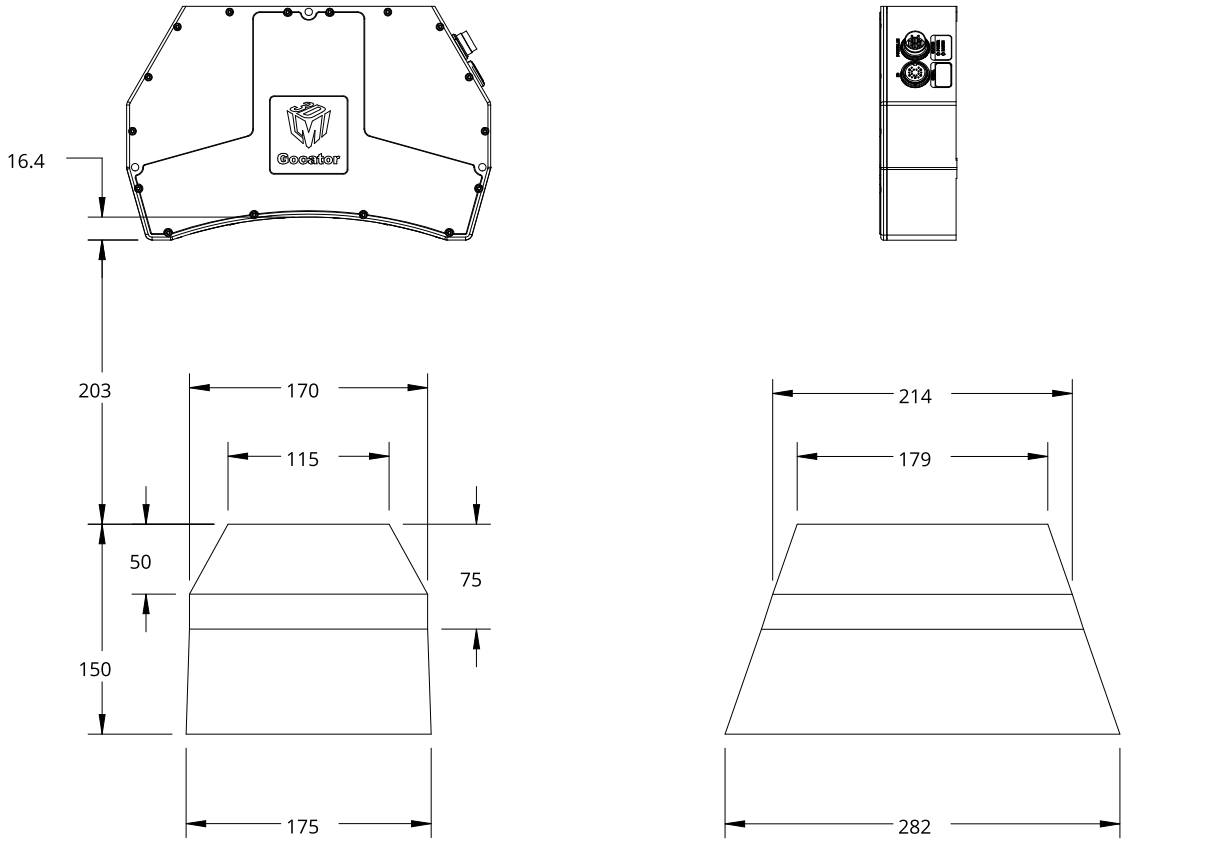

4X M5X0.8 - 6H ↧ 10.0
30.0
9.5
3.5
163.0
180.7
170.0
157.7
3.6
64.6
82.9
135.6
1/4-20 UNC ↧ 7.0
3X M6X1.0 ↧12.0
R300
49.0
3X ⌀ 5.3 THRU
Gocator

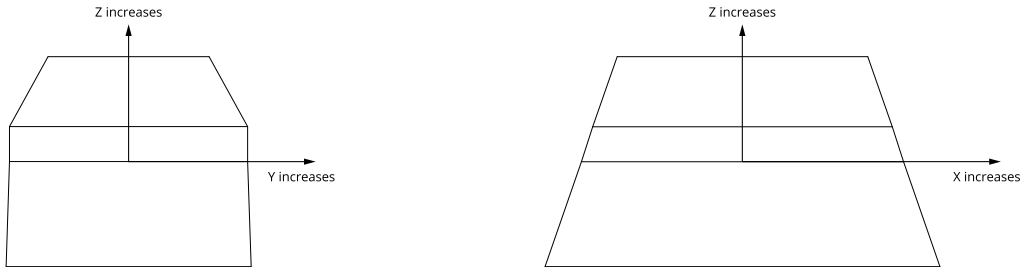
Envelope



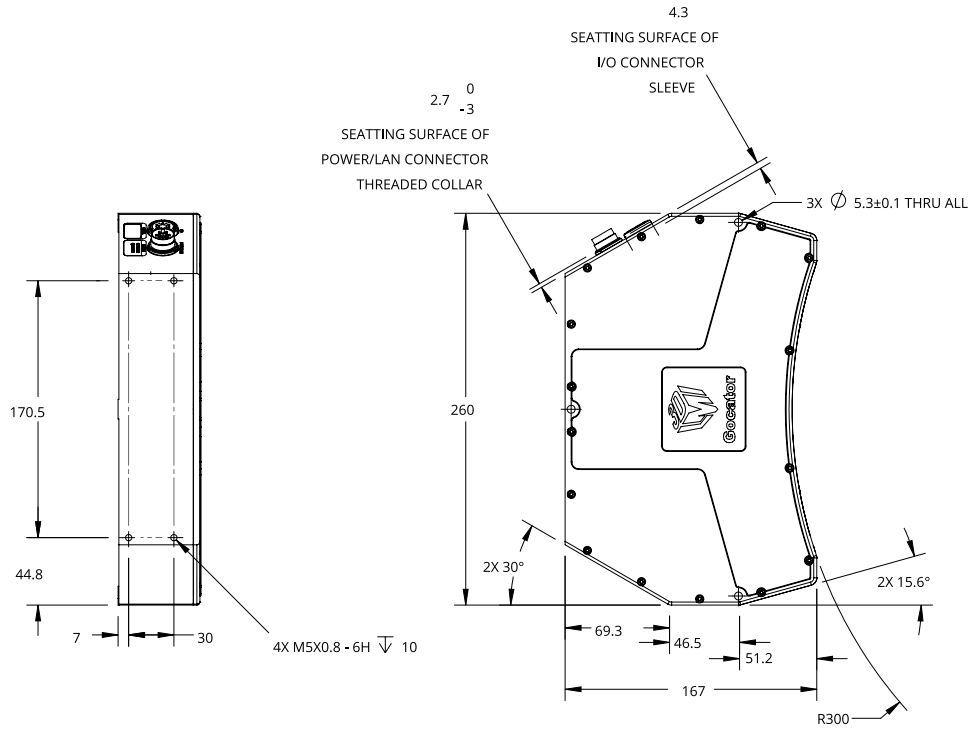
Gocator 3520

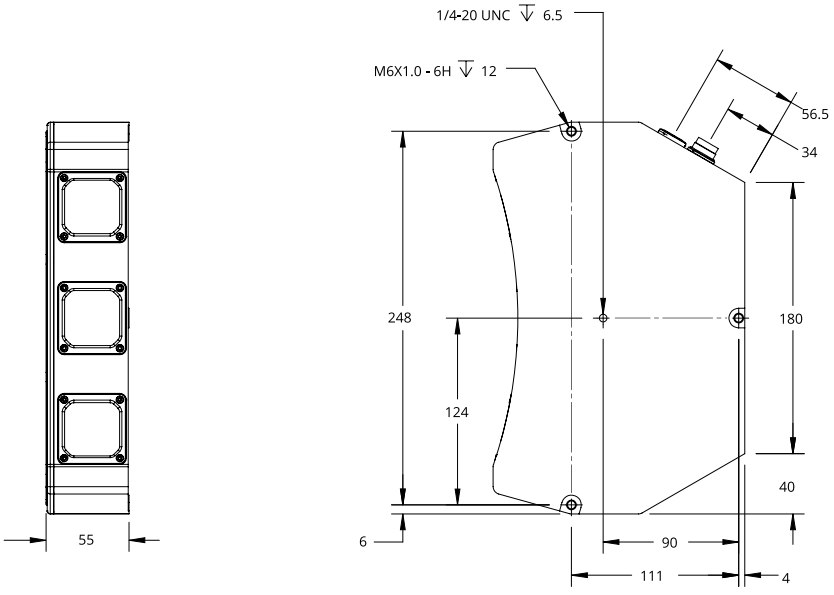
Field of View / Measurement Range / Coordinate System Orientation

