

# Locator Studio

## User Manual

Version 1.0

# Content

Content	2
Preface	4
Purpose of the manual	4
Trademarks	4
About Locator Studio	4
Robot compatibility	5
Scope of delivery	6
About this User Manual	7
Installation	8
Hardware specification	8
Compliance with Standards	8
Vision Controller overview	8
Connection of peripherals	9
First boot of Locator Studio	10
User Interface Overview	13
Network setup	15
Standard topologies	15
Interfaces configuration	16
Network page	16
Robot interface	17
Vision controller Interface for 3D Sensor	18
3D Sensor network configuration	19
Valid network configurations	20
Connection to Local Network	21
Robot module setup	23
Robotic brands with native support	23
Robot modules architecture	24
Custom Integrations	27
Solution management	27
Solution configuration	29
Solution	29
Picked object	30
Vision System & Localization	31

3D Sensor and scanning parameters of the Vision system	32
Object and localization parameters of the Vision system	34
Calibration matrix of the Vision system	36
Settings of the Vision system	36
Calibration (Robot-Camera)	36
Types of calibration	37
Requirements and setup	39
Manual and Semi-automatic calibration	41
Calibration procedure	42
Tips and Troubleshooting	46
Settings (Solution-level)	47
Deployment	48
Remote access	50
VNC	50
TeamViewer	51
Web browser on another computer	52
Remote support	54

# Preface

Thank you for choosing the Locator Studio. Please, take a few minutes to read this manual and become familiar with the hardware and software.

For more information on our products, accessories, replacement parts, software and services, see the website [www.photoneo.com](http://www.photoneo.com) or contact our team at [support@photoneo.com](mailto:support@photoneo.com).

## Purpose of the manual

This manual provides information about installing and setting up the Locator Studio. It is designed for engineers, programmers, installers, and electricians with a general knowledge of automation and robotics.

## Trademarks

All names identified by ® are registered trademarks of Photoneo s.r.o. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

# About Locator Studio

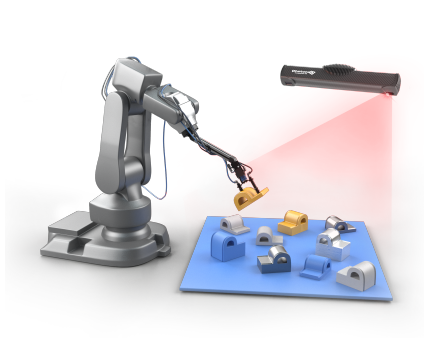
**Locator Studio** is a set of Photoneo 3D Sensor(s) and Vision Controller with pre-installed software that provides:

- PhoXi Control for configuration of the Photoneo 3D Sensor
- Easy solution configuration via web-based interface
  - Robot-Camera Calibration
  - CAD-based object localization for finding object position and orientation (6-DOF pose) in the scene captured by 3D sensor
  - TCP/IP interface for communication with an arbitrary robotic controller, PC, or PLC

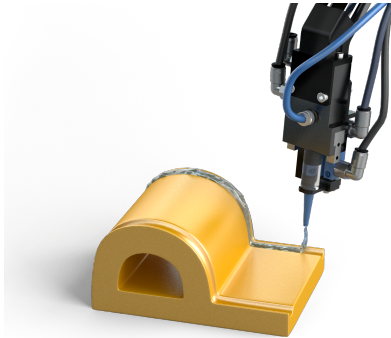
Locator Studio is ideal for vision-guided robotic applications based on referenced position: picking objects from a tray, conveyor belt, rack, or to locate components, fixtures for precise placing,



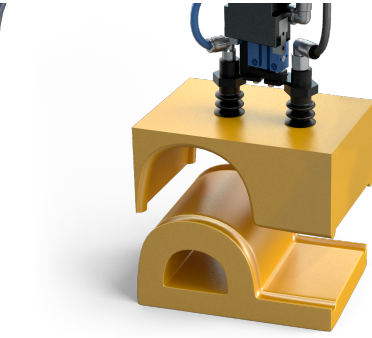
screwing, and assembly. It serves well for picking parts from a non-collision environment, handling structured and semi-oriented parts, or de-racking applications.



Localization of multiple objects at the time for robotic picking



Trajectory correction based on object position



Assembly of objects based on object position

## Robot compatibility

Locator Studio provides TCP/IP interface for binary encoded exchange of 6-DOF pose data into a robotic controller, PC, or PLC. This universal interface allows the connection of practically any robot manipulator (6-axis, delta, scara). Locator Studio can report the position and orientation of an object in the native rotation formalism of the robot (applies to robot modules developed by Photoneo). Sample robot programs that implement the interface can be found on the [Photoneo website](https://www.photoneo.com).



6-axis robot arm



Delta robots



Palletizing robots



Scara robots

Selected robotic examples are provided for ABB, Fanuc, Kawasaki, and Universal Robots. For other clients (robot controller, PC, PLC), the implementation of communication needs to be developed based on the provided specification for request/response protocol.

## Scope of delivery

Photoneo 3D Sensor with accessories\*



Vision Controller with pre-installed software and license



Power adapter



Power cable (with country-specific plug type)



Power connector



Power DC Cables



Calibration object



Marker pattern



*\*See the scope delivery in the [Photoneo 3D Sensor Quick Start Guide](#) to see what comes with the Photoneo 3D Sensor*

# About this User Manual

This user manual provides guidance on how to configure a Locator Studio divided into following chapters:

Configuration Step	Related Chapters
Physical installation of the Vision Controller and setting up a network connection	- <a href="#">Installation</a> - <a href="#">Network setup</a>
Deployment of the robot program	- <a href="#">Robot module</a>
Configuration of the Solutions containing all information about the 3D sensor, object and localization parameters	- <a href="#">Solution management</a> - <a href="#">Solution configuration</a>
Running a solution and status monitoring	- <a href="#">Deployment</a>

Supportive chapters:

- [Remote access](#)
- [Remote support](#)

# Installation

## Hardware specification

Please consult user manuals for the vision controller and Photoneo 3D Sensors:

- Vision Controller  
<https://www.neousys-tech.com/en/support-service/resources/category/234-nuvo-7160gc-user-manu>
- Photoneo 3D Sensors  
<https://www.photoneo.com/downloads/device-resources/>

## Compliance with Standards

Locator Studio and Photoneo 3D Sensors conform to the following standards and test specifications. Please note that the certification status may change without notification. Consult your local Photoneo representative if you need additional information related to the latest listing of exact approvals.







Vision Controller Nuvo-7000 Series satisfies requirements and safety-related objectives according to the EC directives. This CE mark is supported by tests conducted by the manufacturer.

Photoneo 3D Sensors satisfy requirements and safety-related objectives according to the EC directives listed below. This CE mark is supported by tests conducted by the manufacturer.

## Vision Controller overview

The image below offers an overview of the port selection on the Vision Controller and it highlights the supported/unsupported ports.

-  Supported display ports
-  Unsupported display ports (must not be used)
-  Ethernet ports (from left to right - Robot, Scanner, and Network)\*
-  Power socket

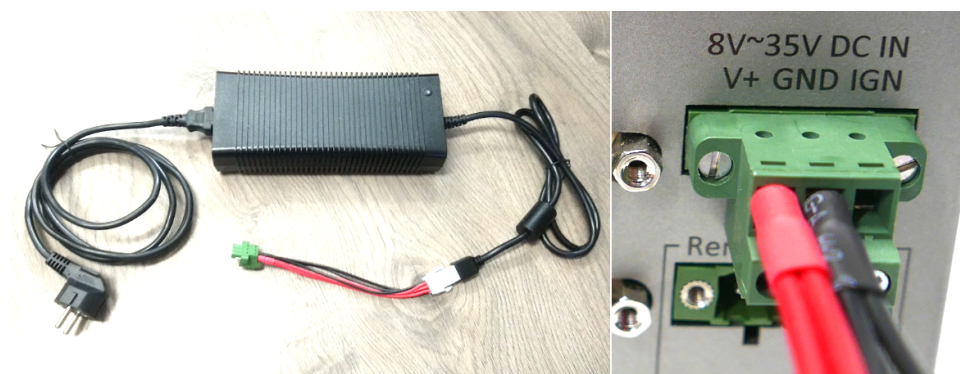
*\*Other ports must not be used*



## Connection of peripherals

### Power

Connect the power cable to the power adapter and the power adapter to the DC power cables. Install the power connector (red cable - left socket, black cable - middle socket). Make sure the screws are tightened. Finally, plug the power connector inside the power socket on the VC and secure it with screws.



*Note: The vision controller is configured to turn on automatically when the power source is connected.*

### Mouse and keyboard

Use any USB ports to connect a mouse and keyboard to the Vision Controller.

## Display

The vision controller is equipped both with an integrated graphics card by Intel and a dedicated GPU by NVIDIA.

The vision controller makes use of the NVIDIA graphics card and thus its GPU ports must be used to connect a display. These ports are located on the backside of the vision controller (where the power input is). There are 3 available types of display connection:

- DisplayPort
- HDMI
- DVI

The usage of multiple displays and touchscreen displays is also supported.

*Note: Onscreen keyboard can be enabled in the Ubuntu settings. Open the Settings panel, navigate to the category Universal access, and in the section Typing enable the Screen Keyboard. The keyboard will appear every time a text field is selected.*

It is also possible to operate the Studio remotely without the use of a display. Please refer to the [Remote access](#) to find out more about controlling the Studio remotely.

*Note: When using a display make sure it is connected before the vision controller is turned on. When no display is connected upon boot a virtual display is used for remote access which means the physical display will not work after being connected until the vision controller is rebooted.*

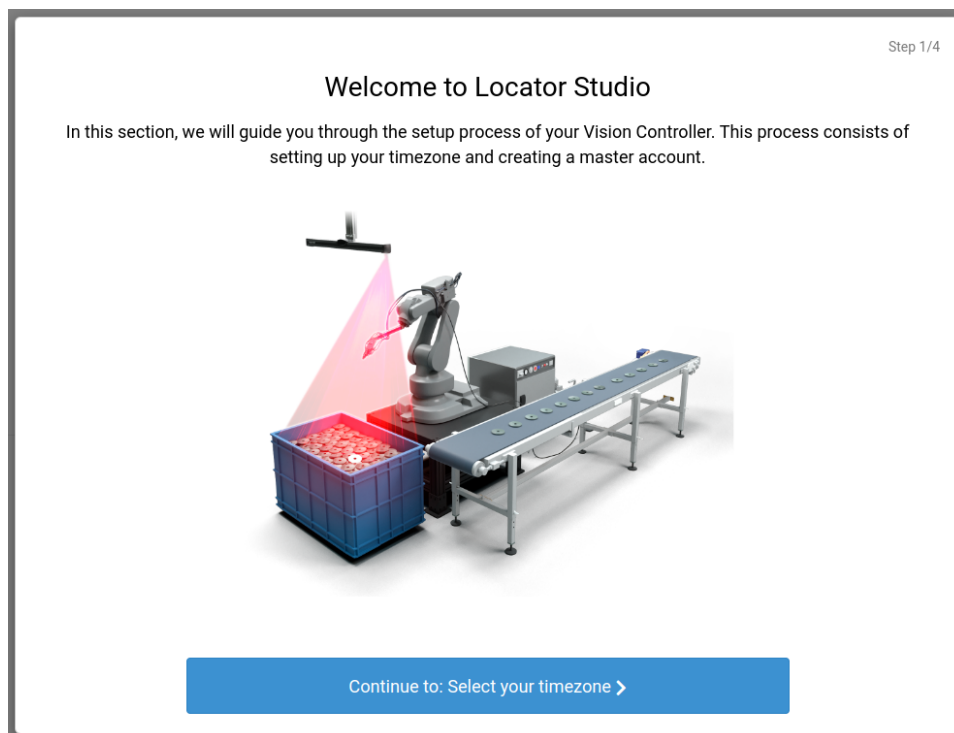
## Photoneo 3D Sensor, robot, and network

The Ethernet cables for the connection of the 3D Sensor, robot, and network must be plugged into the dedicated Ethernet ports. Please follow the instructions in the [Network setup](#).

