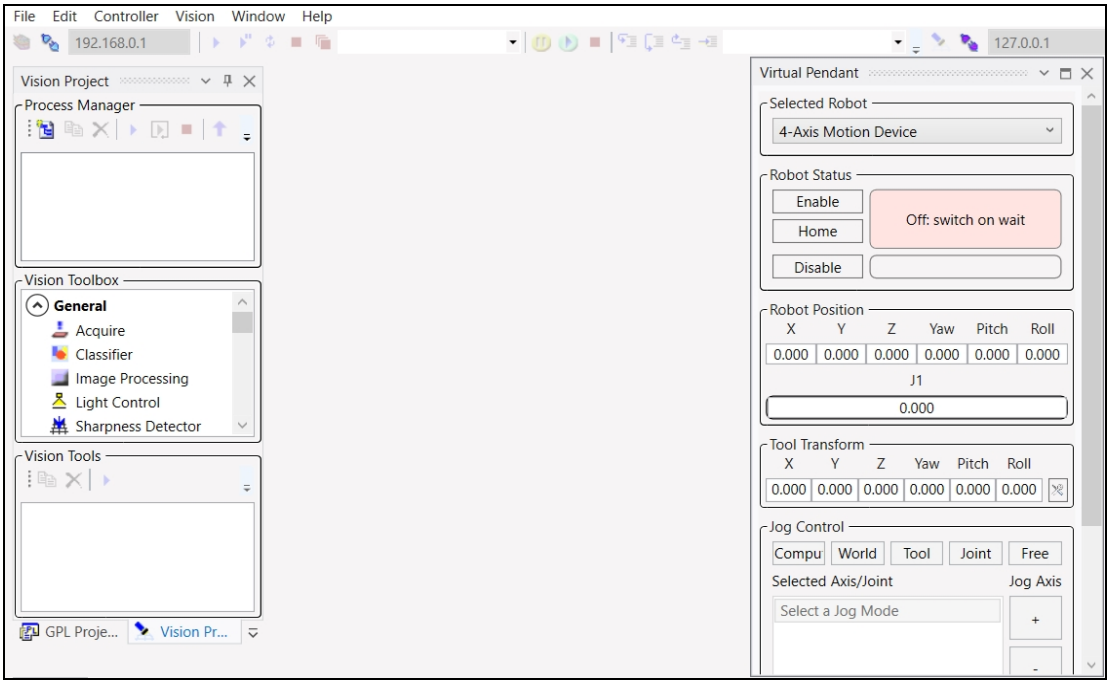


Guidance Development Studio (GDS)

User Manual

Part Number 643533, Revision A

Brooks Automation

Information provided within this document is subject to change without notice, and although believed to be accurate, Brooks Automation assumes no responsibility for any errors, omissions, or inaccuracies.

AcuLigner™, Advan Tag, AutoTeach™, ATR™, AXM™, BiSymmetrik™, CenterSmart™, Crate to Operate™, CrossingConnect™, DARTS™, Enerta™, e-RMA™, e-Spares™, e-Volution™, Falcon™, FIXLOAD™, FrogLeg™, GuardianPro™, Independent Twin Linear Exchange™, InCooler™, InLigner™, Isoport™, ITLX™, Jet™, Jet Engine™, LEAP™, LeapFrog™, LowProfile™, LPT™, M2 Nano™, Marathon 2, Marathon Express, PASIV™, Pathway™, PowerPak™, PowerTools™, PuroMaxx™, QuadraFly™, Radius™, Radient™, Radient Express™, Reliance™, Reliance ATR™, RetroEase™, SCARA™, SmartPM™, SMIF-INX™, SMIF-LPT™, SPOTLevel™, The New Pathway to Productivity™, Time Optimized Trajectory™, Time Optimal Trajectory™, Time Optimized Path™, TopCooler™, TopLigner™, VacuTran™, VersaPort™, WaferEngine™, LEAP™, Pathway™, GIO, GSB, Guidance 6600, Guidance 6430, Guidance 6420, Guidance 6410, Guidance 6000, Guidance 3400, Guidance 3300, Guidance 3200, Guidance 2600, Guidance 2400, Guidance 2300, Guidance 2200, Guidance 1400, Guidance 1300, Guidance 1200, Guidance 0200 Slave Amplifier, Guidance 0006, Guidance 0004, Guidance Controller, Guidance Development Environment, GDE, Guidance Development Suite, GDS, Guidance Dispense, Guidance Input and Output Module, Guidance Programming Language, GPL, Guidance Slave Board, Guidance System, Guidance System D4/D6, IntelliGuide, PreciseFlex™ 100, PreciseFlex™ 300, PreciseFlex™ 400, PreciseFlex™ 3400, PreciseFlex™ c3, PreciseFlex™ c5, PreciseFlex c8a™, PreciseFlex c10™, PreciseFlex™ 1300, PreciseFlex™ 1400, PreciseFlex™ DD4, PreciseFlex™ DD6, PreciseFlex™ DDR, PreciseFlex™ G5400, PreciseFlex™ G5600, PreciseFlex™ G6400, PreciseFlex™ G6410, PreciseFlex™ G6420, PreciseFlex™ G6430, PreciseFlex™ G6600, PreciseFlex™ GSB Slave Amp, PreciseFlex™ PFD0, PrecisePlace 100, PrecisePlace 0120, PrecisePlace 0130, PrecisePlace 0140, PrecisePlace 1300, PrecisePlace 1400, PrecisePlace 2300, PrecisePlace 2400, PrecisePower 300, PrecisePower 500, PrecisePower 1000, PrecisePower 2000, PreciseVision, IntelliGuide Vision, and RIO logos are trademarks of Brooks Automation.

Fusion®, Guardian®, MagnaTran®, Marathon®, Razor®, Spartan®, Vision®, Zaris®, and the Brooks and design logo are registered U.S. trademarks of Brooks Automation.

All other trademarks are properties of their respective owners.

This technology is subject to United States export Administration Regulations and authorized to the destination only; diversion contrary to U.S. law is prohibited.

Brooks Automation
15 Elizabeth Drive
Chelmsford, MA
01824-2400
Tel: +1 978-262-2400
Fax: +1 978-262-2500

Brooks Automation, PreciseFlex Collaborative Robots
201 Lindbergh Avenue
Livermore, CA
94551
Tel: +1-408-224-2838

Worldwide Headquarters
15 Elizabeth Drive
Chelmsford, MA 01824 U.S.A.

Brooks Automation,
PreciseFlex Collaborative Robots
201 Lindbergh Avenue
Livermore, CA 94551 U.S.A

Technical Support

Location	Contact	Website
North America	+1-800-447-5007 (Toll-Free) +1-978-262-2900 (Local) +1-408-224-2838 (PreciseFlex™)	http://www.brooks.com/
Europe	support_preciseflex@brooksautomation.com	
Japan	+81 120-255-390 (Toll Free) +81 45-330-9005 (Local)	
China	+86 21-5131-7066	
Taiwan	+886 080-003-5556 (Toll Free) +886 3-5525258 (Local)	
Korea	1800-5116 (Toll Free)	
Singapore	+65 1-800-4-276657 (Toll Free) +65 6309 0701 (Local)	

General Emails

Division	Email Address
Sales	sales_preciseflex@brooksautomation.com
Technical Support	support_preciseflex@brooksautomation.com
Technical Publications	Technical.Publications@brooksautomation.com

Brooks Automation
15 Elizabeth Drive
Chelmsford, MA
01824-2400
Tel: +1 978-262-2400
Fax: +1 978-262-2500
www.brooks.com

Brooks Locations Worldwide:

Brooks Automation
46702 Bayside Parkway
Fremont,CA 94538
Tel: +1-510-661-5000
Fax: +1-510-661-5166

Brooks Automation
AIM Servicios Administrativos
S de RL de CV
Carretera Huinalá km 2.8
Parque Industrial Las Américas
66640 Apodaca, NL Mexico
Tel: +52 81 8863-6363

Brooks Automation (Germany) GmbH
Ernst-Ruska-Ring 11
07745 Jena, Germany
Tel: +49 3641 4821 100
Fax: +49 3641 4821 4100