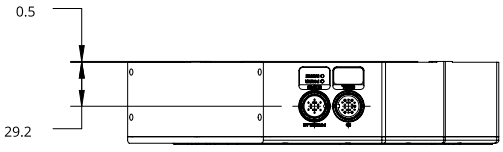




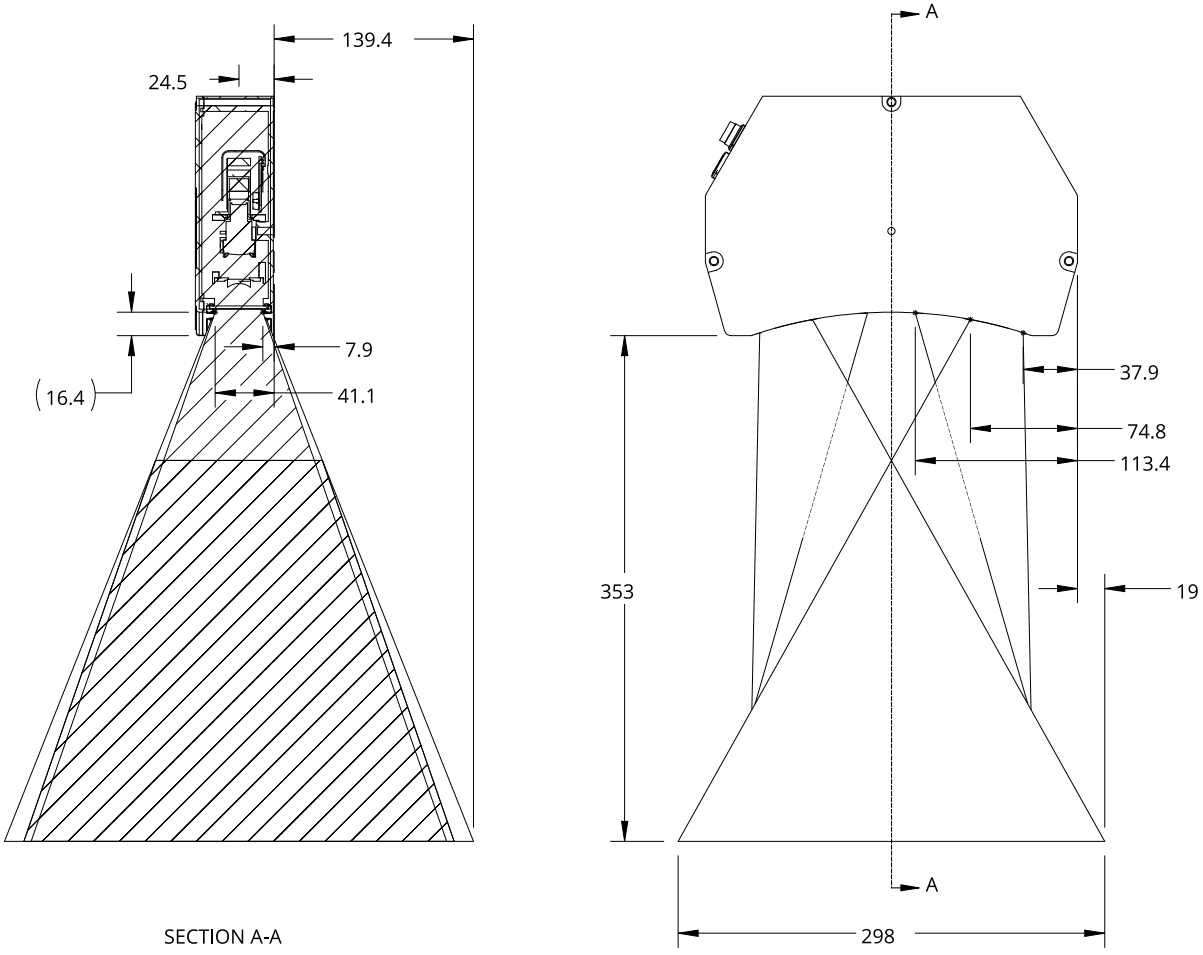
Dimensions


4.3
SEATTING SURFACE OF
I/O CONNECTOR
SLEEVE
2.7 0 -3
SEATTING SURFACE OF
POWER/LAN CONNECTOR
THREADED COLLAR
3X ∅ 5.3±0.1 THRU ALL
170.5
260
44.8
2X 30°
2X 15.6°
7
30
4X M5X0.8 - 6H ↧ 10
69.3
46.5
51.2
167
R300


1/4-20 UNC ↧ 6.5
M6X1.0 - 6H ↧ 12
56.5
34
248
180
124
40
55
6
90
111
4


0.5
29.2

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.

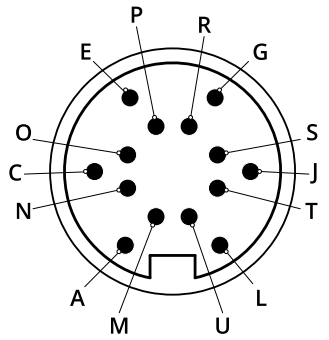
> This connector is rated IP67 only when a cable is connected or when the protective cap is in place.

> 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 1332.

This section defines the electrical specifications for Power/LAN Connector pins found on Gocator sensors, 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
Ethernet MX4-	U	Brown	Brown



*View: Looking into the connector **on** the sensor*

*Both wires must be connected to the power supply. The leads are connected to the same connector pin, and both are required to support the maximum possible current draw.

**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.

> It is not necessary to power down a sensor's power source such as a Master before unplugging the sensor from the Master. (Sensors can be "hot-swapped.")

> 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 1332.

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

> For maximum system reliability, use Phoenix Contact QUINT power supplies.

> For Gocator 3506 and 3210, operate sensors at 48 VDC for optimal performance.

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

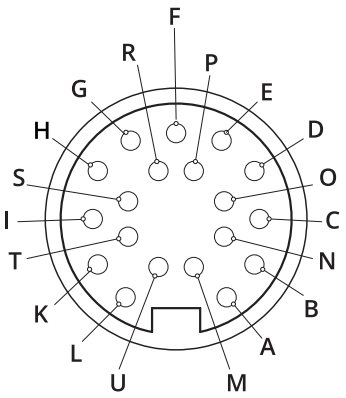
The Gocator I/O connector is a 19-pin, M16 style connector that provides encoder, digital input, digital outputs, serial output, and analog output signals.

> This connector is rated IP67 only when a cable is connected or when the protective cap is in place.

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+	P	Green	Gray / Red
Analog_out-	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.