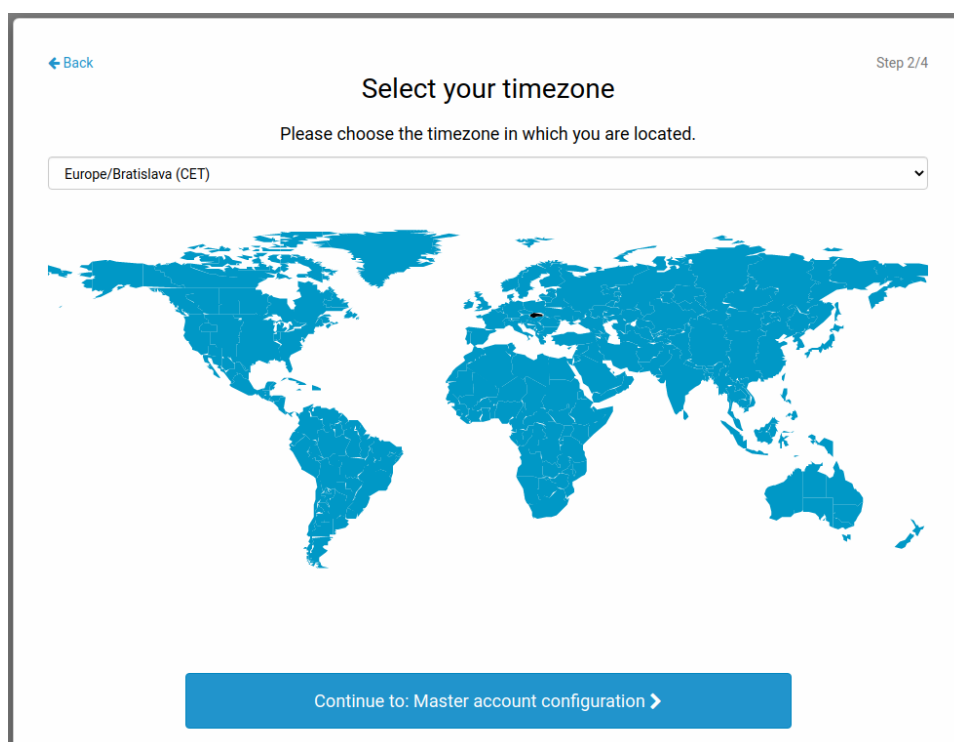
## First boot of Locator Studio

When all your accessories are connected to the Vision Controller and the power, you may turn on your Vision Controller. When the Vision Controller is booted up, the Locator Studio application will launch automatically.

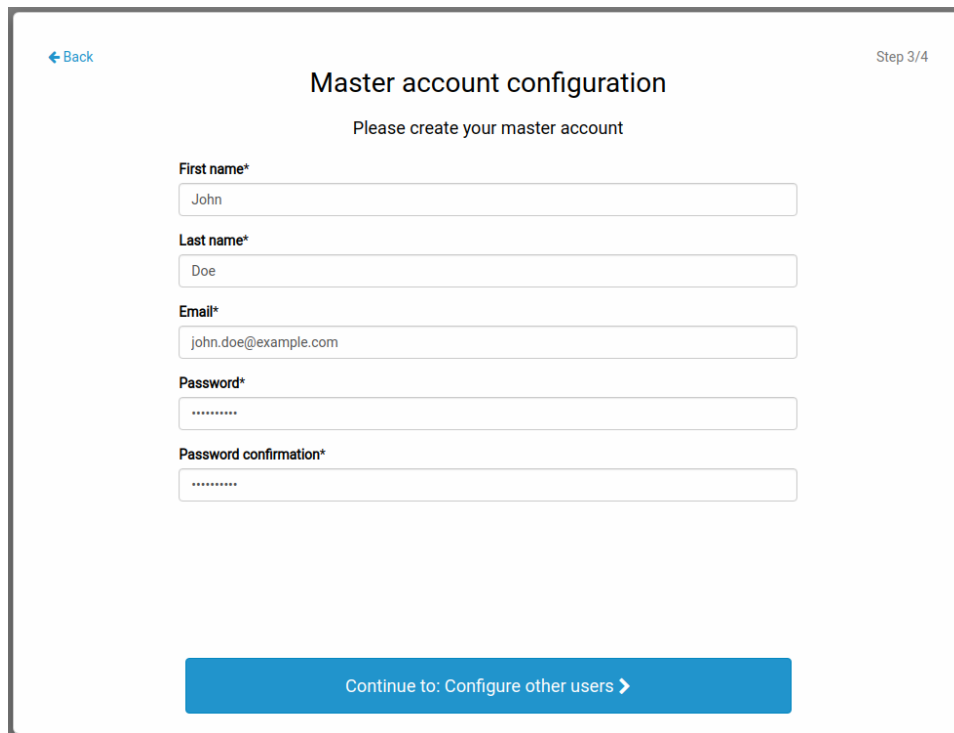
When Locator Studio runs for the first time, a **Welcome Page** appears, which will guide you through the initial setup of LS.



Firstly you will be asked to select your local timezone, either by clicking on the map or by selecting it from the list.



After selecting your timezone, you will be prompted to create a Master account that on top of the configuration of solutions allows regular user management.



The screenshot shows a web form titled "Master account configuration" with a subtitle "Please create your master account". The form is part of a 4-step process, currently on Step 3/4. It includes a "Back" link in the top left. The form fields are: "First name\*" with the value "John", "Last name\*" with the value "Doe", "Email\*" with the value "john.doe@example.com", "Password\*" with masked characters "\*\*\*\*\*", and "Password confirmation\*" with masked characters "\*\*\*\*\*". A blue button at the bottom says "Continue to: Configure other users >".

← Back

Step 3/4

### Master account configuration

Please create your master account

**First name\***

**Last name\***

**Email\***

**Password\***

**Password confirmation\***

Continue to: Configure other users >

The next step allows the creation of basic user accounts with access to the configuration of solutions only. This step can be skipped, user accounts can be added later via Account management.



← Back Step 4/4

## Configure other users

On this page, you may create employee accounts.

Amelie Poulein ✕

Victor Hugo ✕

[+ Add a new user](#)

**First name\***

**Last name\***

**Email\***

**Password\***

**Password confirmation\***

[Continue to: Finish >](#)

After finishing this initial setup, Locator Studio will have one Master account and an arbitrary number of regular user accounts.

## User Interface Overview

User interface is divided into the following pages accessible via the top navigation bar.

### Solutions

Management of solutions that contain the complete configuration of the application.

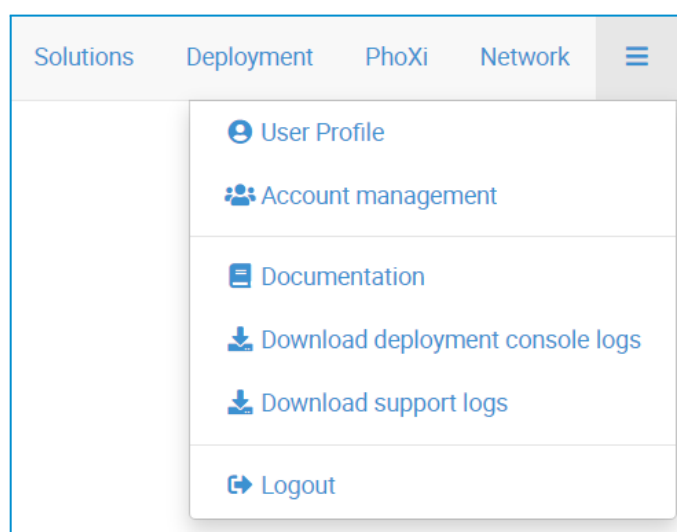
### Deployment

Starting, stopping, and monitoring of completed solutions.

### PhoXi

Viewer of data from the Photoneo 3D Sensor.

### Network



Configuration of connection to the 3D Sensor and robot controller.

### **System menu**

User management, documentation, and diagnostics.

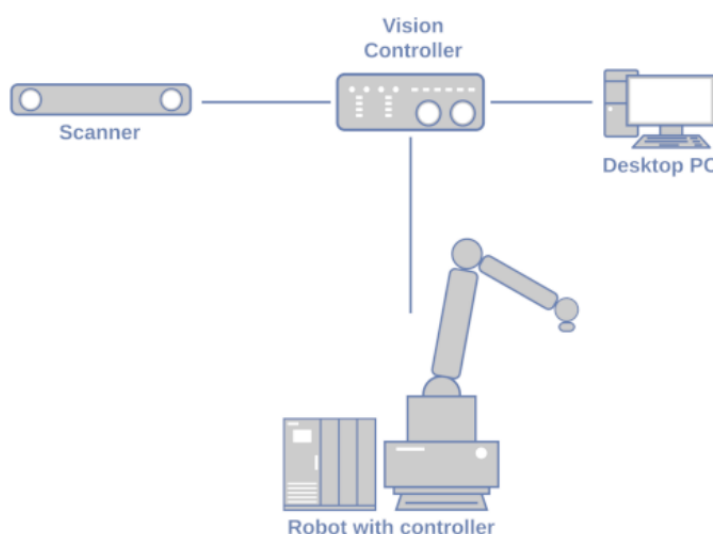
# Network setup

The standard robotic application consists of a robot with a robotic controller and a Photoneo 3D Sensor with the vision controller. A computer with a web browser is used for the configuration of the vision system.