Brooks Automation (Germany) GmbH
Daimler-Straße 7
78256 Steißlingen, Germany
Tel: +49-7732-9409-0
Fax: +49-7732-9409-200

Brooks Automation
9601 Dessau Road, Suite 301
Austin, TX 78754
Tel: +1 512-912-2840
Toll-Free: +1 800-367-4887

Brooks Automation (Israel) Ltd.
Mevo Yerach 5
Kiryat-Gat 82000
Israel
Tel: +972 8672 2988
Fax: +972 8672 2966

Brooks Technology (Shanghai) Limited
2nd Floor, No. 72,
887 Zuchongzhi Road
Zhangjiang Hi-Tech
Park Pudong, Shanghai
China 201203
Tel: +86-21-5131-7070
Fax: +86-21-5131-7068

Brooks Japan K.K.
HEADQUARTERS
Nisso Bldg. No 16, 9F
3-8-8 ShinYokohama, Kohoku-ku
Yokohama, Kanagawa 222-0033
Tel: +81-45-477-5570
Fax: +81-45-477-5571

Brooks Japan K.K.
YOKOHAMA TECHNICAL CENTER
852-1 Kawamuko-cho, Tsuzuki-ku
Yokohama, Kanagawa 224-0044
Tel: +81-45-477-5250
Fax: +81-45-470-6800

Brooks Japan K.K.
KUMAMOTO SERVICE OFFICE
202 Mirai Office II
312-1 Tatsudamachi Yuge
Tatsuda, Kumamoto 861-8002
Tel: +81-96-327-9720
Fax: +81-96-327-9721

Brooks CCS Japan K.K.
CONTAMINATION CONTROL SOLUTIONS
Nisso Bldg. No 16, 9F
3-8-8 ShinYokohama, Kohoku-ku
Yokohama, Kanagawa 222-0033
Tel: +81-45-477-5570
Fax: +81-45-477-5571

Brooks Automation Ltd.
TAIWANHEADQUARTERS
5F-5, No.32, Tai-Yuen Street
Chu-Pei City
Hsinchu County 302, Taiwan, R.O.C.
Tel: +886-3-552 5258
Fax (G&A): +886-3-552 5255
Fax (Sales): +886-3-552 5200

Brooks Automation Korea, Inc.
#35 Giheungdanji-Ro 121Beon-Gil
Giheung-Gu, Yongin-Si
Gyeonggi-Do, 17086
Korea
Tel : +82-31-288-2500
Fax: +82-31-287-2111

Brooks Automation CCS RS AG
Lohstampfestrasse 11
CH-8274 Tagerwilen, Switzerland
Tel: + 41 71-666-72-10
Fax: + 41 71-666-72-11

Brooks Automation Korea
#35 Giheungdanji-Ro 121Beon-Gil
Giheung-Gu, Yongin-Si
Gyeonggi-Do, 17086
Korea
Tel : +82-31-288-2500
Fax: +82-31-287-2111

Brooks Automation (S) Pte Ltd
51-18-C1 Menara BHL,
57 Jalan Ahmad Shah,
10050, Penang,
Malaysia
Tel: +60 4 3701012
Fax: +60 4 3701015

Brooks Automation (Singapore) Pte Ltd
Blk 5008 Ang Mo Kio Avenue 5
#05-08, Techplace II
Singapore 569874
Tel: +65-6836-3168
Fax: +65-6836-3177

Brooks Automation Ltd.
TAINAN OFFICE
3F., No.11, Nanke 3rd Rd., Xinshi Dist.
Tainan Science Park
Tainan City 74147, Taiwan (R.O.C.)
TEL: +886-6-505-0268
FAX: +886-6-505-5228

Brooks Automation Precise Collaborative Robotics
201 Lindbergh Drive
Livermore, CA 94551
Tel: +1-978-262-2400

Revision History

Revision	ECO	Date	Action	Author
A	EC159653	11/19/2024	Updated for GDS 5.1	M. Ashenfelder

Table of Contents

1. Safety

Safety Setup

Brooks uses caution, warning, and danger labels to convey critical information required for the safe and proper operation of the hardware and software. Read and comply with all labels to prevent personal injury and damage to the equipment.

DANGER

Read the Safety Chapter

Failure to review the *Safety* chapter and follow the safety warnings can result in serious injury or death.

- All personnel involved with the operation or maintenance of this product must read and understand the information in this safety chapter.
- Follow all applicable safety codes of the facility as well as national and international safety codes.
- Know the facility safety procedures, safety equipment, and contact information.
- Read and understand each procedure before performing it.

Authorized Personnel Only

This product is intended for use by trained and experienced personnel. Operators must comply with applicable organizational operating procedures, industry standards, and all local, regional, national, and international laws and regulations.

Explanation of Hazards and Alerts

This manual and this product use industry standard hazard alerts to notify the user of personal or equipment safety hazards. Hazard alerts contain safety text, icons, signal words, and colors.

Safety Text

Hazard alert text follows a standard, fixed-order, three-part format.

- Identify the hazard
- State the consequences if the hazard is not avoided
- State how to avoid the hazard.

Safety Icons

- Hazard alerts contain safety icons that graphically identify the hazard.
- The safety icons in this manual conform to ISO 3864 and ANSI Z535 standards.

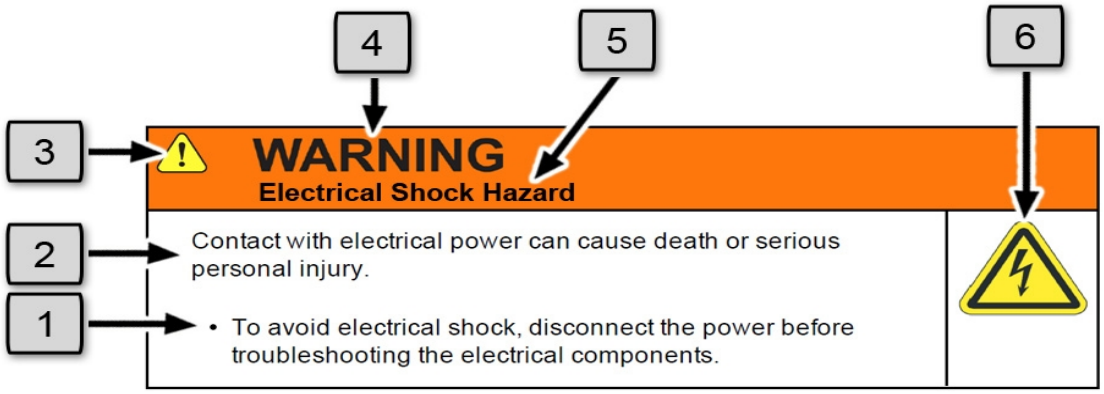
Signal Words and Color

Signal words inform of the level of hazard.

	Danger indicates a hazardous situation which, if not avoided, **will result in serious injury or death**. The Danger signal word is white on a red background with an exclamation point inside a yellow triangle with black border.
	Warning indicates a hazardous situation which, if not avoided, **could result in serious injury or death**. The Warning signal word is black on an orange background with an exclamation point inside a yellow triangle with black border.
	Caution indicates a hazardous situation or unsafe practice which, if not avoided, **may result in minor or moderate personal injury**. The Caution signal word is black on a yellow background with an exclamation point inside a yellow triangle with black border.
	Notice indicates a situation or unsafe practice which, if not avoided, **may result in equipment damage**. The Notice signal word is white on blue background with no icon.

Alert Example

The following is an example of a Warning hazard alert.



Number	Description
1.	How to Avoid the Hazard
2.	Source of Hazard and Severity
3.	General Alert Icon
4.	Signal Word
5.	Type of Hazard
6.	Hazard Symbol(s)

General Safety Considerations

WARNING

Software

Software is not safety rated. Unplanned motion can occur as long as power is supplied to the motors. Maximum torque could be momentarily applied that may cause equipment damage or personal injury.

- Only operate the robot with its covers installed.
- Guarantee that safety controller features are in place (for example, an emergency stop button and protective stop).
- Regularly test safety components to prove that they function correctly.

WARNING

Robot Mounting

Before applying power, the robot must be mounted on a rigid test stand, secure surface, or system application. Improperly mounted robots can cause excessive vibration and uncontrolled movement that may cause equipment damage or personal injury.