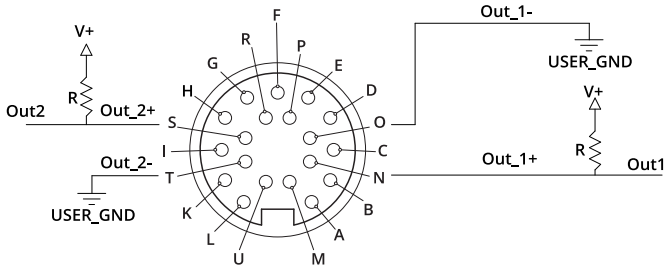
Digital Outputs

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

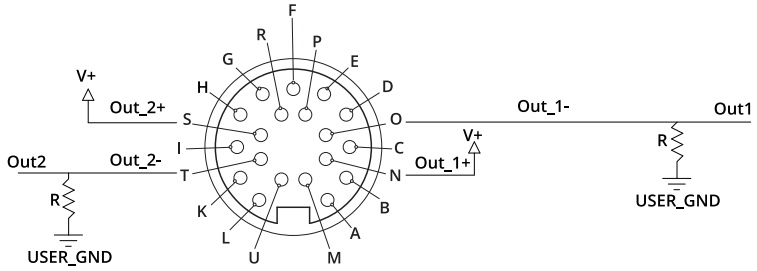


The resistors shown above are calculated by R = (V+) / 2.5 mA.

The size of the resistors is determined by power = (V+)^2 / R.

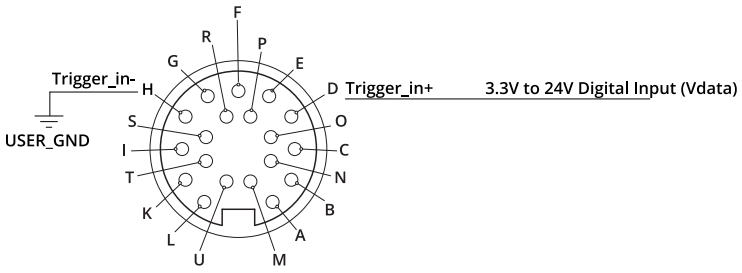
Inverting Outputs

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.



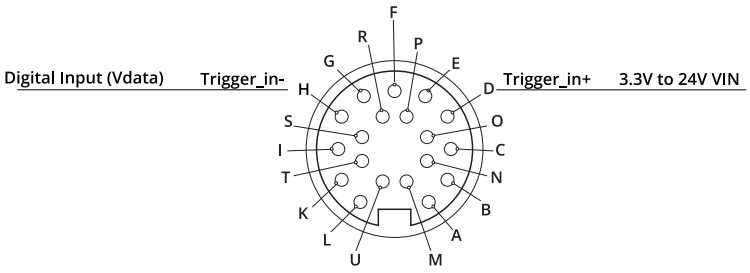
Digital Input

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.



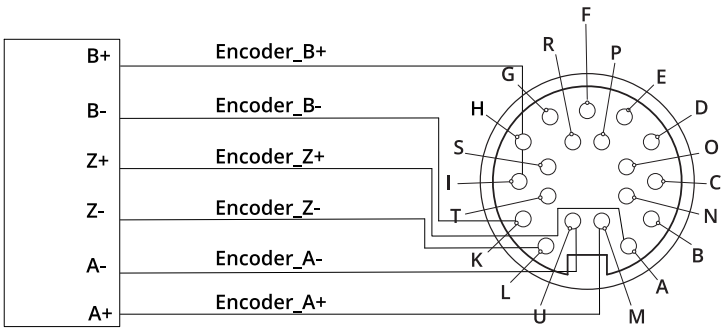
Active Low

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

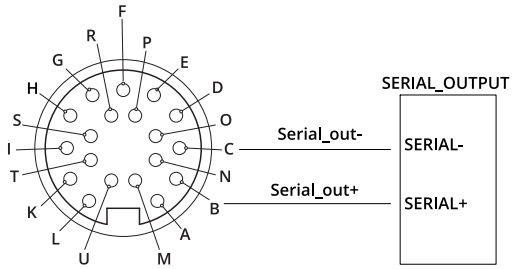
Gocator supports differential RS485 or RS422 signalling, as well as TTL. Both + and - signals must be connected.

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

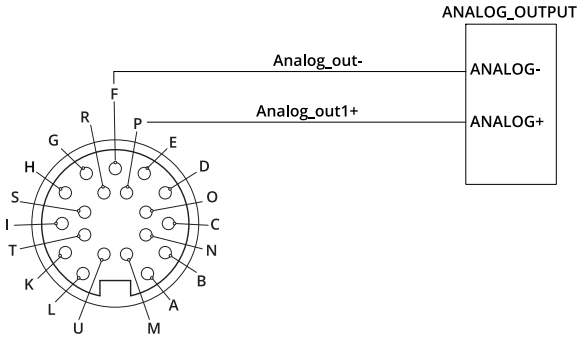


Analog Output

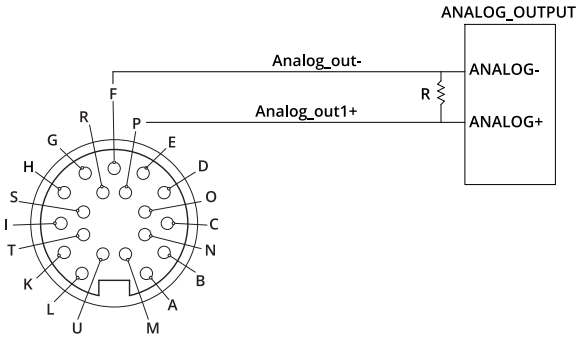
The Sensor I/O Connector defines one analog output interface: Analog_out.

You do not need to supply an external power source.

Function	Pins	Current Range
Analog_out	P, F	4 – 20 mA

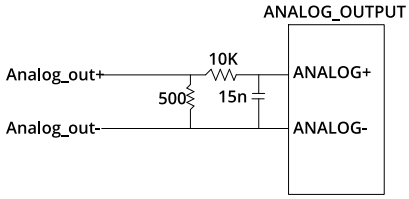


Current Mode



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.



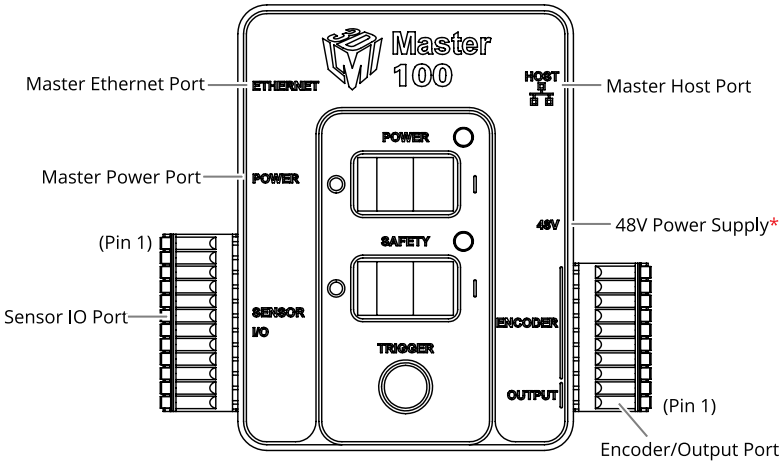
Master Network Controllers

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 144.