## Standard topologies

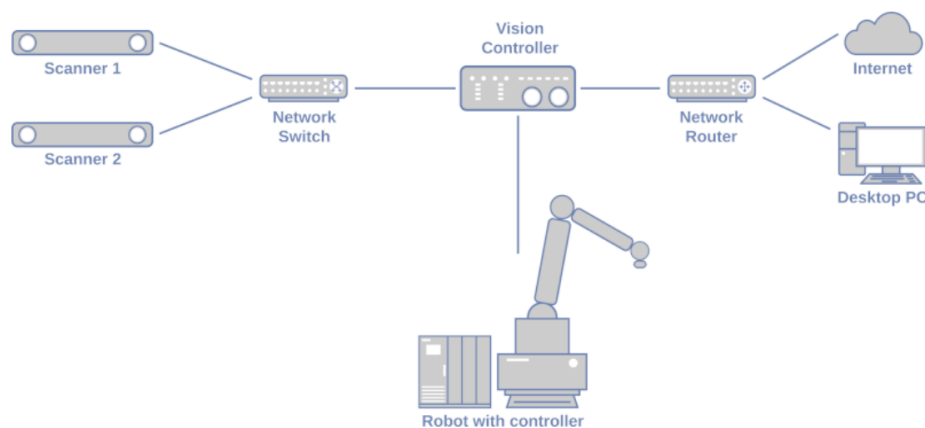
### Basic connection scheme

- A single Photoneo 3D Sensor is connected directly to the **Scanner** port of the Vision Controller.
- A Desktop PC for remote control is connected directly to the Network port of the Vision Controller.
- The Robot controller is connected directly to the Robot port of the Vision Controller.



### Multiple 3D Sensors and Internet access

- Two Photoneo 3D Sensors and the Vision Controller are connected through a Gigabit Network switch (the **scanner** port of the Vision Controller is used).
- The Vision Controller is connected to the local network (LAN) through a Network router. The Vision Controller has access to the Internet (if available on the LAN) and other devices on the LAN have access to the Vision Controller (remote control).
- The Robot controller is connected directly to the Robot port of the Vision Controller.



## Interfaces configuration

Network interfaces of the vision controller are configured on the Network page of the Studio. Through these interfaces, the vision controller communicates with all connected hardware components of the application. Correct network setup is a fundamental part of the Studio configuration to achieve a working, robust and stable run of your application.

### Network page

Serves for the configuration of the communication interfaces:

- Robot interface and Robot controller  
Configuration of a static IP to which the client connects
- Scanner interface  
Configuration of a static IP or an IP on which a DHCP server runs
- Network interface  
Configuration of a static IP or a DHCP client, through this interface the user connects to the Internet, factory network, etc.

Note: When the **Save** button is pressed on a specific section of the Network page, the settings within this section are applied to the host system (Ubuntu) network settings. It is not possible to amend the network settings through the Ubuntu network panel.

## Robot interface

It is recommended to connect the robotic controller to the vision controller directly using the network port labeled **Robot** without using any additional communication device between them.

The vision controller communicates with the robotic controller over TCP/IP. No other communication protocols are currently supported.

The Robot section of the Network page serves for the configuration of the interface through which the vision controller communicates with the robotic controller.

This section is divided into two subsections:

- **Robot interface:** used for network setup of the Ethernet port on the vision controller.
- **Robot controller:** used to know the IP address of the robot controller.

Make sure both the robot interface IP address and the robot controller IP address are on the same subnet.

As described in the Robot communication, the communication between the vision controller and the robotic controller is realized using the Action Request Communication Channel. The vision controller creates a TCP Server to which the robotic controller connects and sends requests.

This TCP server runs on the default port and it is recommended not to change it and leave the field **Action Request Server port** empty. When a custom port is specified, using the same port on the robot side is required.

For setting up the communication with the robot (unique for each robot brand), please read the user manuals and tutorials which you can find on the Documentation page. To obtain a robotic package for a specific brand (modules and robot program), please visit [photoneo.com](https://photoneo.com).

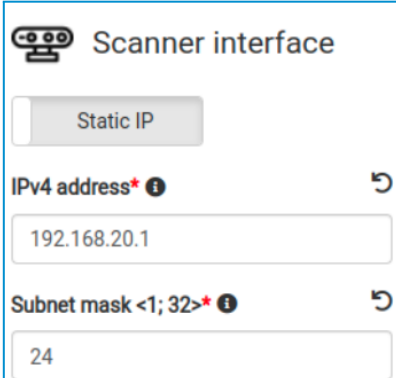
To quickly check the network reachability of the robotic controller from the vision controller, the **Test connection** button located on the **Robot** subsection of the Network page can be used. The connection check can result in

- Available connection
- Unreachable connection state, possible causes:
  - Wrong network configuration
  - The robotic controller is turned off (connection is not established)
  - Cable is unplugged/damaged

## Vision controller Interface for 3D Sensor

The 3D Sensor requires connection via Gigabit Ethernet cable (Cat5e or higher category). Multiple 3D sensors can be connected via a Gigabit switch. In both cases, the Ethernet cable from the 3D Sensor or switch is connected to the physical network port labeled **Scanner**.

A properly configured connection to the 3D Sensor is a precondition for visualizing 3D scans, performing robot camera calibration, and setting up localization settings.



The screenshot shows a window titled "Scanner interface" with a robot icon. It contains a "Static IP" toggle switch which is turned on. Below it, there are two input fields: "IPv4 address\*" with the value "192.168.20.1" and "Subnet mask <1; 32>\*" with the value "24". Both fields have information icons (i) and refresh icons (↺) to their right.

In the **Network** page, use the section **Scanner interface** to configure the network interface through which the vision controller communicates with the connected Photoneo 3D Sensor(s).

Photoneo 3D Sensors have a fixed IPv6 link-local address and a configurable IPv4 address.

IPv6 connection is preferred over IPv4. However, when the IPv6 connection is blocked or fails, the IPv4 address is used. Therefore, it is recommended to have a valid IPv4 network configuration.

The interface can be configured to use an arbitrary static IP address or to run a DHCP server. In both cases, make sure to use compatible IP address settings on the sensor side via PhoXi Control application.

### Static IP (recommended)

Use PhoXi Control for the configuration of the static IP address on the 3D Sensor. Then use the IPv4 address and subnet mask for the Vision controller that is on the same subnet. (Even when a vision controller has static IPv4 with a different subnet than the 3D Sensor, it might be visible via the IPv6 interface).

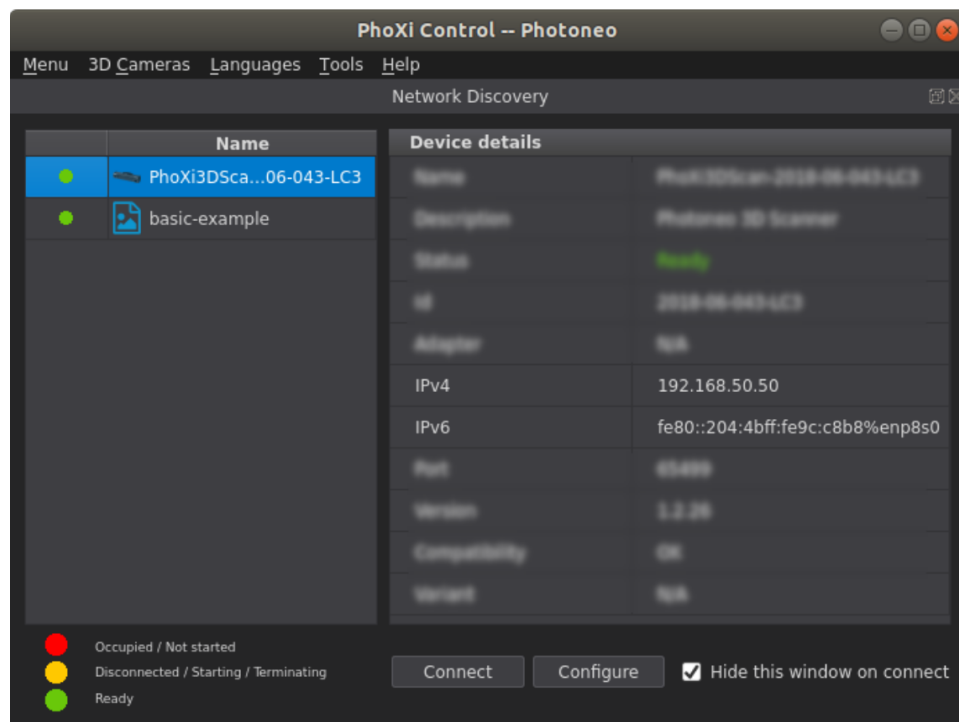
### DHCP server

DHCP server IP address is 192.168.101.1. Use PhoXi Control to configure 3D Sensor to use DHCP IP assignment. The 3D Sensor will be assigned an IP address from the range 192.168.101.2 - 192.168.101.255.

## 3D Sensor network configuration

Open **PhoXi Control** application from the tray.

Once the 3D Sensor is connected to the vision controller via the **scanner** interface port, (either directly or through a network switch), it will be listed in PhoXi Control. Click on the name of the device to see the assigned IPv4 and IPv6 addresses. Use **Configure** button to change the IP address assignment type: Static or DHCP.



## Valid network configurations

Scanner interface	Photoneo 3D Sensor	Scanner IPv4 field	Note
Static IP	Static IP	The IP address configured by the user	Recommended network configuration.  The scanner interface IP address and the 3D Sensor IP address are on the same subnet.
DHCP server	DHCP client	An IP address in range 192.168.101.2 - 192.168.101.255	
Static IP	DHCP client	An IP address in format 169.254.x.x	Not recommended network configuration.  The 3D Sensor was assigned an IP via Zeroconf technology. This connection type might be unstable.

In case the IPv4 field in PhoXi Control shows a value N/A, the network configuration is invalid - the scanner interface IP address and the Photoneo 3D Sensor IP address are not on the same subnet.



When the Deployment is starting, each vision system attempts to connect to the associated 3D Sensor several times. If none of these attempts are successful, a FATAL error message is raised.

## Connection to Local Network

Use the port labeled **Network** to connect the vision controller to the outside networks.

On the **Network** page, use the **Network interface** section for the configuration of the interface through which the vision controller connects to the Internet, the factory network, or other devices on the network. Read about [Remote access](#) methods to find out more.

The interface can be configured to use an arbitrary static IP address or to run a DHCP client.

### Static IP

This option is used when connecting directly to a remote device or to a local network without a DHCP server.

#### Direct connection to the remote device

Configure the network interface of the Vision controller and the remote device to be on the same subnet. The vision controller will be accessible from the remote device on the IP address of the network interface.

#### Connection to a local network

Configure the network interface of the Vision controller so that the IP address is within the range of the local network. If on a local network there is a DHCP server running, ensure that the static IP address assigned on the vision controller is outside of the range in which the DHCP server assigns IP addresses to prevent conflict of IP addresses.

The vision controller will be accessible from any device on the same local network on the IP address of the network interface.