- Always mount the robot on a secure test stand, surface, or system before applying power.

WARNING

Do Not Use Unauthorized Parts

Using parts with different inertial properties with the same robot application can cause the robot's performance to decrease and potentially cause unplanned robot motion that could result in serious personal injury.

- Do not use unauthorized parts.
- Confirm that the correct robot application is being used.

WARNING

Magnetic Field Hazard

This product contains magnetic motors that can be hazardous to implanted medical devices, such as pacemakers, and cause personal harm, severe injury, or death.

- Maintain a safe working distance of 30 cm from the motor when with an energized robot if you use a cardiac rhythm management device.

CAUTION

Unauthorized Service

Personal injury or damage to equipment may result if this product is operated or serviced by untrained or unauthorized personnel.

- Only qualified personnel who have received certified training and have the proper job qualifications are allowed to transport, assemble, operate, or maintain the product.

CAUTION

Damaged Components

The use of this product when components or cables appear to be damaged may cause equipment malfunction or personal injury.

- Do not use this product if components or cables appear to be damaged.
- Place the product in a location where it will not get damaged.
- Route cables and tubing so that they do not become damaged and do not present a personal safety hazard.

CAUTION

Inappropriate Use

Use of this product in a manner or for purposes other than for what it is intended may cause equipment damage or personal injury.

- Only use the product for its intended application.
- Do not modify this product beyond its original design.
- Always operate this product with the covers in place.

CAUTION

Seismic Restraint

The use of this product in an earthquake-prone environment may cause equipment damage or personal injury.

- The user is responsible for determining whether the product is used in an earthquake prone environment and installing the appropriate seismic restraints in accordance with local regulations.

Mechanical Hazards

CAUTION

Pinch Point

Moving parts of the product may cause squeezing or compression of fingers or hands resulting in personal injury.

- Do not operate the product without the protective covers in place.

WARNING

Automatic Movement

Whenever power is applied to the product, there is the potential for automatic or unplanned movement of the product or its components, which could result in personal injury.

- Follow safe practices for working with energized products per the facility requirements.
- Do not rely on the system software or process technology to prevent unexpected product motion.
- Do not operate the product without its protective covers in place.
- While the collaborative robotics system is designed to be safe around personnel, gravity and other factors may present hazards and should be considered.

CAUTION

Vibration Hazard

As with any servo-based device, the robot can enter a vibratory state resulting in mechanical and audible hazards. Vibration indicates a serious problem. Immediately remove power.

- Before energizing, ensure the robot is bolted to a rigid metal chamber or stand.

Electrical Hazards

Refer to the specifications of the *Guidance Controller Quick Start Guide* for the electrical power.

DANGER

Electrical Shock Hazard

Contact with electrical power can cause personal harm and serious injury.

- To avoid electrical shock, disconnect the power before troubleshooting the electrical components.
- Check the unit's specifications for the actual system power requirements and use appropriate precautions.
- Never operate this product without its protection covers on.

WARNING

Electrical Burn

Improper electrical connection or connection to an improper electrical supply can result in electrical burns resulting in equipment damage, serious injury, or death.

- Always provide the robot with the proper power supply connectors and ground that are compliant with appropriate electrical codes.

WARNING

Electrical Fire Hazard

All energized electrical equipment poses the risk of fire, which may result in severe injury or death. Fires in wiring, fuse boxes, energized electrical equipment, computers, and other electrical sources require a Class C extinguisher.

- Use a fire extinguisher designed for electrical fires (Class C in the US and Class E in Asia).
- It is the facility's responsibility to determine if any other fire extinguishers are needed for the system that the robot is in.

NOTICE

Improper handling of the power source or connecting devices may cause component damage or equipment fire.

- Connect the system to an appropriate electrical supply.
- Turn off the power before servicing the unit.
- Turn off the power before disconnecting the cables.

Ergonomic Hazards

CAUTION

Heavy Lift Hazard

Failure to take the proper precautions before moving the robot could result in back injury and muscle strain.

- Use a lifting device and cart rated for the weight of the drive or arm.
- Only persons certified in operating the lifting device should be moving the product.

CAUTION

Tipover Hazard

This product has a high center of gravity which may cause the product to tip over and cause serious injury.

- Always properly restrain the product when moving it.
- Never operate the robot unless it is rigidly mounted.

CAUTION

Trip Hazard

Cables for power and communication and facilities create trip hazards which may cause serious injury.

- Always route the cables where they are not in the way of traffic.

Emergency Stop Circuit (E-Stop)

The integrator of the robot must provide an emergency stop switch.

WARNING

Emergency Stop Circuit

Using this product without an emergency stop circuit may cause personal injury.

- Customer is responsible for integrating an emergency stop circuit into their system.
- Do not override or bypass the emergency stop circuit.

Recycling and Hazardous Materials

Brooks Automation complies with the EU Directive 2002/96/EU Waste Electrical and Electronic Equipment (WEEE).

The end user must responsibly dispose of the product and its components when disposal is required. The initial cost of the equipment does not include cost for disposal. For further information and assistance in disposal, email Brooks Automation Technical Support at support_preciseflex@brooksautomation.com.

2. GDS Overview

GDS Introduction

Guidance Development Studio (GDS) is a software programming environment that enables users to develop and debug the Guidance Programming Language (GPL) that controls the robot. Output resources include GPL software projects and GP Flow Sequences.

GDS runs on a Windows PC. Connect the PC to a Guidance Controller locally or remotely via Ethernet to execute and compile programs. The PC does not need to be connected to a Guidance Controller for offline development.

Minimum PC Requirements for GDS

Executing these examples requires:

- A 500 MHz or faster PC running a Windows 10 system or later
- A web browser
- A 10/100 Ethernet card for the PC
- At least 100 MB of space on the PC's disk
- Access to the Brooks support website (https://www.brooks.com/support/brooks-preciseflex-support/)

WARNING

Before proceeding with this GDS Guide, perform the following steps:

- Properly mount the robot
- Install and test all required safety interlocks
- Connect the power

Prior to reading this document, set up the controller as detailed in Controller Setup.

A thorough understanding of GPL is not necessary to use the examples in this manual, but for additional information about GPL functions and their associated syntax, see the *Guidance Programming Language User Manual*. All GPL statements and classes with their methods and properties are described in the *Guidance Programming Language Dictionary*.

Brooks Video Tutorials

Brooks has a library of detailed video tutorials for GDS topics that include:

- Connecting to Controllers
- GPL Programming
- GP Flow Programming
- IntelliGuide Vision Configuration

Visit https://www.brooks.com/support/brooks-preciseflex-support/guidance-development-studio/ to watch the videos.