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	Pink / Red
Encoder_A-	2	Pink / Black
Encoder_Z+	3	White / Red
Encoder_Z-	4	White / Black
Trigger_in+	5	Blue / Red

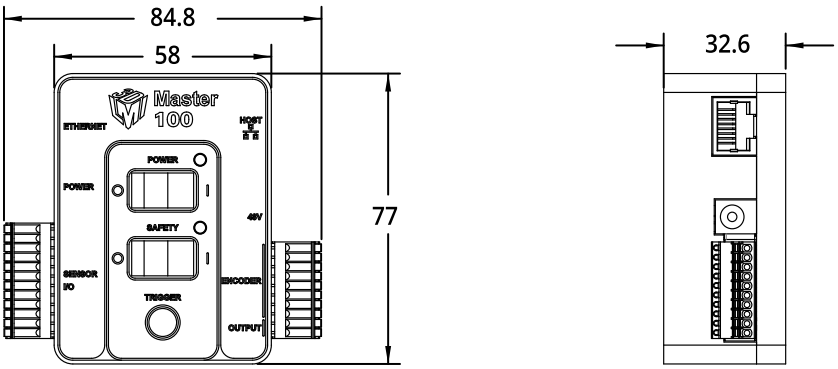
I/O Pin	Master Pin	Conductor Color
Trigger_in-	6	Blue / Black
Out_1-	7	Brown / Black
Out_1+	8	Brown / Red
Encoder_B+	11	Yellow / Red
Encoder_B-	12	Yellow / Black

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



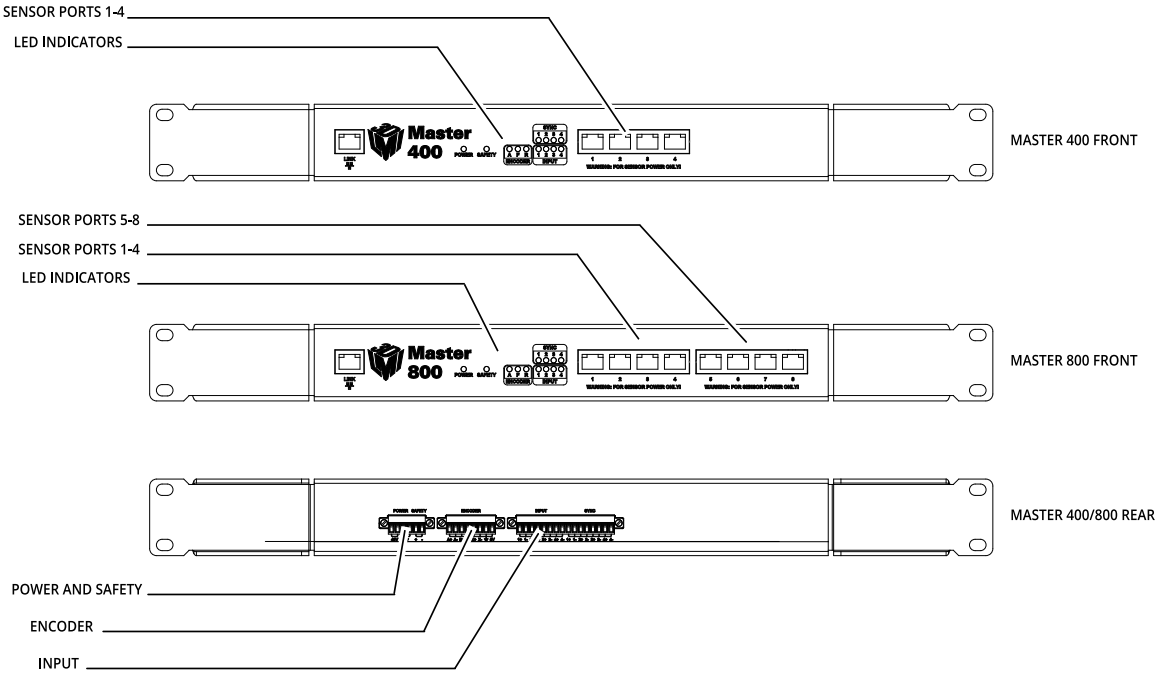
Master 400/800

Master network controllers provide sensor power and laser safety, 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.

> It is not necessary to power down a sensor's power source such as a Master before unplugging the sensor from the Master. (Sensors can be "hot-swapped.")

> Laser safety 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

Power and Safety (6 pin connector)

Function	Pin
+48VDC	1
+48VDC	2
GND (24-48VDC)	3
GND (24-48VDC)	4

Function	Pin
Reserved	5
Reserved	6

The 6-pin Power and Safety connector's specifications are as follows:

- Manufacturer PN: 1847097
- Phoenix Contact MC 1,5/ 6-STF-3,5

> The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

Encoder

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

The 8-pin Encoder connector's specifications are as follows:

- Manufacturer PN: 1847181
- Phoenix Contact MC 1,5/ 8-STF-3,5

Input

Input and Sync (16 pin connector)

Function	Pin
Input 1	1
Input 1 GND	2
Reserved	3
Reserved	4
Reserved	5
Reserved	6

Function	Pin
Reserved	7
Reserved	8
Reserved	9
Reserved	10
Reserved	11
Reserved	12
Reserved	13
Reserved	14
Reserved	15
Reserved	16

The 16-pin Input and Sync connector's specifications are as follows:

- Manufacturer PN: 1847262
- Phoenix Contact MC 1,5/ 16-STF-3,5

🗋	The Input and Sync connector does not need to be wired up for proper operation.

Master 400/800 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: +3.3 to +24 VDC

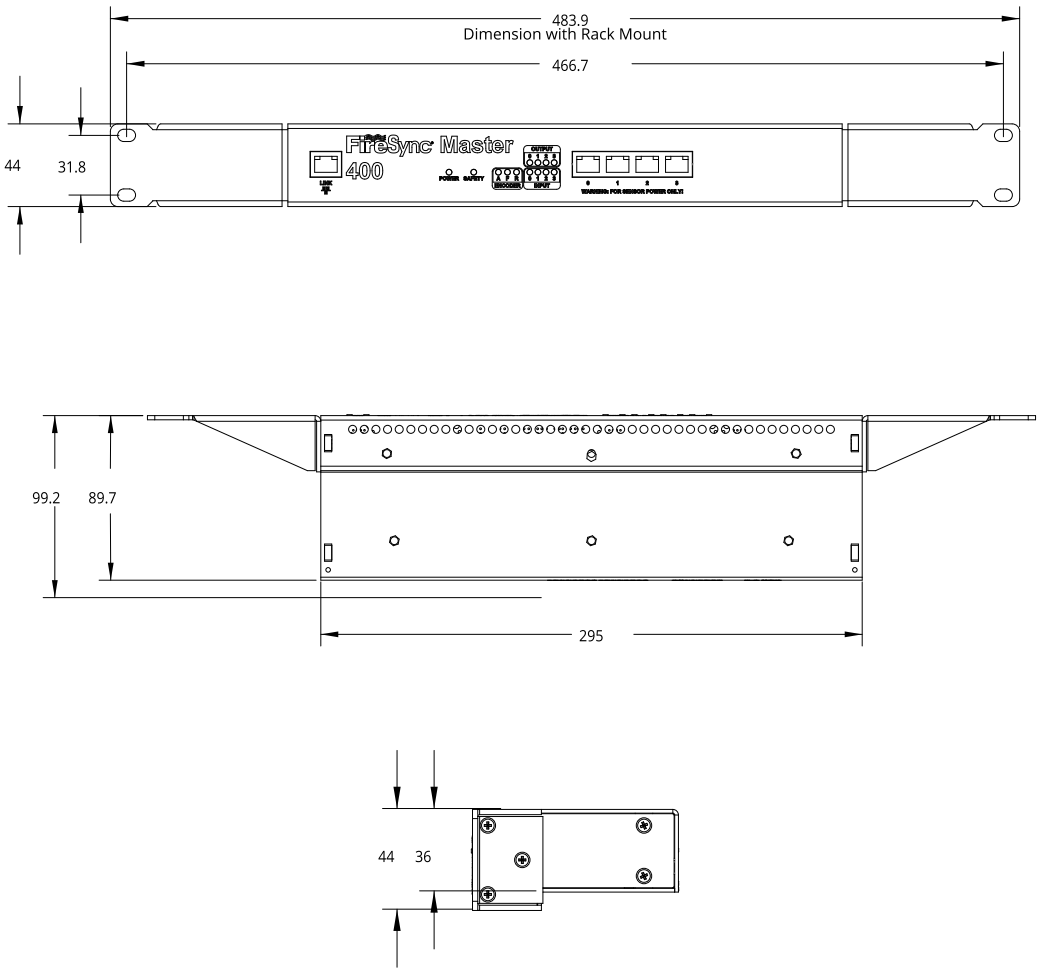
⚠	When using a Master hub, the chassis must be well grounded.