If the local network is connected to the Internet, the vision controller will also have access to the Internet (provided the network interface has been configured properly - correct Gateway and optionally the DNS server).

### DHCP client

Note for remote access: *For the VNC and TeamViewer over LAN, it is recommended to use a static IP in order to make sure to have the Vision Controller is always accessible on the same IP address. Alternatively, it is possible to use this option (DHCP client) but it is recommended to use DHCP reservation to make sure the DHCP server always assigns the Vision Controller the same IP address.*

This option is used when connecting to a local network with a DHCP server. The parameters of the interface are non-editable - they are obtained from the DHCP server. The vision controller will be accessible from any device on the same local network on the IP address assigned to it by the DHCP server. If the local network is connected to the Internet, the vision controller will also have access to it.

# Robot module setup

## Robotic brands with native support

Photoneo offers Robot modules for Locator Studio for the following robotic brands:

- ABB
- FANUC
- Kawasaki
- Universal Robots

The brand of the connected robot is determined automatically when a new connection is established which enables proper communication with the vision controller and defines usage of coordination formalism for a specific brand.

The Robot module is a ZIP archive and can be downloaded from the [Photoneo website](#).

### Robot module content:

- **Photoneo robotic API**

A set of procedures (jobs/routines/programs) written in the native programming language(s) of the particular robotic brand. These procedures are used for easy interfacing with the Studio.

- **Example programs**

Several simple exemplary robotic programs with different approaches in standard applications.

- **Integration guide**

A guide that covers the configuration of the robotic controller for use with the Studio, installation of the Robot module, API documentation, and instructions for usage.

## Robot modules architecture

### Robot Interface and Robot Module

Communication between the vision controller and the robot controller is realized through the Robot interface on the vision controller and the Robot module running on the robot controller.

The vision controller is preconfigured with Robot interfaces for all [supported robot brands](#). The Robot module for the specific robot brand requires an installation on the robot controller in order to be able to communicate with the vision controller via the Robot interface.



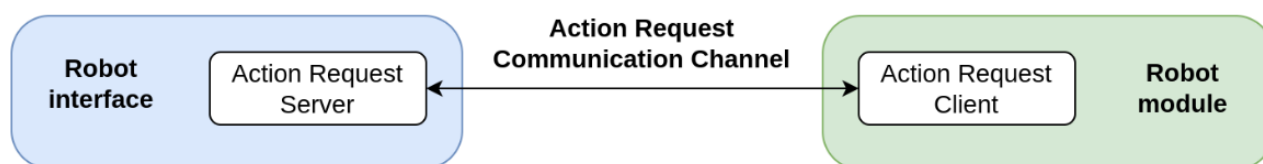
### TCP/IP protocol

TCP/IP protocol is utilized for communication between the vision controller and the robot controller. Some robot controllers do not include network communication out-of-the-box but as an additional option. Section Prerequisites of the Robot integration guides lists all requirements for complete installation of the Robot module.

*Note: No other communication protocols are currently supported.*

### Communication channel

The vision controller creates a TCP server and the Robot module acts as a TCP client. The server is referred to as the Action Request Server and the client is referred to as the Action Request Client. Through this channel, the Action Request Client sends action requests to the Action Request Server. The vision controller executes the received action request and sends a response back to the robotic controller.



## General principles

### Persistent communication

The Robot interface on the vision controller is available right after the vision controller boots up and runs until it's shut down. The Action Request Server on the vision controller is always waiting for the Action Request Client to connect. Similarly, the Robot State Client repeatedly tries to connect to the Robot State Server.

The part of the Robot interface whose network parameters were changed on the Network page of the Studio is restarted when the changes are applied by pressing the Save button.

### Full operator control

The Studio running on the vision controller does not take control over the robot - it sends responses to the client through the Action Request Communication Channel, e.g. the pose of the localized object for LS.

With the installation of the Robot module, the standard function library of the robot is extended by the Photoneo robotic API. It is the robot programmer/operator who decides when to send an action request and what should be done with the result received in the response.

#### Warning!

Extreme caution is necessary, especially during initial commissioning. Users must always make sure that the movement speed and acceleration are set appropriately and should be prepared to halt robot program execution immediately if necessary.

### Easy to integrate

Robot modules have been developed to be easily integrated into existing robotic applications written in standard robotic languages. From a robot programming point of view, the Robot module is a set of jobs/routines/programs that are available for the robot programmer/operator to be called when needed.

## List of Robot requests

### Locator requests

Request	Description	Blocking	Input parameters	Output parameters
Scan	Triggers a new scan	Non-blocking - response is received by the procedure waiting for scan completion	Vision System ID, TCP pose (only for hand-eye)	Error code
Get objects	Request poses of all localized objects. Timeout for response is set in the solution settings	Blocking	Vision system ID, number of requested objects	Error code, poses of objects

### Calibration requests

Request	Description	Blocking	Input parameters	Output parameters
Add calibration point	Triggers a new scan and adds calibration point	Blocking	TCP pose	

### Solution requests

Request	Description	Blocking	Input parameters	Output parameters
Change solution	Requests change of the deployed solution	Blocking	Solution ID	Error code
Start solution	Start of deployment of a solution	Blocking	Solution ID	Error code
Stop solution	Termination of deployment	Blocking	-	Error code
Get running solution	Request for ID of the deployed solution	Blocking	-	Error code Solution ID
Get available solutions	Requests IDs of the ready (to be deployed) solutions	Blocking	-	Error code List of solutions IDs

Note: For the specific controller calls, see the specific guide (for the specific robot)

## Custom Integrations

Custom integrations require the use of binary coded TCP/IP communication. Specification of the Communication protocol is available upon request.

*Note: In the case of developing a custom communication client based on open communication protocol - quaternions and meters are used.*

## Solution management

Use the top menu item **Solutions** to enter the solution management page. Each solution holds full application configuration (project) except for the network configuration of your installation.

Click on the solution to enter the Solution Configuration Wizard (chapter [Solution Configuration](#)).

Solutions							<a href="#">+ Create new</a> <a href="#">Import</a>	
<input type="checkbox"/>	Name	ID	Robot	Author	Last modified	Flags	Actions	
<input type="checkbox"/>	<a href="#">My ready solution</a>	★ 1		me	2022-10-25 15:16	<a href="#">▶ ready</a>	<a href="#">Duplicate</a>	<a href="#">Export</a> <a href="#">Delete</a>
<input type="checkbox"/>	<a href="#">My deployed solution</a>	★ 2		me	2022-10-25 15:16	<a href="#">deployed</a>	<a href="#">Duplicate</a>	<a href="#">Export</a>
<input type="checkbox"/>	<a href="#">My not ready solution</a>	★ 3		me	2022-10-25 15:15		<a href="#">Duplicate</a>	<a href="#">Export</a> <a href="#">Delete</a>
<input type="checkbox"/>	<a href="#">Photoneo factory solution</a>	★ 4		Admin	2022-10-25 15:14	<a href="#">factory default</a> <a href="#">▶ ready</a>	<a href="#">Duplicate</a>	<a href="#">Export</a>

## Flags

<a href="#">▶ ready</a>	completely configured solution (can be deployed)
<a href="#">deployed</a>	currently running solution
<a href="#">factory default</a>	preconfigured by Photoneo and (must be duplicated)

## Actions

### Create

Create a new empty solution from a scratch.

### Import

Import of solution from \*.pbcf file with all contained settings. The imported solution does not have to be complete. If the \*.pbcf file contains data from the previous version of the software, non-existing parameters will get default values and obsolete parameters will be omitted. Please, make sure to review the final configuration before deploying the solution.

### Duplicate

Action button **Duplicate** allows the user to create a duplicate of the existing solution. This solution contains all of the settings from the original solution and altering them does not change the settings in the original solution. The activity log of the duplicated does not contain the logs from the original solution. Users can simply make a backup by duplicating solutions and testing changes in the new one.

### Export

Each solution can be exported in a form of a \*.pbcf file. This solution can be then imported into the same or different Vision Controller using the **Import** button in the upper right corner of the Solutions. All of the settings from the original solution are preserved. The activity log is not exported nor imported.

### Delete

Deletes the solution permanently.



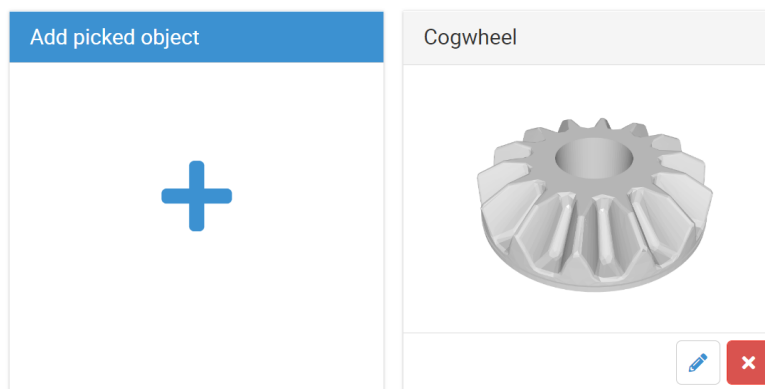


## Picked object

This step serves for uploading the CAD model (STL format) of the object to be localized. Localization is the process of estimating object position and orientation in the 3D scan.

It is possible to define up to 4 objects to be localized using the plus sign in the *Picked object*. In the next step of the wizard, the object will be associated with the vision system and its localization profile. The vision system can contain only one associated object, but it is possible to define 4 vision systems. This way by switching the vision systems one solution can localize 4 different objects.

Always only one type of object is being localized.



By default, the found position of the object is reporting the origin of its CAD model. This behavior can be changed by enabling **Model Origin Offset**.

Use the **Set Recommended Origin** button to let Locator Studio choose the recommended point of model origin.

**Picked object**

Name\*  
TFit

Description

CAD file (mesh)\* ⓘ  
Browse... ⓘ

Volumetric model (generated automatically) ⓘ  
Successfully generated  
Browse... ⓘ

Image  
Browse... ⓘ Delete

☒ Enable model origin offset ⓘ ↺

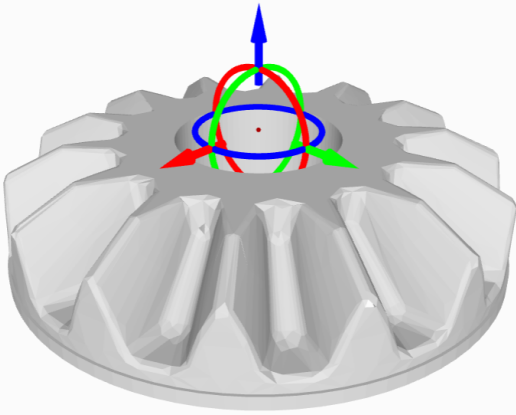
Translation X [mm]	Translation Y [mm]	Translation Z [mm]
0	0	-10

Rotation X [°]	Rotation Y [°]	Rotation Z [°]
0	0	0

↺ Reset model origin

⊙ Set recommended origin

Save



## Vision System & Localization

Vision System is a concept that encapsulates information about

- Photoneo 3D Sensor
- Scanning parameters (PHOP profile)
- Localized objects
- Localization parameters (PLCF profile)
- Robot-camera calibration

**Vision system**

Save Reset to defaults

Name\*  
TFit

Unique ID\*  
1

Scanner ID\* ⓘ  
UserExamples-CogWheel Available scanners

Scanner model\*  
PhoXi 3D Scanner L

⚠ File camera device is used. The scanner model can not be determined.