3. Installation and Startup

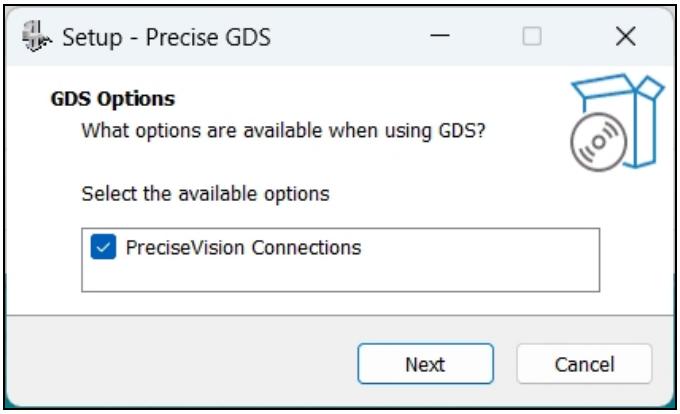
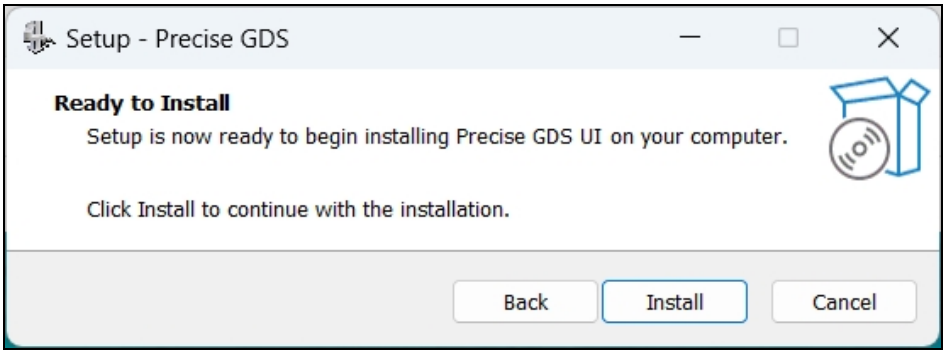
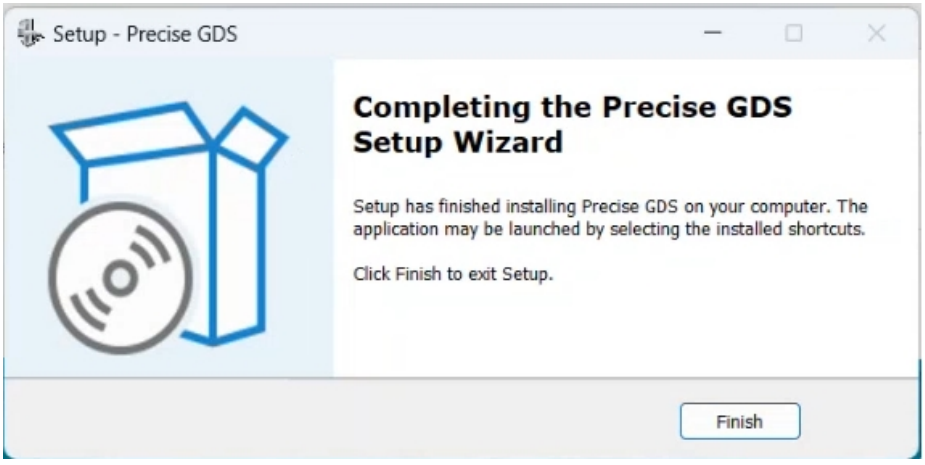
Uninstalling Previous Versions of GDS

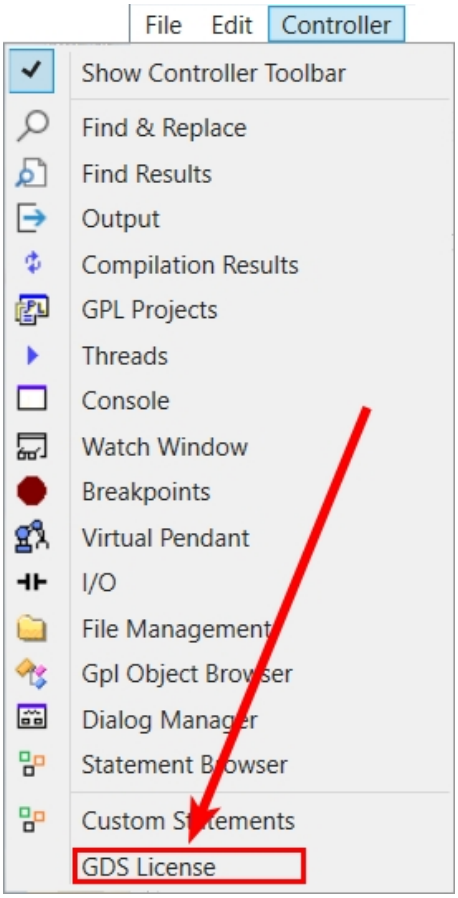
Perform the following steps to uninstall an old version of GDS.

Step	Action
1.	Shut down all programs that are running including virus protection programs
2.	Bring up the Window's Control Panel by clicking **Start > Settings > Control Panel**.
3.	Depending on your version of Windows, these instructions may vary, but the "uninstall" function will be similar in each version. In Windows 11, double-click on **Programs > Uninstall a Program**.
4.	Scroll down the list of programs, and select **Precise GDS xx** .
5.	Right-click the program, select **Uninstall** and click **Yes** to confirm the action.

Installing GDS

To install GDS and all its components on your computer, shut down all programs that are running, including virus protection programs, double-click the installation file, and follow the instructions.

Step	Action
1.	When the installation file opens, in the *GSS Options* window, PreciseVision Connections should be selected. Click **Next**. 
2.	In the *Ready to Install* window, click **Install**. 
3.	If the installation is successful, the *Completing the Precise GDS Setup Wizard* window will display. Click **Finish**. 

Step	Action
4.	Open Windows' Apps menu, scroll to the Precise GDS folder, and start GDS.
5.	In the Main Menu Bar, open the **Controller** drop-down menu, and select **GDS License** to register the GDS license. 

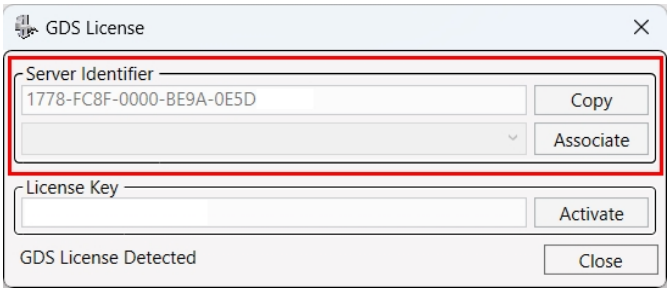
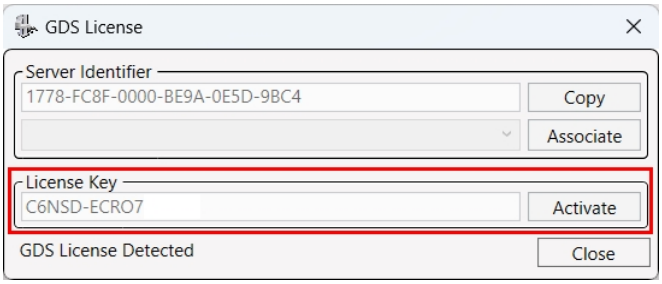
Acquiring and Activating the GDS License

All applications within GDS are functional for 30 days *without* the license being activated. The GDS license enables the user to use GDS to develop software for an unlimited number of controllers. However, the package is only licensed to execute on a single PC.

> ***NOTE:*** Whether the PC is connected to or disconnected from a controller, the GDS license is associated with the PC GDS is running on.

To complete the software activation process, follow the procedure below and send your PC information to sales_preciseflex@brooksautomation.com. Executing the license-activation process in a single GDS component will activate all components.

Follow the procedure below to acquire and activate the GDS license.

Step	Action
1.	In GDS, select **Controller > GDS License**.
2.	Copy the *Server Identifier* to submit to Brooks. ***NOTE:*** *If the Server Identifier is empty or blank, select a network adapter. See Changing the Computer's IP Address and Subnet Mask for more information on changing adapter options. When the adapter is selected, click* ***Associate****. The built in Ethernet port is the best device for association. The selected network adapter should be hardware that can not be removed from the PC. Copy the Server Identifier when you are finished.* 
3.	Email the *Server Identifier* number to sales_preciseflex@brooks.com to start the order process for a GDS License Key.
4.	When you receive the GDS License Key, enter it into the *License Key* field and click **Activate**. 

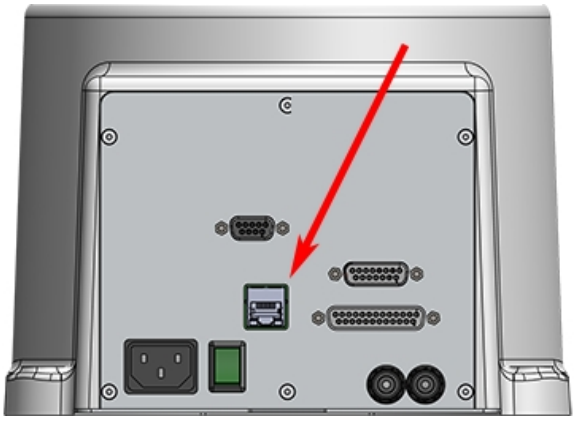
NOTE: You must run the application in "Administrator Mode" in order for the license to be successfully applied.