🗋	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 laser safety, 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.

> It is not necessary to power down a sensor's power source such as a Master before unplugging the sensor from the Master. (Sensors can be "hot-swapped.")

> Laser safety 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 1332.

The following table summarizes Master 810 and 2410:

Master 810 and 2410	
Input Voltage (Power)	+24-48 VDC (2 Watts); ripple ± 10%[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 1363.
Encoder	Differential (5 VDC, 12 VDC) Single-ended (5 VDC, 12 VDC)[3] For more information, see *Electrical Specifications* on page 1363.
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
Storage Temperature	-30 to 70 °C
Operation Temperature	0 to 50 °C

Notes

1. 48 VDC is recommended for multi-sensor systems and is required for G4000, G5000, and G200 systems. Refer to your sensor's "Input Voltage (Power)" specification on its datasheet to calculate the total power required.
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

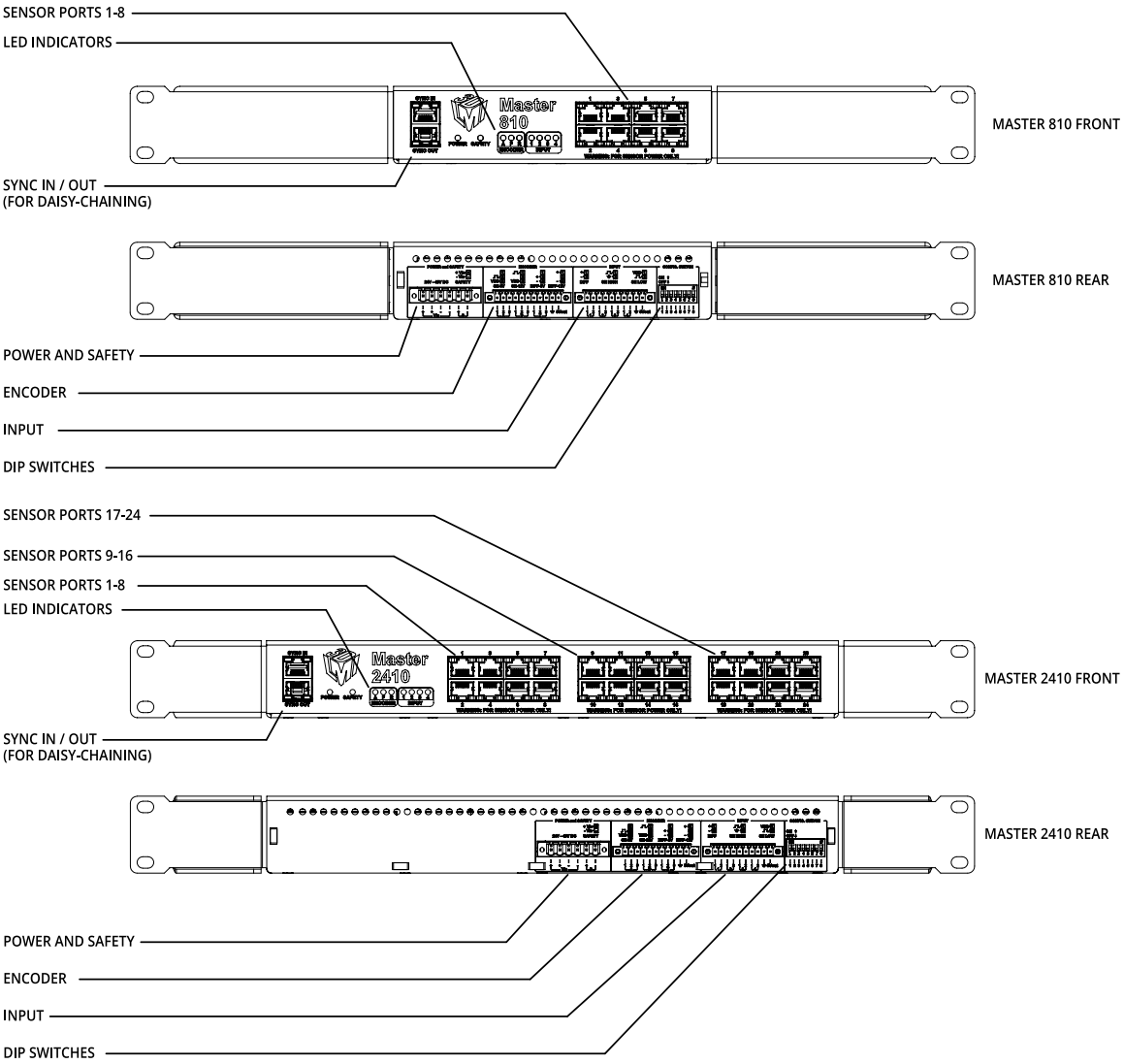
Indicator	Description
Power	Device is on.
Safety	Indicates the status of the Laser Safety circuitry. The “On” state indicates that all sensor light sources are active.
Encoder A	Reserved
Encoder F	**On continuously**: Forward motion with no indexing is detected. **Blinking**: Forward motion with indexing is detected.
Encoder R	**On continuously**: Reverse motion with no indexing is detected. **Blinking**: Reverse motion with indexing is detected.
Input 1-4	Digital input ports 1-4 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 36). 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 38.



Power and Safety

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 6-pin Power and Safety connector's specifications are as follows:

- CONNECTOR, 6 Position Terminal Block Plug, Female Sockets 0.200" (5.08mm) 180° Free Hanging (In-Line)
- Manufacturer PN: 1912223
- Supplier Part Number 277-11017-ND
- Manufacturer: Phoenix Contact

> The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

Encoder

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 page 1364.

The 11-pin Encoder connector's specifications are as follows:

- CONNECTOR, 11 Position Terminal Block Plug, Female Sockets 0.138" (3.50mm) 180° Free Hanging (In-Line)
- Manufacturer PN: 1847217
- Supplier Part Number 277-8897-ND
- Manufacturer: Phoenix Contact

Input

> 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

For Input connection wiring options, see *Input* on page 1366.

The following are the 10-pin Input connector's specifications:

- CONNECTOR, 10 Position Terminal Block Plug, Female Sockets 0.138" (3.50mm) 180° Free Hanging (In-Line)
- Manufacturer PN: 1847204
- Supplier Part Number 277-6350-ND
- Manufacturer: Phoenix Contact

The Input connector does not need to be wired up for proper operation.

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 1332

Electrical Specifications

Specification	Value
Power Supply Voltage	+24 VDC to +48 VDC; ripple ± 10% 48 VDC is recommended for multi-sensor systems and is required for G4000, G5000, and G200 systems. Refer to your sensor's "Input Voltage (Power)" specification on its datasheet to calculate the total power required.
Power Supply Current (Max.)*	Master 810: 9 A Master 2410: 25 A * Fully loaded with 1 A per sensor port.

Specification	Value
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* below.
Digital Input Voltage Range	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 *Input* on page 1366. 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 to 12 VDC, 12 to 24 VDC) and Differential (5 to 12 VDC, 12 to 24 VDC).

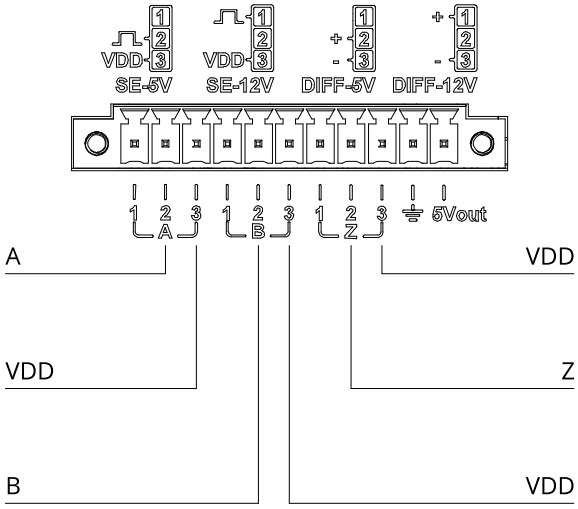
For 5 to 12 VDC operation, pins 2 and 3 of each channel are used.