Scanner mount position\*  
Robotic cell

Picked object\*  
TFit

Localization profile\* ⓘ  
+ Create new Upload other Edit current ⓘ

Scanning profile type\* ⓘ  
Default for PhoXi 3D Scanner

Calibration matrix

0.007560	-0.999969	-0.002221	1076.963696
-0.988574	-0.007808	0.150534	-348.394838
-0.150547	0.001058	-0.988602	1487.541572
0.000000	0.000000	0.000000	1.000000

Edit matrix Copy to clipboard

It is possible to use up to 4 vision systems per solution.

The vision system is associated with exactly one Photoneo 3D sensor (PhoXi 3D Scanner or MotionCam-3D). One 3D Sensor can be associated with multiple vision systems - this is usually the case when utilizing the hand-eye approach or when a statically mounted Photoneo 3D sensor is able to cover multiple regions of interest: zones, pallets, etc.

3D Sensor can be mounted:

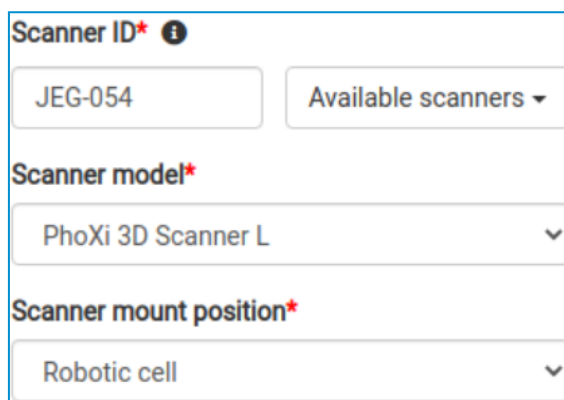
- Statically at the fixture outside of the robot - referred to as an extrinsic vision system or as robotic cell mount
- On the robotic arm - referred to as a hand-eye vision system

## 3D Sensor and scanning parameters of the Vision system

Selected 3D Sensor for the Vision system is defined by its ID as shown on the device label or in PhoXi Control. Supported product families are PhoXi 3D Scanners and MotionCam-3D.

The specific 3D Sensor model is selected automatically based on the sensor's ID.

Specifying the 3D Sensor mount position (robotic cell or hand-eye) is important for running the robot-camera calibration.

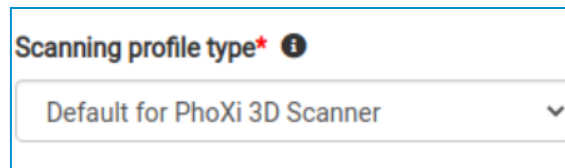


The screenshot shows a configuration window for a 3D sensor. It contains three main sections, each with a label and a dropdown menu. The first section is labeled 'Scanner ID\*' with an information icon, showing 'JEG-054' and a button 'Available scanners'. The second section is labeled 'Scanner model\*', showing 'PhoXi 3D Scanner L'. The third section is labeled 'Scanner mount position\*', showing 'Robotic cell'. All dropdown menus have a downward arrow icon.

## Scanning profiles

Each vision system might be used for the localization and picking of one object. When the properties of the objects differ too much it might not be possible to achieve optimal scan quality with the same scanning parameters. Therefore, it is possible to use different scanning parameters with each vision system. A set of scanning parameters is defined by a scanning profile.

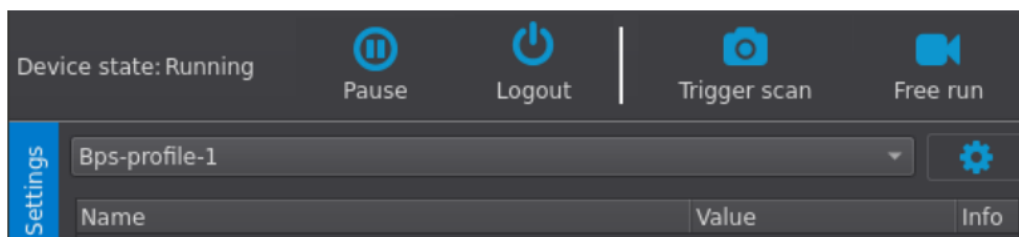
The Default profile is optimized for scanning a wide range of objects of different properties (surface/material).



Based on the Photoneo 3D sensor type the profile options are:

- Default for PhoXi 3D Scanner
- Default for MotionCam-3D in Camera mode
- Default for MotionCam-3D in Scanner mode
- Custom - when selected, it is necessary to upload a scanning profile exported from PhoXi Control

The scanning profile is imported into the associated 3D sensor under the name **Bps-profile-ID** where the ID is the ID of the vision system. The profile is then activated when the corresponding vision system triggers a scan.



During the vision system calibration, a preconfigured scanning profile is used containing calibration-specific settings.


*Note: Each 3D sensor provides several pre-configured factory profiles for different uses. These profiles serve as examples and starting points when creating a custom profile. The profile named Default in the PhoXi Control is not the same one as the Default profile used by the Studio.*


For specific cases, to achieve the best scanning results, it is recommended to use a **custom scanning profile** with different settings. Select *Custom* from the drop-down menu and the option for uploading a custom profile will show up. Upload your custom profile once the settings are finalized.


**Scanning profile type\*** ⓘ
 

Custom

**Scanning profile \*** ⓘ
 

 Browse...



 Delete

### Exporting scanning profile from PhoXi Control

Each Photoneo 3D sensor contains several factory profiles pre-configured with different scanning parameters for different uses. Factory profiles are not editable, and cannot be deleted or renamed; however, it is possible to clone them and edit the cloned profile. The profiles can also be imported/exported. To manage your scanning profiles open the User profile settings panel (gear button ⚙️).

After making changes in the scanning parameters, use the Set and Store button at the bottom of the PhoXi Control settings pane to permanently store the settings into your custom-made profile.

In order to use this profile within the vision system, export it and upload it to the Locator Studio web interface.

## Object and localization parameters of the Vision system

One picked object that was created in the wizard step **Picked object** can be associated with one vision system.

Localization parameters are stored within the Localization profile that can be created and adjusted in the web interface.

## Localization parameters

**Stop criteria**

Timeout [ms] <0; 60000> ⓘ  ☐ Off Number of localized objects <1; 999> ⓘ  ☐ Off If no object is found in [ms] <0; 60000> ⓘ

---

**Settings**

**General**

Overlap [%] <1.0; 100.0> ⓘ  ☐ Off Use the Empty scene feature ⓘ Empty scene threshold <0; 1000000> ⓘ

---

**Segmentation**

☒ On Scene noise reduction ⓘ  Scene clustering level ⓘ  Scene minimal cluster size <0; 9999999> ⓘ  Scene maximal cluster size <0; 9999999> ⓘ

---

**Searching**

Matching algorithm ⓘ  Model keypoint sampling ⓘ  Local search radius ⓘ  Feature fit consideration level <0; 100> ⓘ

Global maximum feature fit overflow <0; 100> ⓘ

---

**Occlusion handling**

Projection tolerance <0; 100> ⓘ  Projection hidden part tolerance <0; 100> ⓘ

---

**Fine alignment**

Fine alignment iterations <1; 1000> ⓘ  Fine alignment point set ⓘ  Fine alignment point set sampling ⓘ

- Explanation of each parameter is inside the tooltip.
- An overlap value higher than 15% is recommended.
- To achieve more precise localization results, lower the Global maximum feature fit overflow and Projection tolerance



## Bounding box

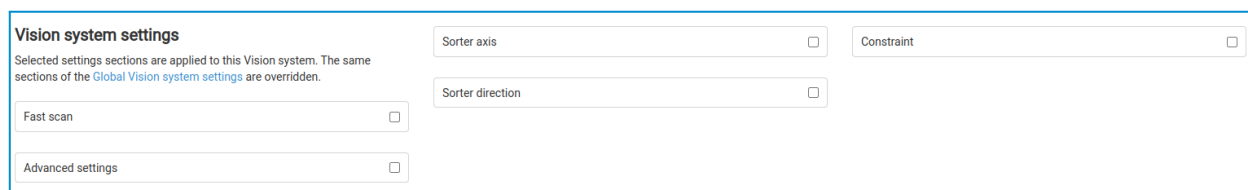
A bounding box defines the area of interest for localization which leads to higher speed and increased safety. It can be enabled from the drop-down menu, moved, and resized by the navigation tool in the visualizer



## Calibration matrix of the Vision system

Calibration matrix for the vision system is generated automatically after calibration (chapter [Calibration](#)), Editing the calibration matrix will modify the position of the sensor defined by the calibration. Use with caution.

## Settings of the Vision system



**Vision system settings**

Selected settings sections are applied to this Vision system. The same sections of the [Global Vision system settings](#) are overridden.

Fast scan ☐

Advanced settings ☐

Sorter axis ☐

Sorter direction ☐

Constraint ☐

Each vision system can have associated general settings affecting the limit of time used for localization or sorting direction of the found objects.

If vision system settings are not set, global solution-level settings are used.

When activated, vision system settings override global solution-level settings.

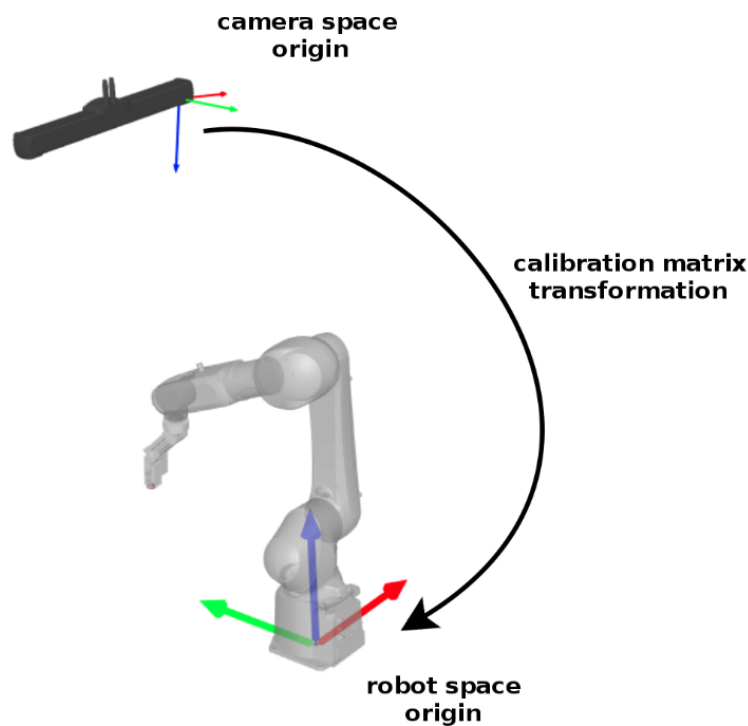
Consult Chapter [Settings](#) for additional information.