Connecting the Controller to a PC

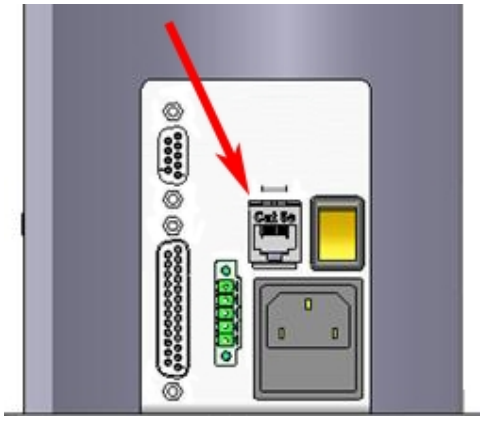
NOTE: You can run GDS without a controller, and you can edit local Projects. However, you need a controller to compile and execute projects. Any controller is OK as long as it runs GPL, but for this manual we refer to the PreciseFlex Guidance Controller.

The Guidance Controller includes all the software necessary to operate the robot. However, a PC or other computer or tablet must be used as the graphical user interface (GUI) to manually operate and program the robot.

Connect the controller to a computer directly or via a hub or switch. If connecting to a robot or Guidance Controller, plug the cable into the RJ45/Ethernet port on the Facilities Panel (shown below). The user can plug the cable into any of the controller's RJ45/Ethernet ports if directly connecting to a controller.



PreciseFlex c10/c8A Facilities Panel

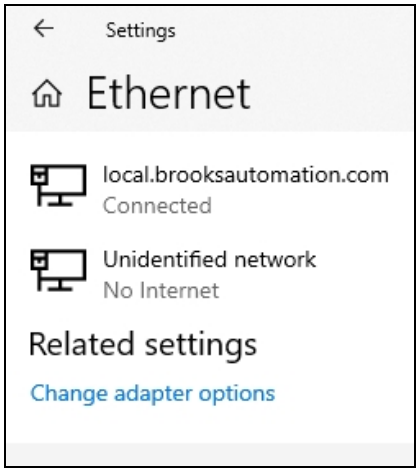
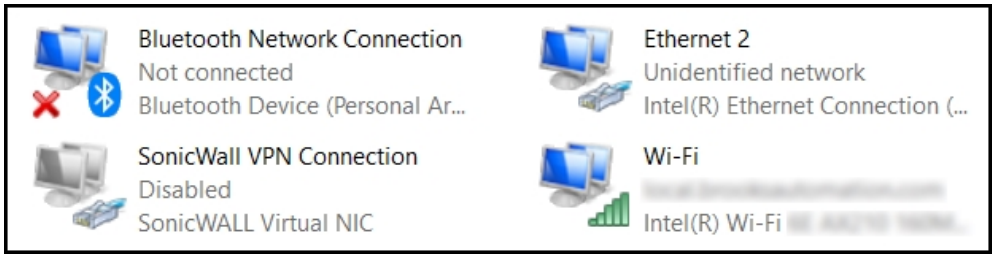
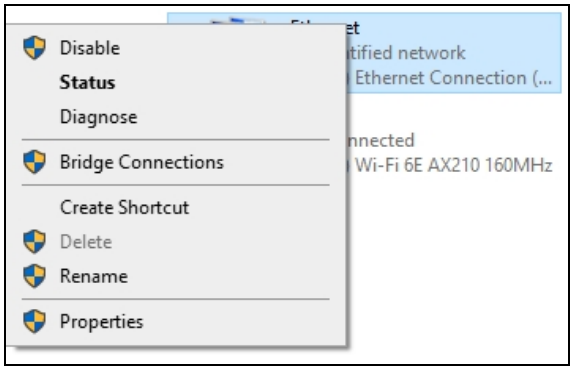


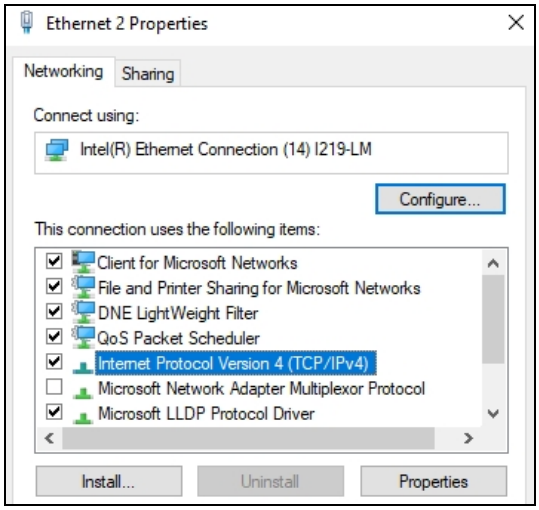
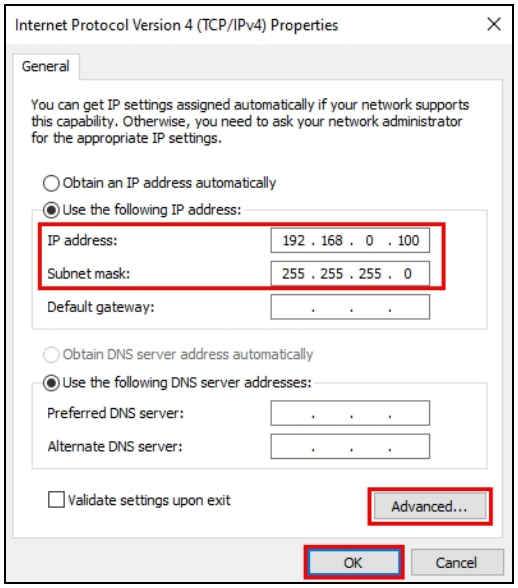
PreciseFlex 400/3400 Facilities Panel

Changing the Computer's IP Address and Subnet Mask

NOTE: This procedure is applicable to Windows 10.

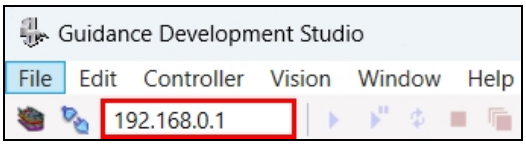
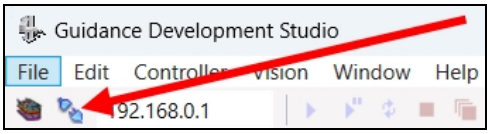
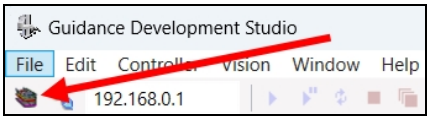
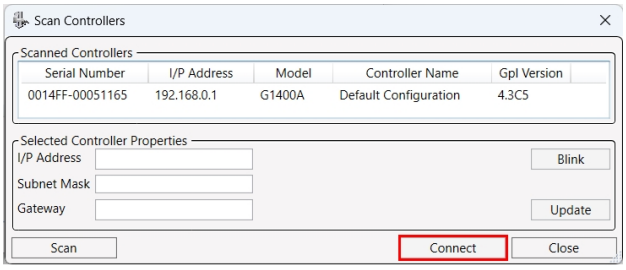
By default the Guidance Controller is factory configured with IP address 192.168.0.1 and subnet mask 255.255.255.0. The computer must be compatible with this IP information. To change the computer's IP address and subnet mask, perform the following procedure:

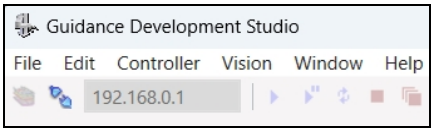
Step	Action
1.	Go to **Start** > **Settings** > **Network & Internet** > **Ethernet** and click **Change adapter options**. 
2.	Right-click the Ethernet card connected to the controller. 
3.	Select **Properties**. 

Step	Action
4.	Select and highlight **Internet Protocol Version 4 (TCP/IPv4)** and click **Properties**. 
5.	If the PC is using Dynamic Host Configuration Protocol (DHCP) and is not compatible with the controller, select the **Use the following IP address** option, and enter a compatible IP address and subnet mask. A compatible IP address is 192.168.0.XXX.  Once connected to the controller, the IP address and subnet mask of the controller can be changed to be compatible with a network.
6.	If the PC is using a static IP address, the IP address does not need to be changed. Click **OK**.