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.

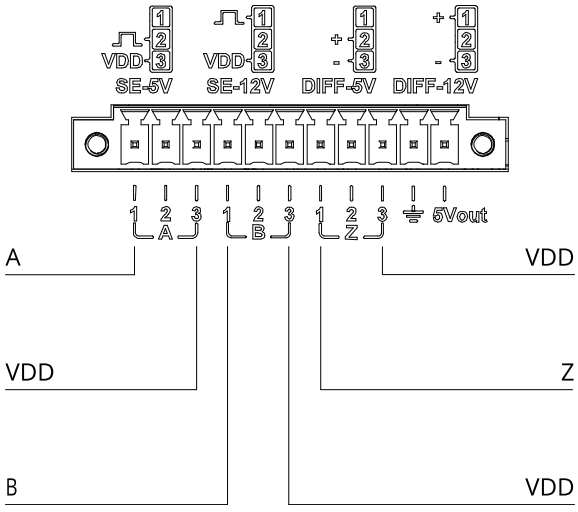
For 12 to 24 VDC operation, pins 1 and 3 of each channel are used.

> Although Master 810 and 2410 are currently labeled as only supporting up to 12 VDC, they can support up to 24 VDC.

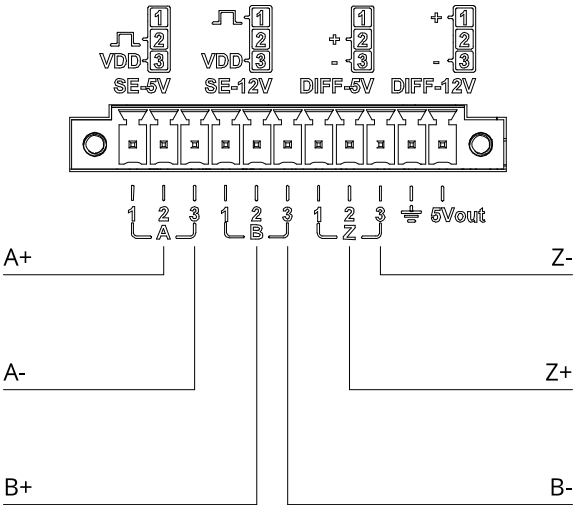
To determine how to wire a Master to an encoder, see the illustrations below.



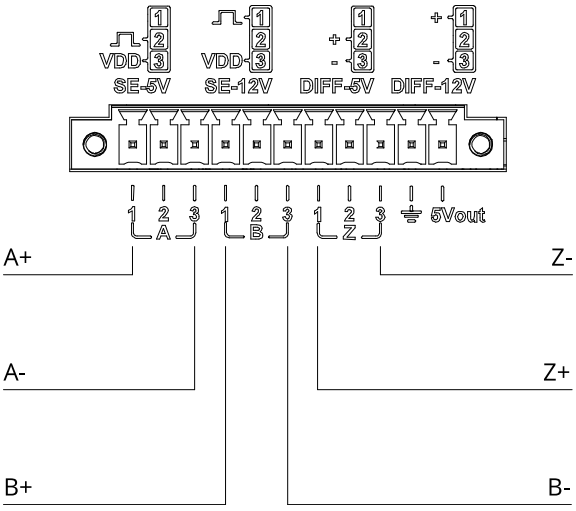
Single-Ended 5 VDC



Single-Ended 12 VDC



Differential 5 VDC



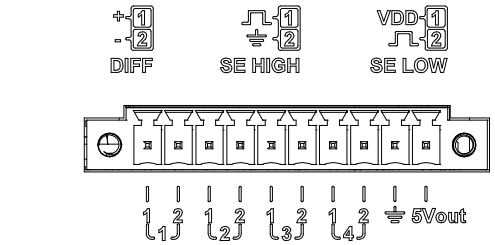
Differential 12 VDC

Input

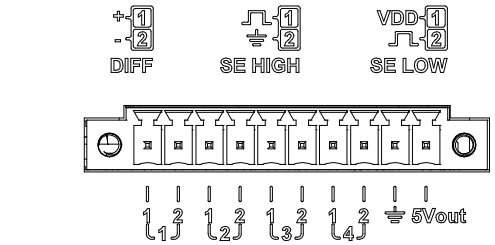
Master 810 and 2410 support the following types of input: Differential, Single-Ended High, and Single-Ended Low.

	Currently, Gocator only supports Input 0.

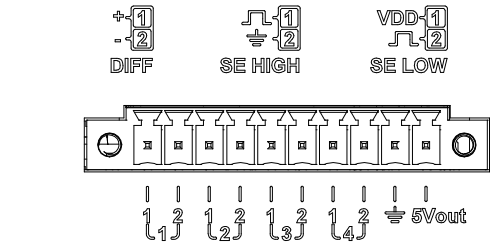
For digital input voltage ranges, see the table below.



Differential



Single-Ended Active High



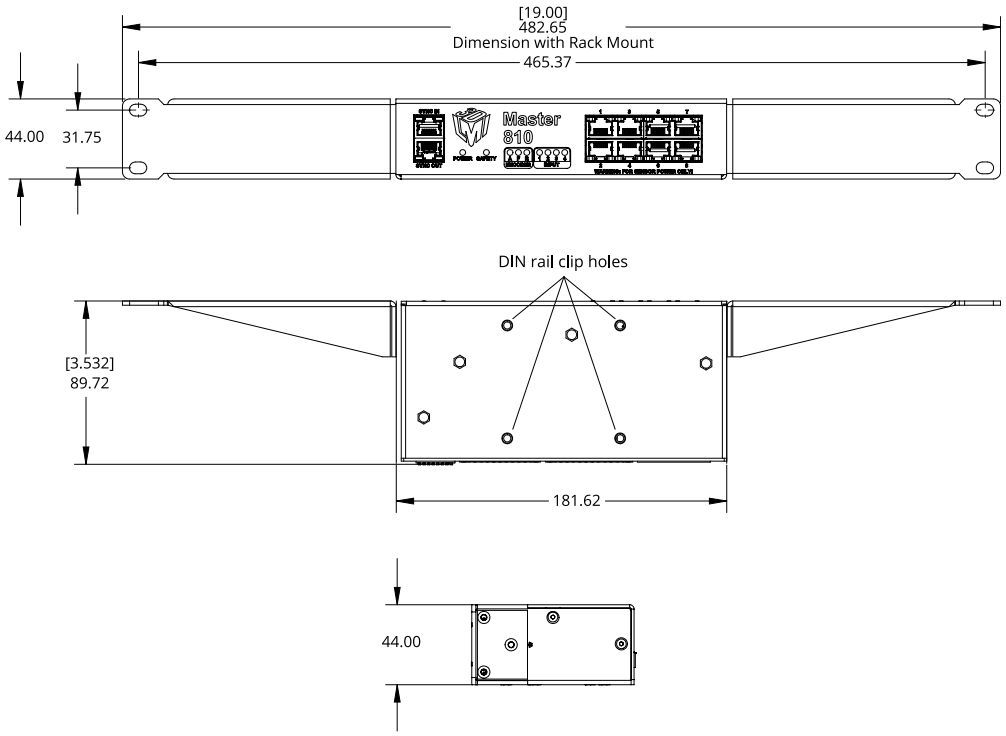
Single-Ended Active Low

Digital Input Voltage Ranges

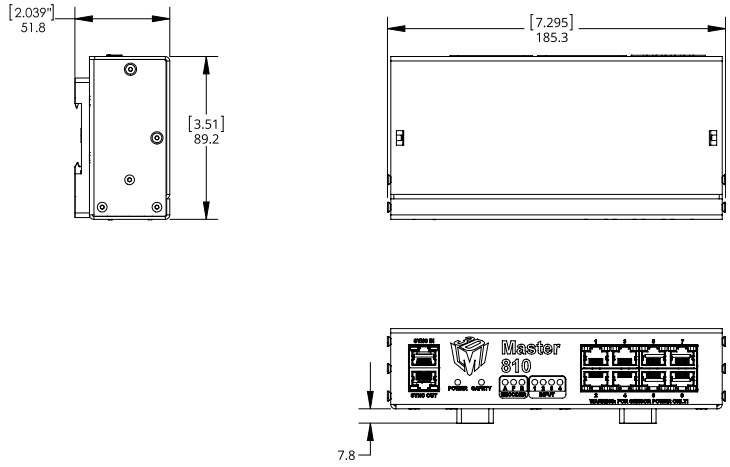
	Input Status	Min (VDC)	Max (VDC)
Single-ended Active High	Off	0	+0.8
	On	+3.3	+24
Single-ended Active Low	Off	(V_{DD} - 0.8)	V_{DD}
	On	0	(V_{DD} - 3.3)
Differential	Off	-24	+0.8
	On	+3.3	+24