## Calibration (Robot-Camera)

Robot-camera calibration is an essential part of a configuration. Good calibration is the key to precise position reporting, picking, and, in the end, the correct placement of the localized object.

The reported position of the localized object in the 3D scan is returned in the coordinate space of the 3D Sensor. By default, the coordinate space of the 3D scan has its origin in the 3D sensor. This coordinate space needs to be transformed into the robot coordinate space with the origin in the robot base. The transformation from the sensor to the robot coordinate space is calculated by running the robot-camera calibration routine and on the output, there is a calibration matrix that contains the exact rotation and translation between the camera space and the robot space.





## Types of calibration

Based on the way the 3D sensor (PhoXi 3D scanner or MotionCam-3D) is mounted, we recognize two basic types of calibration:

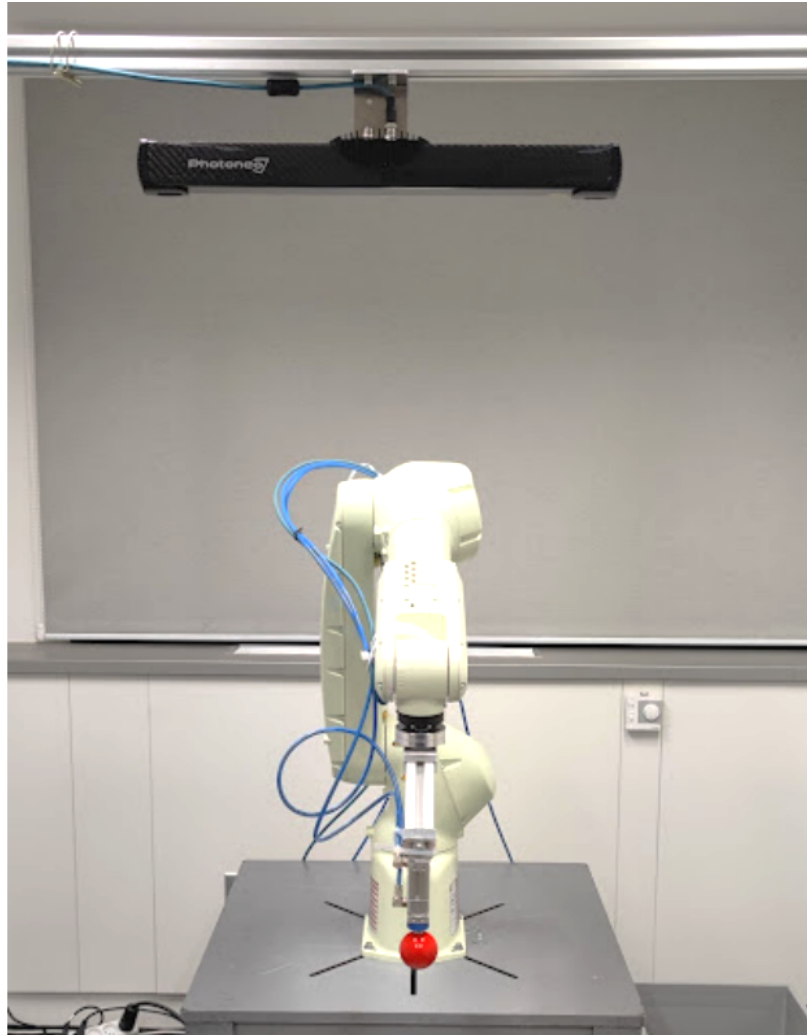
- Extrinsic - the 3D sensor is in a fixed, stationary position with respect to the robot
- Hand-eye - the 3D sensor is mounted on the robotic arm

### Extrinsic calibration

The extrinsic calibration is used in setups where the 3D sensor is mounted in a stationary position in the robotic cell, usually above the bin. It is only required that the 3D sensor is stationary relative to the robot's base and the 3D sensor does not need to be stationary relative to the robotic cell itself. Any change in the relative position between the 3D sensor and the robot after the calibration will render the calibration matrix invalid and the whole calibration procedure must be repeated. In other words, the 3D sensor must not be moved in relation to the robot after the system has been calibrated.

Since the 3D sensor is fixed, the calibration matrix defines the transformation directly to the robot base coordinate system.

This setup does not require the robot to stop its movement during the scan acquisition (as is the case with the hand-eye approach) and the robot's movement is not limited in any way.

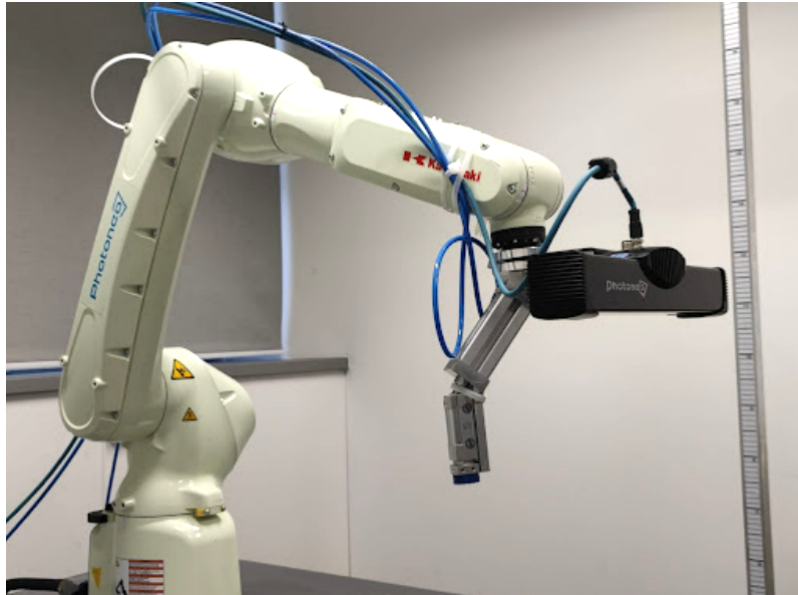


## Hand-eye calibration

Hand-eye calibration is used in setups where the 3D sensor is mounted on the robotic arm itself. For the hand-eye calibration to work correctly, the 3D sensor must be mounted behind the very last joint (e.g. on the robotic flange or the gripper). Any change done to the sensor's mount position after the calibration renders the calibration matrix invalid and the whole calibration procedure must be carried out again.

Since the 3D sensor is not fixed, the calibration matrix defines the transformation to the robot's TCP which needs to be known in the moment of object localization. Using the TCP data the pose of the object can be transformed into the robot base coordinate system.

This setup allows for dynamic change of the scanning distance from the bin just by moving the robot. Similarly, the scene can be scanned from various angles.



## Requirements and setup

Before the calibration can be performed, it is necessary to configure the network parameters, calibration-related parts of the solution, robot controller, and to prepare the calibration tool.

### Network

Network settings must be configured correctly so that the Vision controller is able to communicate with both the 3D sensor and the robot - it is necessary to correctly configure the robot interface (and robot controller IP) and the 3D Sensor interface of the Vision controller.

Please read the [Network setup](#) to find out more about the configuration of the network interfaces.

### Vision system

The vision system that is to be calibrated needs to be fully configured. Before the calibration is started, make sure that the following parameters of the vision system are configured correctly:

- Scanner ID - 3D sensor used with this vision system. If the scanner interface is configured correctly, the connected 3D sensors will be in the drop-down list of available 3D sensors
- Scanner model - determined automatically based on the chosen 3D sensor

- Scanner mount position - specifies the mount point of the 3D sensor and hence the calibration type

Note: During the calibration procedure, preconfigured default scanning profile for calibration is used.

## Robot

### Base frame offset

The origin of camera space is calibrated with respect to the coordinate frame in which the robot reports the TCP pose. Adjust the coordinate frame of the robotic controller according to the requirements of your application.

### Tool center point

Based on the calibration type the following requirements must be met:

- Extrinsic calibration
  - During manual calibration, a tool must be selected (*tool0* or custom) and cannot be changed during the whole calibration procedure.
  - During semi-automatic calibration, the same TCP pose (*tool0* or custom) must be included and transferred in all Add calibration point requests.
  - During localization, the selected tool is not relevant, no TCP pose should be included and transferred in the Scan request.
- Hand-eye calibration
  - During manual calibration, a tool must be selected (*tool0* or custom) and cannot be changed during the whole calibration procedure.
  - During semi-automatic calibration, the same TCP pose (*tool0* or custom) must be included and transferred in all Add calibration point requests.
  - During localization, the same TCP pose as the one sent in the Add calibration point requests during calibration must be included and transferred in the Scan request.

### Robot State Server

The Robot State Server functionality is not used in the Locator Studio. The robot state is provided by the operator input during manual calibration. For semi-automatic calibration, the robot state is included and transferred in the **Add calibration point** request.

## Calibration objects

Based on the calibration type, different calibration objects are used. These objects are included in the standard scope of delivery.

### Extrinsic calibration

The calibration object is a calibration ball. Instead of the calibration ball delivered by Photoneo, it is also possible to use a custom ball with suitable properties - the ball must be perfectly round and made of a material (surface) appropriate for scanning (smooth and not highly reflective).

The ball must be mounted onto the robot's endpoint (either the flange or the gripper). Make sure the ball is properly fixed and does not move during the calibration procedure. Make sure that the 3D sensor has good visibility to the ball.

### Hand-eye calibration

The calibration tool used is a marker pattern.

Place the pattern within the reach of the robot so that the calibration pattern is visible to the 3D sensor. Make sure the pattern does not move during the calibration procedure and that it is perfectly flat (when using a custom-printed pattern).

## Manual and Semi-automatic calibration

### Manual calibration

During manual calibration the operator requests adding of a calibration point from the web interface. The current robot state is provided manually by the operator.

### Semi-automatic calibration

The robot itself is also able to request adding of a calibration point - the operator just starts, verifies, and finishes the calibration procedure (semi-automatic calibration). The Robot module contains a program template for adding the calibration points by calling the **Add calibration point** request. It is necessary to teach/update the robot calibration poses before running the program.

The robot state is included and transferred in the request (the selected Robot state input is not relevant).