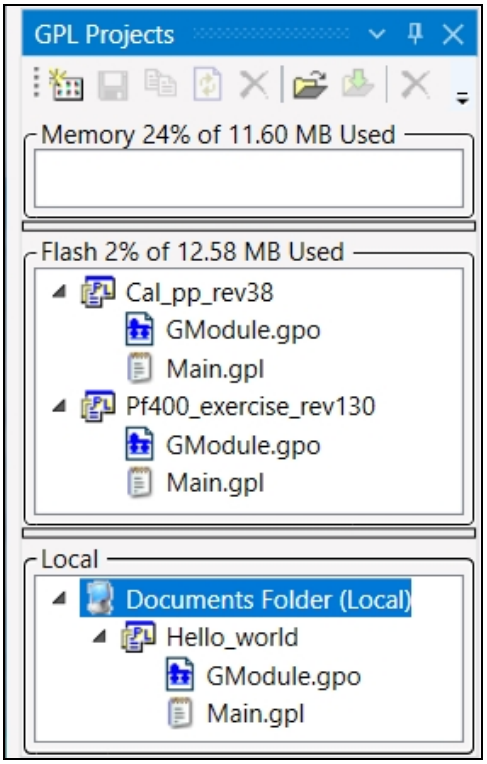
Connecting to the Controller

To execute GPL code on the controller, establish an Ethernet connection between the PC and the controller. Follow this procedure:

Step	Action
1.	In the toolbar **IP or Host Name** field, verify that the IP address for the controller is correct. By default, Guidance controllers are configured to respond to 192.168.0.1. 
2.	Click the **Connect/Disconnect** icon to connect. 
3.	If more than one controller is available on your network, you can scan for all available controllers. Click the **Scan for controllers** toolbar button below the File menu. For more details, see "Changing the Controller's IP Address and Subnet Mask ." 
4.	Select a controller from the list, and click Connect. 

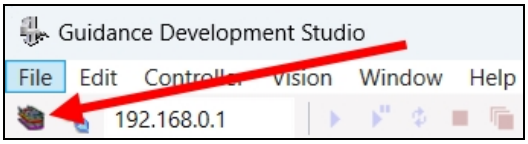
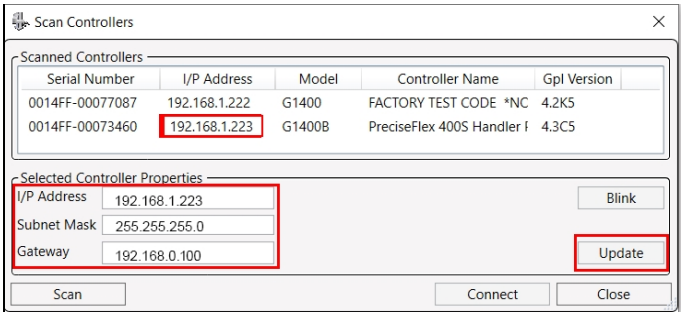
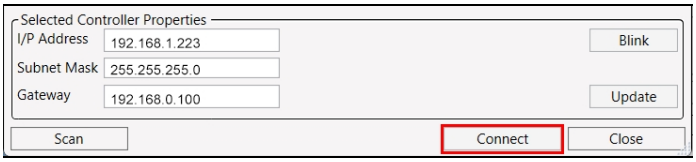
Step	Action
5.	When the controller is connected, the IP address will be grayed out. 

Once connected, the *GPL Projects* window will display files in memory, flash, and stored locally.



Changing the Controller's IP Address and Subnet Mask

To change the controller's IP address, subnet mask, and gateway, follow the procedure below.

Step	Action
1.	Click **Scan for Controllers** in the *Guidance Discovery* screen. 
2.	In the *Scan Controllers*pop-up window, select a controller and enter the IP address, subnet mask, and optionally the gateway address. Click **Update**. 
3.	Click the **Connect** button on the connection pop-up window to connect to the controller. 

4. Windows and Projects

Windows

GDS features specialized windows for editing three different types of programming

- Guidance Programming Language (GPL) - the computer language for programming the robot
- Vision - camera-oriented instructions and data for the IntelliGuide Vision Gripper
- GP Flow - an interface that enables the user to create a high-level programming sequence without text-based coding

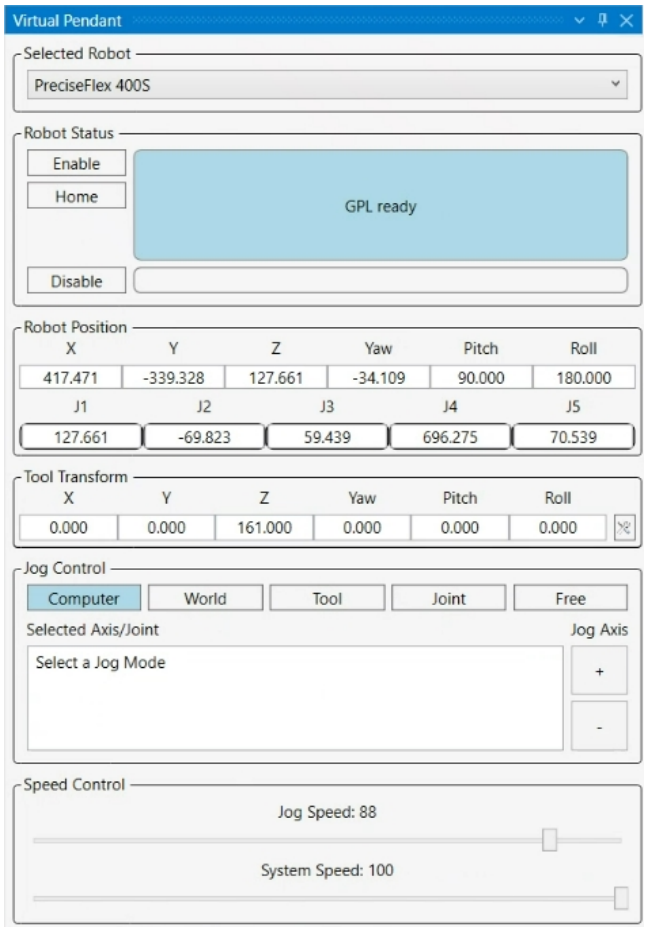
Projects

A project is the basic executable entity in GDS. It consists of two or more text files stored within a single folder/directory.

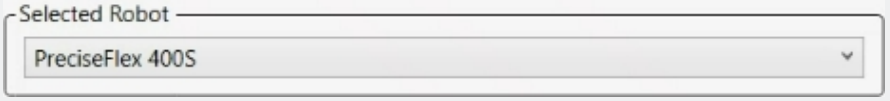
Virtual Pendant Window

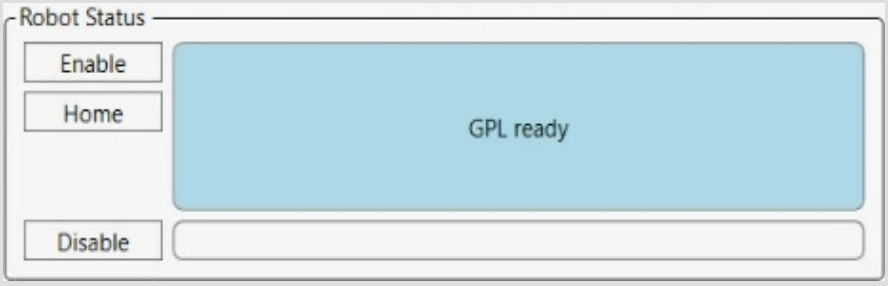
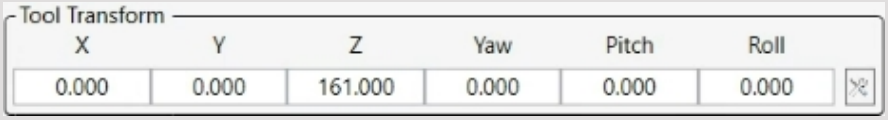
GDS includes a *Virtual Manual Control Pendant* that enables the robot to be manually jogged in the same manner as the VMCP built into the controller's web browser interface. It displays the current position of the robot as well as the tool transformation that is currently in effect. The pendant features are:

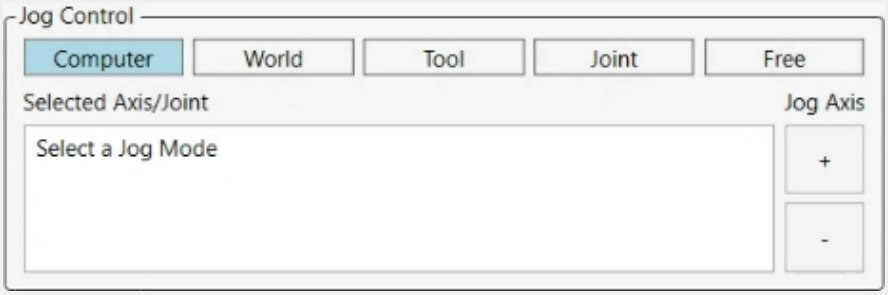
- Selected Robot
- Robot Status
- Robot Position
- Tool Transform
- Jog Control
- Speed Control



Below are descriptions of each pendant section.