Master 810 Dimensions

With 1U rack mount brackets:



With DIN rail mount clips:



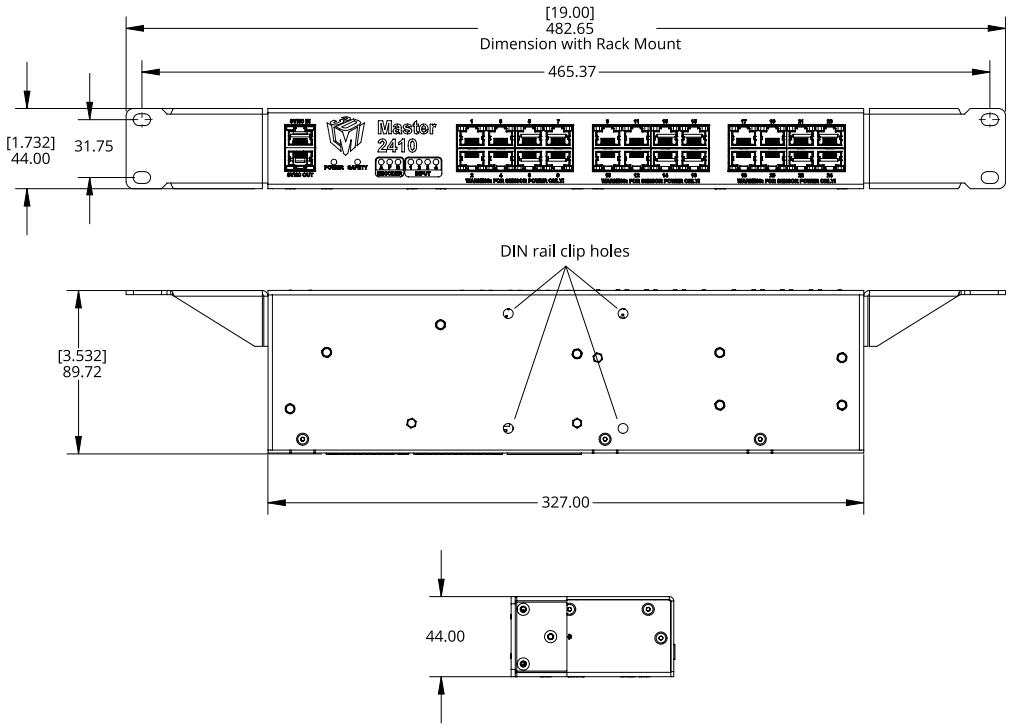
Older revisions of Master 810 and 2410 network controllers use a different configuration for the DIN rail clip holes.

For information on installing DIN rail clips, see *Installing DIN Rail Clips: Master 810 or 2410* on page 36.

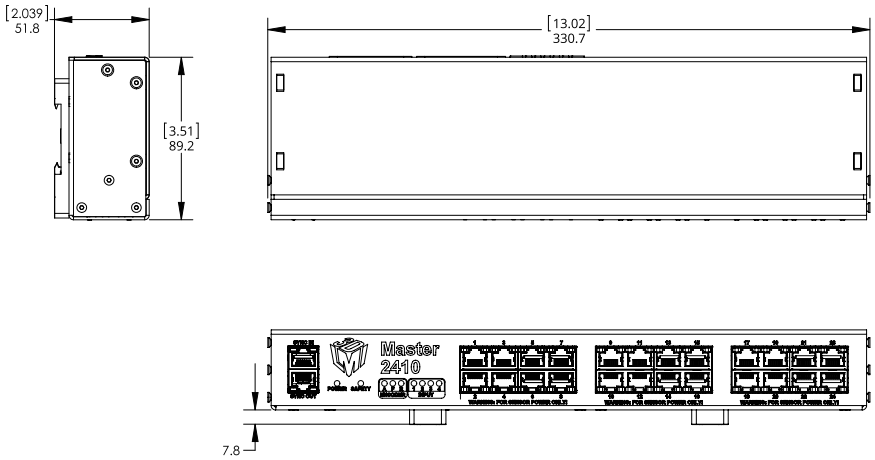
The CAD model of the DIN rail clip is available at https://www.winford.com/products/cad/dinm12-rc.igs.

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 36.

The CAD model of the DIN rail clip is available at https://www.winford.com/products/cad/dinm12-rc.igs.

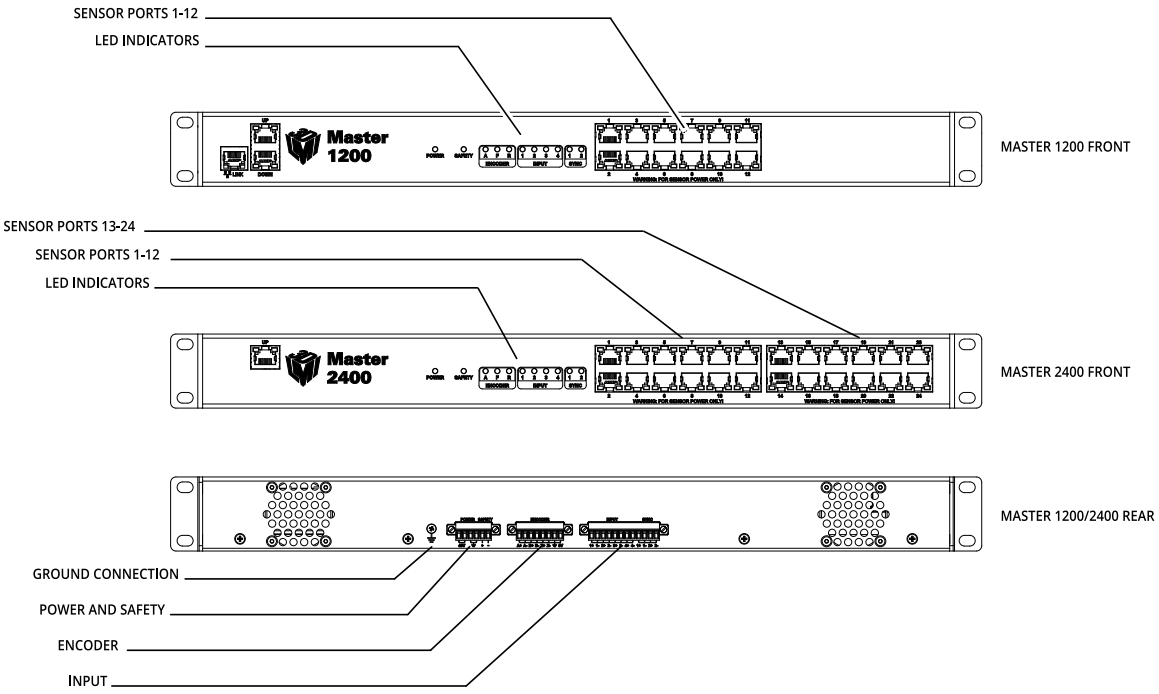
Master 1200/2400

Master network controllers provide sensor power and laser safety, 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.