## Calibration procedure

To start the calibration, navigate to the **Calibration page** inside the solution. It shows the list of all defined Vision systems with the following information:

- ID - ID of the Vision system
- Name - the name of the Vision system
- Scanner ID - ID of the 3D sensor used by the Vision system
- Scanner status - current status of the 3D sensor used by the Vision system
- Scanner model - the model of the 3D sensor used by the Vision system
- Calibration type - calibration type based on chosen 3D sensor mount position
- Calibrated - whether the vision system has been already calibrated
- Calibration accuracy [mm] - accuracy of any previous calibration if applicable

ID	Name	Scanner ID	Scanner Status	Scanner model	Calibration type	Calibrated	Calibration accuracy [mm]	
1	Object detector 1	2019-03-007-LC3	Not available	PhoXi 3D Scanner M	Extrinsic	✓	0.076	<a href="#">Start calibration</a>
2	Object detector 2	JKE-055	Available	PhoXi 3D Scanner M	Hand-eye	✗	N/A	<a href="#">Start calibration</a>

Calibration can only be successfully started if the 3D sensor used by the Vision system is available. Use the **Reload scanners** button to update the statuses of 3D sensors.

Please verify the configuration one more time and then click the button **Start calibration** for the **Vision system** you wish to calibrate.

### Confirming parameters

Before the calibration is started its parameters must be confirmed.

### Calibration ball selection

*Note: Only for extrinsic vision systems*

The radius of the calibration ball needs to be specified. It is possible to choose one of the predefined options (table tennis or snooker ball) or to define a custom one.

### Robot state input

The Robot State Server functionality is not available in the Locator Studio.

When performing manual calibration, the current state of the robot (the TCP pose) is provided by the operator manually. When adding a new calibration point is requested, a form appears that needs to be filled with the current TCP data from the teach pendant.

The brand of the robot that needs to be chosen specifies the rotation formalism used in the TCP pose.

For other robotic brands, it can be used only in case the rotation formalism of a given brand is the same as that of one of the supported robotic brands or, when converting the rotation formalism of the robotic brand to the rotation formalism of one of the supported robotic brands.

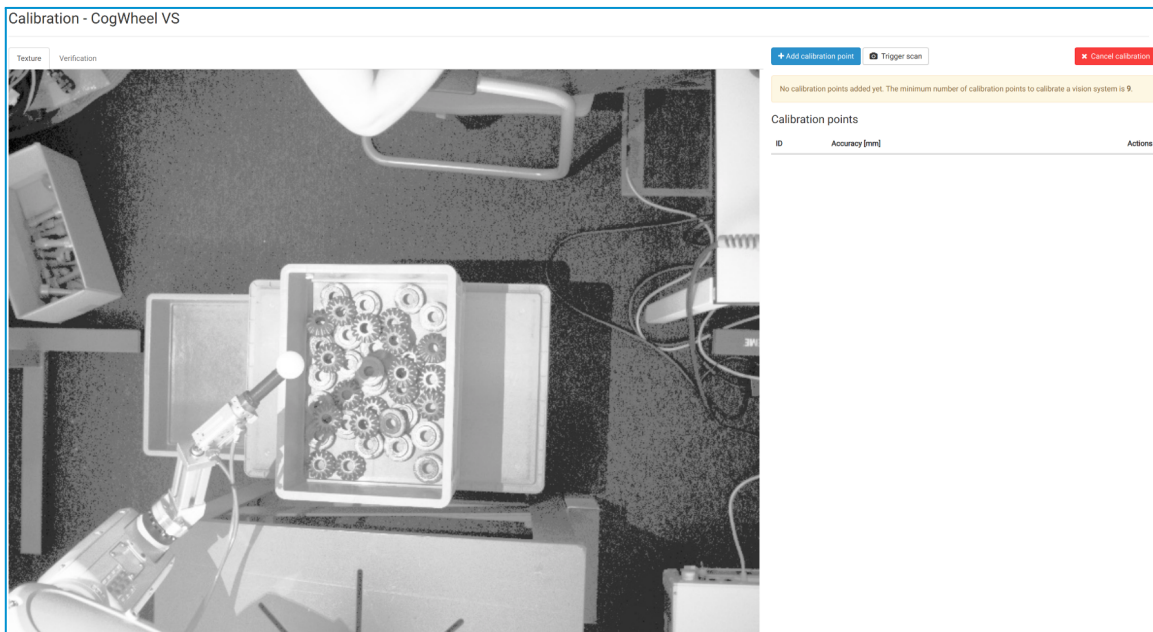
## Calibration

The calibration interface consists of the following parts:

- Visualization - Texture tab and Verification tab which can be switched between
  - Texture image (Texture tab)
 

The texture image is from the last scan acquisition.

For the extrinsic calibration, the localized calibration ball is highlighted with green color if it was found successfully. On the other hand, if not enough surface of the ball is visible or another object of a similar shape was detected, that piece of texture will be highlighted with red color.
- 3D visualizer (Verification tab)
  - The visualizer displays 3D sensors of all vision systems and the point cloud.
- Add calibration point - button for manual adding of a calibration point
- Trigger scan - button for triggering a new scan, usually used to acquire the current point cloud for verification purposes
- List of calibration points - list of all successfully added calibration points with individual accuracies and the option to delete a point
- After the required number of calibration points has been added, the following items appear:
  - Calibration matrix - the result of the calibration procedure
  - Calibration accuracy - overall accuracy of the calibration
- Finish and save result - button for finishing the calibration procedure and saving the result (calibration matrix) into the vision system



## Robot calibration poses

During the extrinsic calibration, the 3D sensor needs to capture the calibration ball in various places of the whole area of interest. Similarly, during the hand-eye calibration, the 3D sensor needs to capture the marker plate from various viewpoints. Therefore, move the robot to various poses, with joint orientations as diverse as possible, and in every one of them, add a new calibration point.

## Adding a calibration point

A new calibration point is added either

- By the operator manually, by pressing the Add calibration point button in the calibration web interface
- By the robot by calling the Add calibration point request

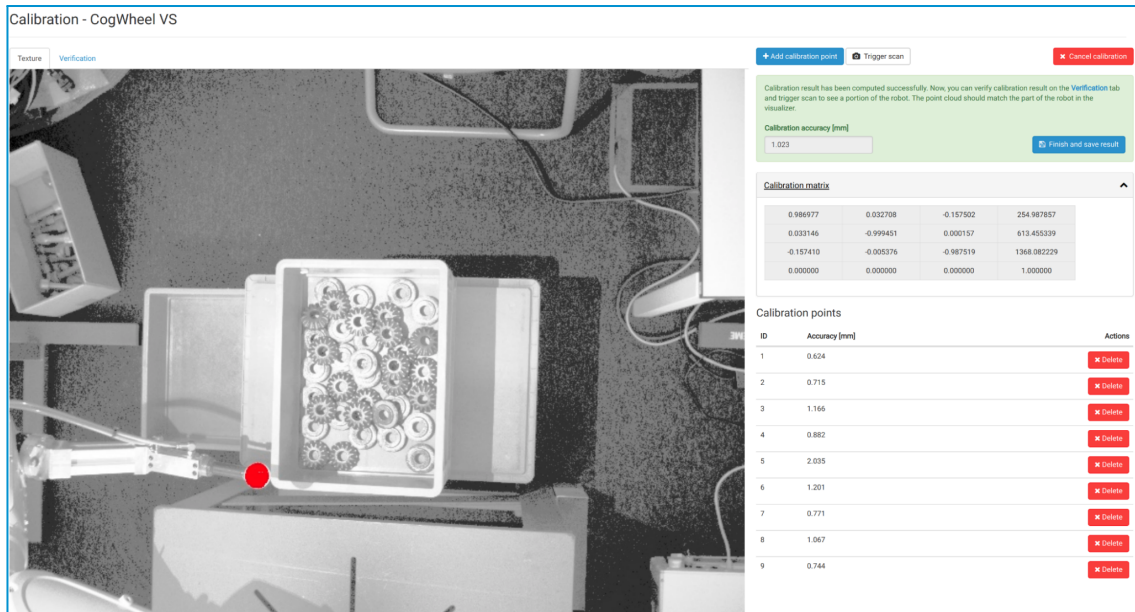
When a new calibration point is added, a scan acquisition is triggered and the calibration tool (ball/pattern) is detected. If the detection is successful, a flash message appears notifying the user about the successful result. In case the calibration point failed to be added, a persistent error message will provide information about the cause and how to solve the error.

The required number of calibration points is 9 for both the extrinsic and hand-eye calibration. Only after the required number of calibration points has been added does the resulting calibration matrix and information about the calibration accuracy become available.



### 3D sensor and point cloud position

After the 4th point is added during the extrinsic calibration and the 6th during the hand-eye calibration, the model of the 3D sensor of the currently calibrated vision system starts to reflect the pose of the real 3D sensor. At the same time, the point cloud is displayed.



### Verification

Each successfully added calibration point has an ID and its own accuracy score expressed in millimeters. The lower the accuracy score, the higher the confidence in the accuracy. When a point's accuracy score is too high, such point can be removed from the dataset using the Delete button.

Calibration quality is expressed by the aggregate score of Calibration accuracy expressed in millimeters. Always aim for a score as low as possible. However, the value is affected by many factors, such as the size of the 3D sensor, the quality of the calibration ball, scanning parameters and the lighting conditions, and the size of the robotic arm. Even an accuracy value of several millimeters can be considered satisfactory.

Calibration quality is verified through the value of Calibration accuracy.

Besides that, the approximately correct result (without taking accuracy into consideration) can be confirmed by evaluating the position of the 3D sensor and the point cloud in the scene.

The origin of the scene lies in the origin of the coordinate frame of the robotic controller with respect to which the calibration was carried out. Usually, that means the origin of the robot base. The relative

position of the model of the 3D sensor and the scene origin should then be similar (the same) as the relative position of the real 3D sensor and the robot base in reality.

## Tips and Troubleshooting

Although calibration looks like a simple task, there are some best practices for reaching the highest possible accuracy:

- Maneuver the robot to positions as diverse as possible. Try to add poses with great joint differences.
- During extrinsic calibration, place the ball in various places of the bin (such as its corners), including its approximate center. Cover the entire bin volume with calibration points if possible.
- During hand-eye calibration, scan the marker pattern from various robot poses.
- Avoid using small balls for extrinsic calibration. A bigger ball means a greater scanned surface and more points of the point cloud dedicated to ball recognition. This is especially important when using larger models of 3D sensors.

If you encounter any of these errors:

- The whole point cloud is offset in the world's Z-axis. It does not match the virtual model.
  - Some robotic brands require a custom user frame to be set on the robotic controller. Refer to the section [Robot](#) for more information.
- I can see the calibration ball on the texture image but the ball is not recognized. Previous points have been added successfully.
  - The camera can sometimes see parts of the scene that cannot be scanned. Make sure the calibration ball is also visible in the point cloud (check in the PhoXi Control). The ball may be too close to the 3D sensor and thus outside of the scanning volume.