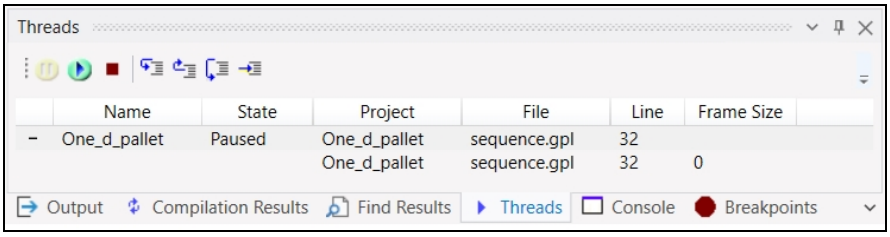
Pendant Section	Description
Selected Robot	 Open the drop-down menu to select a robot connected to your system.

Pendant Section	Description
Robot Status	 **Robot Status** indicates the system state and whether the robot is homed. The **Enable** button enables power to the motors and turns on the power supply to the amplifier for the controller. Click the **Home** button before using the robot. This enables each axis to be set to an absolute position that allows formerly taught positions or location to be repeated. ***NOTE:*** *If your are using a PreciseFlex robot, nothing will move except for maybe the gripper fingers.* The **Disable** button cuts power to the motors.
Robot Position	 **Robot Position** displays each of the robot's joint positions and the current position and orientation of the robot's tooltip. Distances display in millimeters and angles display in degrees.
Tool Transform	 **Tool Transform** is an offset applied to the position and orientation of the last joint of the robot. That point represents the point of the robot that will be aligned when moving to a position, the tooltip of the robot.

Pendant Section	Description
Jog Control	 **Jog Control** enables the user to move the robot in various modes manually. **Computer** mode enables a Project to take control of the robot. Select **Computer** mode when finished moving the robot in a jog mode of **World**, **Tool**, **Joint**, or **Free**. **World** moves the robot's tooltip in a straight path along or rotates around the X, Y or Z axes of the robot's base reference frame. **Tool** mode is similar to **World** mode, except that the **Tool Transform** determines the reference frame to move the tooltip along or rotate it. **Joint** mode moves individual joints one at a time. Even if a joint is beyond its limit stops, this mode can be used to drive the joint back into the operating range. **Free** mode enables one or more joints to be removed from position control mode to allow the axis to move freely. For lightweight robots, **Free** allows the operator to grip an axis and manually reposition it. The user can free multiple joints simultaneously without using the **Jog Speed** setting highlighted yellow on the Virtual Pendant. Each time an axis is selected and the plus (+) button is clicked, the axis is freed until the user clicks the minus (-) button to place the axis back into position control mode
Speed Control	 **Jog Speed** is the speed at which the robot moves in **World**, **Tool** or **Joint Jog Control** modes, from 0-100% of the max Jog Control speed. **System Speed** is the speed at which the robot moves in **Computer** control mode from 0-100% of the max Computer control speed.

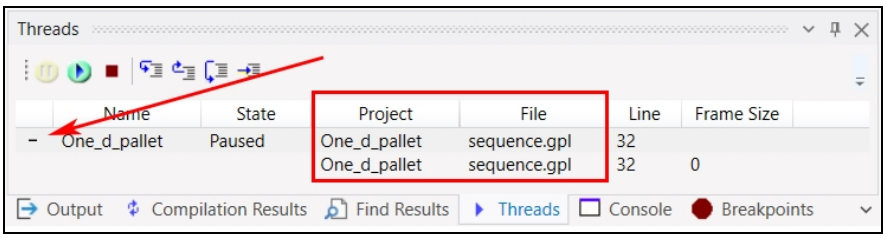
Threads Window

The *Threads* window displays status information for each active execution thread in the controller. The main procedures for your Project will always run in their own thread. In addition, more complex applications may initiate additional threads to enable independent execution of selected code segments. A sample of the *Threads* window is shown below where each top-level line displays the information for a different execution thread.



The *Name* column specifies the name of the thread. The thread name is normally the same as the Project name. *State* indicates if the thread is running or has ceased execution for some reason (e.g. paused due to a breakpoint or error). *Project* displays the name of the Project running in the thread. If the thread has ceased execution, *File* indicates the name of the file that contained the last procedure executed and *Line* indicates the line number of the last step executed relative to the start of the file.

The **program stack** window displays the list of procedures that are currently on the execution stack for the thread specified in the Threads box. The program stack information is displayed in the *Threads* window.



When a procedure is running, information on the current statement being executed is saved on the execution stack. When a procedure calls another procedure, information about the current statement in the calling procedure is preserved on the execution stack and a new "frame" is created on the stack to store the step information for the called procedure.

These frames are displayed under each thread in the thread display window. Each line displays the information for a single stack frame. So, the number of lines indicates the depth of procedure calls currently in effect. Note, this window will only display information when the referenced thread is active but not running.

If your program is paused, you can click the Plus (+) or Minus (-) symbols to the left in the window below, and it will show you the stack of the programs that are being executed on that thread.

When a procedure is running, information on the current statement being executed is saved on the Execution Stack. When a procedure calls another procedure, information about the current statement in the calling procedure is preserved on the Execution Stack and a new "frame" is created on the stack to store the step information for the called procedure.

These frames are displayed under each thread in the thread display window. each line displays the information for a single stack frame. So, the number of lines indicates the depth of procedure calls currently in effect.

> ***NOTE:*** This window will only display information when the referenced thread is active but not running.

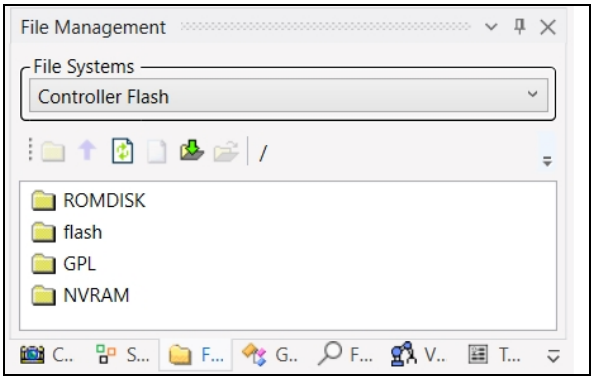
File Management Window

The *File Management* window provides access to the entire file structure of the controller or vision system. Use the File Systems drop-down menu to switch between the two. This is wider access than is available via the *GPL Projects* window, which only displays the Project area of the flash disk and the controller's memory.

The entire file structure includes:

- all folders on the flash, such as the Parameter Database PAC files in the "/config" folder
- the ROMDISK (i.e. the in-memory simulated disk) that contains the Operator Interface web pages
- the GPL memory image that displays the Projects loaded in the controller's memory.
- Vision Projects
- Logged Images

The *File Management* window is provided as a convenience for displaying files not accessible using the *GPL Projects* window, such as the "Project.gpr" files.