> It is not necessary to power down a sensor's power source such as a Master before unplugging the sensor from the Master. (Sensors can be "hot-swapped.")

> Laser safety 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

Power and Safety (6 pin connector)

Function	Pin
+48VDC	1
+48VDC	2
GND (24-48VDC)	3
GND (24-48VDC)	4

Function	Pin
Reserved	5
Reserved	6

The 6-pin Power and Safety connector's specifications are as follows:

- Manufacturer PN: 1847097
- Phoenix Contact MC 1,5/ 6-STF-3,5

> The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

Encoder

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

The 8-pin Encoder connector's specifications are as follows:

- Manufacturer PN: 1847181
- Phoenix Contact MC 1,5/ 8-STF-3,5

Input

Input and Sync (12 pin connector)

Function	Pin
Input 1	1
Input 1 GND	2
Reserved	3
Reserved	4
Reserved	5
Reserved	6

Function	Pin
Reserved	7
Reserved	8
Reserved	9
Reserved	10
Reserved	11
Reserved	12

The 12-pin Input and Sync connector's specifications are as follows:

- Manufacturer PN: 1847220
- Phoenix Contact MC 1,5/ 12-STF-3,5

> The Input and Sync connector does not need to be wired up for proper operation.

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

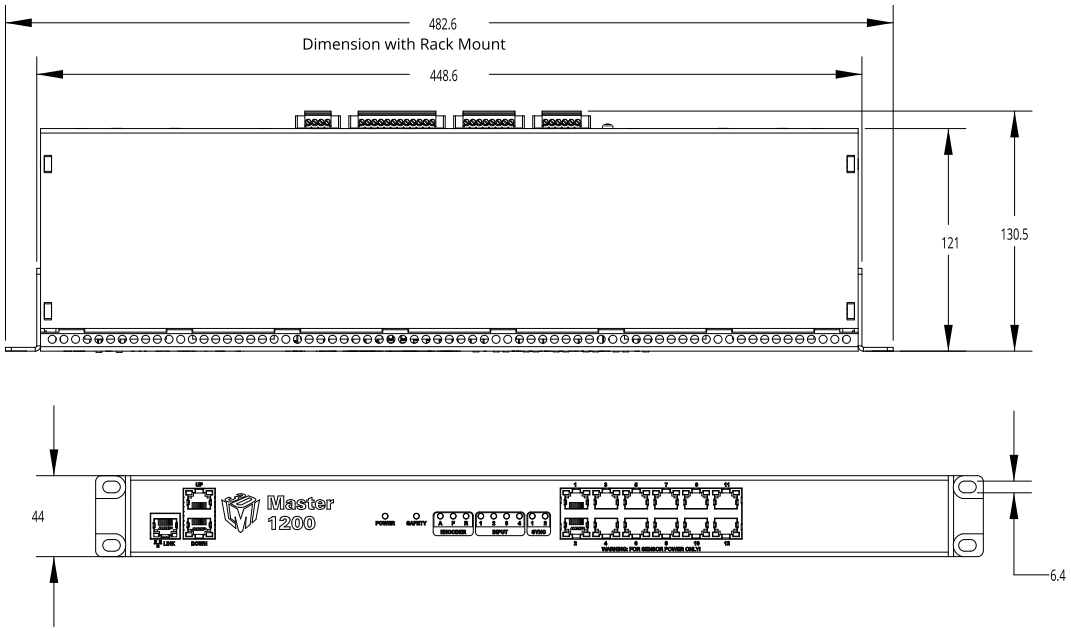
> ⚠ When using a Master hub, the chassis must be well grounded.

> 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.


482.6
Dimension with Rack Mount
448.6
121
130.5
44
Master
1200
6.4

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

High Flex Gocator Cordsets - Straight Connectors

Description	Part Number
1.2m I/O cordset, open wire end	301175-1.2m
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
15m I/O cordset, open wire end	301175-15m
20m I/O cordset, open wire end	301175-20m
25m I/O cordset, open wire end	301175-25m
Custom length (< 25m) I/O cordset, open wire end	301175
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
15m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	301176-15m
20m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	301176-20m
25m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	301176-25m
Custom length (< 25m) Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	301176
1.2m Power and Ethernet to Master cordset, 2x RJ45 ends	301165-1.2m
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
15m Power and Ethernet to Master cordset, 2x RJ45 ends	301165-15m
20m Power and Ethernet to Master cordset, 2x RJ45 ends	301165-20m
25m Power and Ethernet to Master cordset, 2x RJ45 ends	301165-25m
Custom length (< 25m) Power and Ethernet to Master cordset, 2x RJ45 ends	301165

High Flex Gocator Cordsets - 90-degree Connectors

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
15m I/O cordset, 90-deg, open wire end	301172-15m
20m I/O cordset, 90-deg, open wire end	301172-20m
25m I/O cordset, 90-deg, open wire end	301172-25m
Custom length (< 25m) I/O cordset, 90-deg, open wire end	301172
2m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	301171-2m
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
15m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	301171-15m
20m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	301171-20m
25m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	301171-25m
Custom length (<25m) Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	301171
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
15m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	301173-15m
20m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	301173-20m
25m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	301173-25m

Notes related to cordsets

For information on cordset bend radius limits, see *Cordset Bend Radius Limits* on page 32.

Custom cordset lengths between 25 m and 60 m (maximum) and connector orientations are available upon request. Prices depend on length and orientation requested. Extension cordsets with one male and one female Gocator connector are also available on request.

90-degree connectors can be adjusted one of four different orientations by opening the connector.

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 1377.

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 *GoPxL Discovery Tool* on page 1290 for more information.

When attempting to log in, the password is not accepted.

- Use the Sensor Recovery tool. See *GoPxL Discovery Tool* on page 1290 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 141 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.
  For more information on configuring resolution, see *Spacing on X and Y (Uniform Spacing)* on page 140 and *Triggers* on page 141.
- Review the measurements that you have programmed and eliminate any unnecessary measurements.

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

YOLOv9

Website:

https://github.com/WongKinYiu/yolov9

License:

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2024 Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao

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 3 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, see https://www.gnu.org/licenses/.

CLI11

Website:

https://github.com/CLIUtils/CLI11

License:

CLI11 1.8 Copyright (c) 2017-2019 University of Cincinnati, developed by Henry Schreiner under NSF AWARD 1414736.

All rights reserved. Redistribution and use in source and binary forms of CLI11, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. 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.

3. Neither the name of the copyright holder 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 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.

xxhash

Website:

https://github.com/Cyan4973/xxHash

License:

xxHash Library

Copyright (c) 2012-present, Yann Collet

All rights reserved.

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:

MIT License

Copyright (c) 2013-2019 Niels Lohmann

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.

OpENer

Website:

https://github.com/EIPStackGroup/OpENer

License:

SOFTWARE DISTRIBUTION LICENSE FOR THE

ETHERNET/IP(TM) COMMUNICATION STACK

(ADAPTED BSD STYLE LICENSE)

Copyright (c) 2009, Rockwell Automation, Inc. ALL RIGHTS RESERVED. EtherNet/IP is a trademark of ODVA, Inc.

Redistribution of the Communications Stack Software for EtherNet/IP 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 and trademark notices, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

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 Rockwell Automation, ODVA, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission from the respective owners.

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 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.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

mobx

Website:

https://mobx.js.org/README.html

ramda

Website:

https://ramdajs.com/

License:

The MIT License (MIT)

Copyright (c) 2013-2019 Scott Sauyet and Michael Hurley

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.

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

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.

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

LMI Technologies has sales offices and distributors worldwide. All contact information is listed at lmi3D.com/contact.