## Settings (Solution-level)

Each solution contains **Global Vision System settings** that are used for all its vision systems unless overridden by specific vision system settings

## Settings

### Global Vision system settings

These settings are applied to all Vision systems unless overridden. Every Vision system can override any of these settings with custom values.

#### Fast scan

☐ Use fast scan ⓘ

#### Advanced settings

Timeout <0; 10000> ⓘ

#### Sorter axis

Weight <0; 10000> ⓘ

X <-1.0; 1.0> ⓘ

Y <-1.0; 1.0> ⓘ

Z <-1.0; 1.0> ⓘ

#### Sorter direction

Weight <0; 10000> ⓘ

X <-1.0; 1.0> ⓘ

Y <-1.0; 1.0> ⓘ

Z <-1.0; 1.0> ⓘ

#### Constraint

Angle <0; 180> ⓘ

Axis X <-1.0; 1.0> ⓘ

Axis Y <-1.0; 1.0> ⓘ

Axis Z <-1.0; 1.0> ⓘ

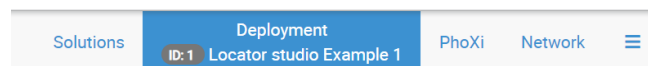
Parameter	Explanation
Fast scan	When on, a 2D image is captured to check changes in the scene. If there are no significant changes, the previous 3D scan is reused for the new localization.
Timeout	Time to search for new objects in localizations. Applies for Get object(s) request from the robot program.
Sorter Axis	<p>Ordering of localization results based on the orientation of their Z-axis. Objects with the Z-axis closer to the defined Axis are prioritized. The axis is specified as a directional vector defined in the base coordinate space. For example, vector [0,0,1] means the object whose Z-axis points upwards is the best candidate.</p> <p>Sorter Axis can be combined with Sorter Direction based on the respective Weight parameter.</p>
Sorter Direction	<p>Ordering of localization results based on specified direction by a vector in the base coordinate space.</p> <p>Vector [0,0,-1] means the objects are sorted from top to bottom</p> <p>Vector [1,0,0] means the objects are sorted respectively to the X axis</p> <p>Sorter Axis can be combined with Sorter Direction based on the respective Weight parameter.</p>
Constraints	<p>A localized object can be rejected if the difference between the object's Z-axis and the defined <b>Vector (Axis X, Axis Y, Axis Z)</b> is higher than the defined <b>Angle</b>.</p> <p>The default maximum allowed angle is set to 50 degrees and the vector is [0,0,1] defined in the world coordinate system - it points upwards.</p>

# Deployment

When no **Solution** is already deployed, the **Deployment Page** provides the list of fully configured solutions with the option to deploy them.

Deployment				
Name		ID	Author	
Locator studio Example 1	★	1	Locator Studio	<button>▶ deploy</button>
Production Solution LS	★	2	Locator Studio	<button>▶ deploy</button>

Button **Deploy** starts the solution after confirming that the environment is compliant with all the required safety regulations.

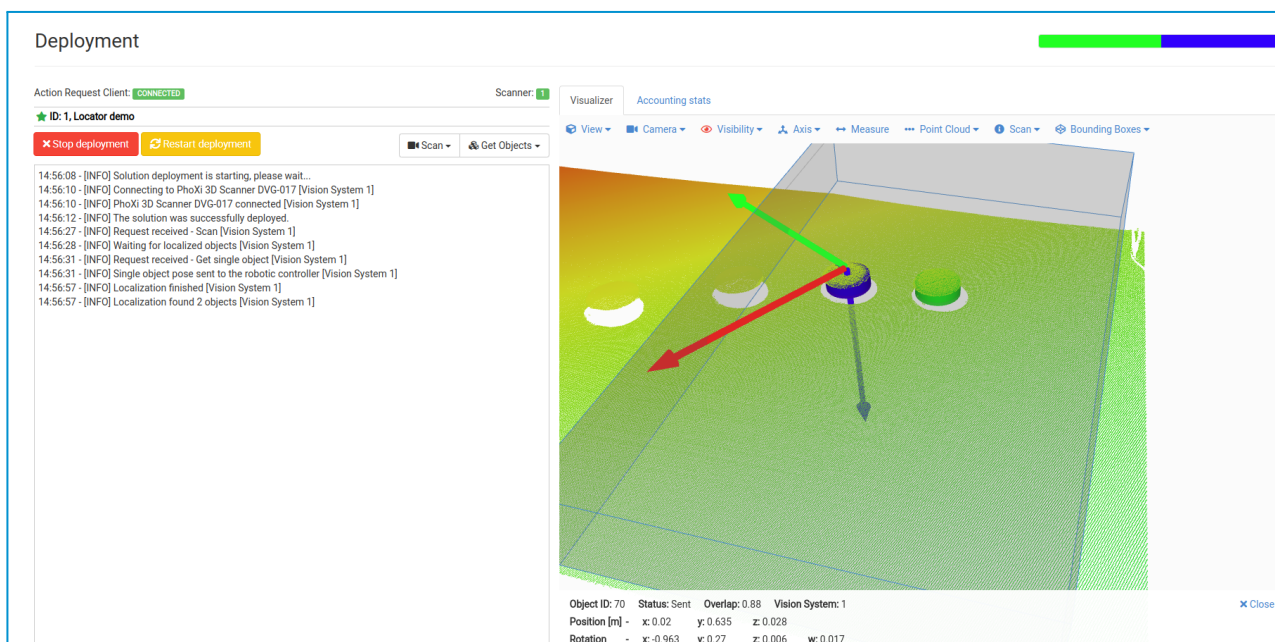


After a configured solution has undergone a deployment, runtime requests from action clients are received and carried out. The user has the option to see the connection statuses of the Photoneo 3D Sensors (vision systems) and the Action Request Client (Robot main program).

Furthermore, it offers an option to stop or restart the deployed solution and call a **Scan** request and **Get object** request.

Deployment page displays console logs informing about the past action and events and visualiser with following data:

- Real-time visualization of the captured 3D point cloud
- Localized objects rendered in colors corresponding to their status
  - Localized
  - Being picked
  - Rejected
- Inspector tool for diagnostics of every localized object



There is an indicator informing about the current state of the robot connection status:

- Action Request Client - status of the connection between the Action Request Server and Action Request Client (the Action Request Communication Channel). It can be in two different states:
  - **CONNECTED** - The Action Request Client is connected to the Action Request Server and can send requests.
  - **DISCONNECTED** - The Action Request Client is not connected to the Action Request Server, the server is waiting for a new connection. If you are encountering connection problems, please check:
    - Is the robotic controller reachable? → Use the **Test connection button** on the Network page.
    - Are you connecting to the correct IP address (Robot interface IP address) in the robotic program?
    - In the case of custom port usage, have you applied the same changes both on the Network page and in the main robotic program?

There is a status indicator for each vision system informing about the current state of the connection to the associated Photoneo 3D Sensor, Each indicator carries the ID of the vision system it belongs to

- **CONNECTED** - The 3D Sensor associated with vision system 1 is connected
- **DISCONNECTED** - The 3D Sensor associated with vision system 2 is not connected

When a valid connection to a Photoneo 3D Sensor is suddenly interrupted during the deployment a FATAL error message is raised.

## Remote access

The Locator Studio is running locally on the vision controller but it is pre-configured to be also accessible remotely using one of the methods such as VNC, TeamViewer, or a Web browser on another computer.

VNC connection and access from a web browser on another computer require both devices (the vision controller and the remote computer) to be on the same local network. TeamViewer connection is also possible through the Internet.

The physical network port reserved for remote control of the Studio is labeled **Network** and can be configured directly in the Studio in the Network section of the Network page.

For the VNC and TeamViewer over LAN, it is recommended to use a static IP in order to make sure to have the Vision Controller is always accessible on the same IP address.

For further information please refer to the [Network setup](#) documentation.

For proper functions of VNC and TeamViewer, it is necessary to connect a monitor to the vision controller.

## VNC

Virtual Network Computing (VNC) is a graphical desktop-sharing system that allows controlling a computer remotely. It transmits keyboard and mouse events from one computer to another, relaying screen updates back in the other direction, over a network.

VNC is platform-independent – there are clients and servers for many GUI-based operating systems. Multiple clients may connect to a VNC server at the same time.

Locator Studio acts as an VNC server.

A VNC client (or viewer) is a program on a remote machine that connects to the VNC server to allow remote control of it and to represent the screen and other data originating from the server on the client's computer.

## Recommended clients

Linux OS: [Vinagre](#).

Windows: [TightVNC](#).

## How to connect

The VNC server is started automatically after the vision controller is booted up. To connect to the VNC server from your device running the VNC client, please use the IP address of the vision controller as configured on the Network page and (if necessary) port 5900. The password is available upon request at [support@photoneo.com](mailto:support@photoneo.com).

*Note: When using the VNC method, to ensure smooth operation, it is recommended not to have a physical display connected to the vision controller during boot. If already booted up, please turn off the controller, disconnect the physical display and start the controller again.*

## TeamViewer

[TeamViewer](#) is remote access and remote control software. It allows connection to a device within the same local network using an IP address or from anywhere through the Internet using a unique device ID. Authentication is done via password - either a one-time-only password or a permanent one. TeamViewer is used for remote support by Photoneo technicians.

## How to connect

To connect to a vision controller TeamViewer has to be also installed on the device used for remote control. In the case of a LAN connection, an IP address of the vision controller is used as an identification of the device to connect. In the case of an Internet connection, a unique TeamViewer ID is used as an identification.

Use the password displayed in the TeamViewer window on the vision controller or the configured permanent password for authentication.

## LAN connection

To connect through a LAN connection the IP address of the vision controller as configured on the Network page is used. The IP address is shown directly in TeamViewer if no Internet connection is available or when configured to only accept LAN connections.

## Internet connection

When connected to the Internet and Internet connections are enabled, TeamViewer shows a unique device ID through which the connection can be established.

## Web browser on another computer

The Studio is a web-based application which means it provides a web server and can be accessed using a web browser. On the vision controller, it is configured to run in pre-installed Google Chrome on the localhost (IP address 127.0.0.1).

It can also be accessed from a web browser on another computer. However, in that case, there is no way to interact with the PhoXi Control or the host system of the vision controller.

## Recommended browsers

It is recommended to use Google Chrome when accessing the Studio from another computer.

## How to connect

To access the Studio through a web browser on another computer please use the IP address of the vision controller as configured on the Network page. Type the IP address in the URL field of your browser to load the Studio web application.



## Remote support

Photoneo offers remote support for its customers. To connect to your Vision Controller, it needs to be connected to the Internet (through the Network port). Your Vision Controller comes with a pre-installed TeamViewer. If you need remote support, please send an email to [support@photoneo.com](mailto:support@photoneo.com) with a description of the issue and login credentials to TeamViewer.