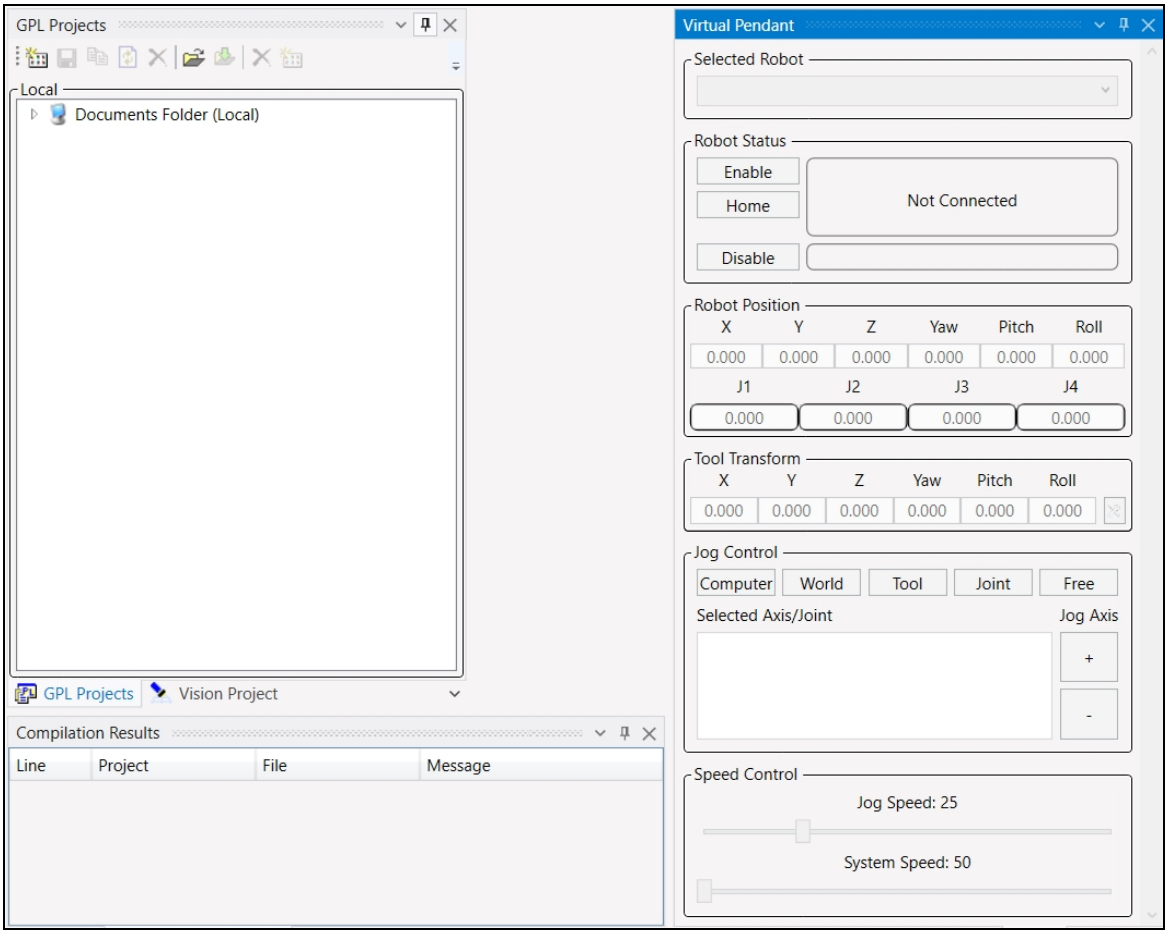


You can double click on a file to open a read-only copy within the GDS editor region.

5. GDS

GDS Screen Layout

After connecting to GDS, a master desktop will display. Below is an example. Desktop layouts will vary depending on the arrangements of windows and where they are docked. GDS remembers the last layout that was created, and it restores that layout when you re-open GDS.



The GDS window contains:

- A title bar that displays the IP address of the connected controller
- A top menu bar
- A tool bar
- The main editor/debugger area
- A variety of (dockable) windows

Each of the dockable windows can be displayed or hidden, resized and repositioned into the arrangement that is most efficient for your use. Any space not occupied by a dockable window is utilized by the editor/debugger.

To reposition a window, click on its title bar and drag it to its new location. If you drag a window on to another window, they can split the space or share the space using tab controls.

If you click on the "push pin" in the top right of a window's title bar, you will either "pin" a window and fix its location or "un-pin" a window so it can share its space with another window. Windows can be resized by grabbing a border and stretching or shrinking it to the desired dimension.

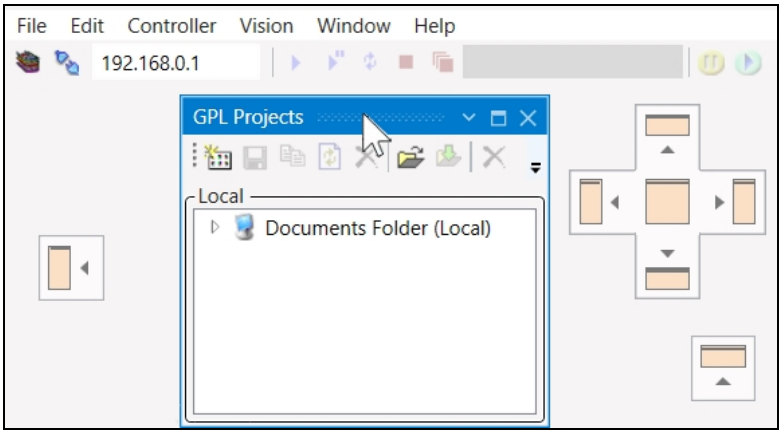
In the following sections, the functions available from each of the major components of GDS (i.e. main menu bar, tool bars, and windows) will be described. The components are presented in order of importance rather than alphabetically. The following table briefly summarizes each of the components.

Component	Description
Main Menu Bar	Main menu bar that provides access to most of the functions provided by GDS.
Main Toolbar	Provides quick access to common editing and debugging functions.
GPL Projects	Displays and manages Projects that are resident in the controller's memory and in the flash disk and the PC's hard drive in the standard GPL Project areas.
Output Window	Displays all output generated by the controller in connection with the execution of GPL Projects and system messages.
Compilation Results	Displays information about the compilation of GPL programs being edited
GPL Object Browser Window	Provides a list of the methods and properties of all GPL Classes. Automatically displays abbreviated help information as text is entered.
Editor/Debugger Window	Main window for editing and debugging procedures and global motion data.
Virtual Pendant Window	Permits the robot to be manually jogged and displays the current robot position.

Component	Description
Threads Window	Displays all threads currently executing and a list of procedures on the execution stack for each GPL thread.
File Manager Window	Displays and manages all folders and files on the controller's flash disk.
Console Window	Provides access to the controller's console. Enables GPL Console Commands to be entered and executed.
Find & Replace	Utility for finding text references across all GPL Projects.
Watch Window	Enables viewing and changing variables for a running application
Breakpoints	Display and manage all breakpoints set on the controller
I/O	Display all digital and analog signals on the controller
Dialog Manager	Display the output of any applications invoking the Controller Show Dialog operations
Statement Browser	Display all statements available on the system
Custom Statements	Display the list of custom statements imported into GDS

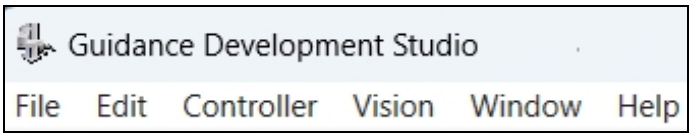
GDS Window Positioning

GDS windows can be moved around and docked. Click the mouse cursor inside the blue top bar, and drag the window around. The positioning icons will display. Drag the window onto a positioning icon and to where you want to dock it on the screen.



Main Menu Bar

The main GDS menu bar contains drop-down menus with options to execute the majority of the GDS functions. Many of the more common functions are provided via the toolbars or within the dockable windows. The following tables describe the operation of selections in the drop-down menus.



File Menu

Menu Item	Description
Save	Saves the file being edited back into its Project folder. The folder can be in memory, in the flash, or on the PC hard drive. Changes to files in memory do not take effect until the file is saved back to memory.
Save All	Save all Projects
New Project	A new Project is created with the name specified in a subsequent pop-up dialog box. That pop-up also permits you to select whether you wish to create the Project in the controller's memory, in the flash disk, or in the PC's hard drive. This operation is equivalent to the "Create a new project file" operation in the GPL Projects window.
Copy Project	Make a copy of the selected Project
Discard Changes	Discard any modifications made to the selected Project that have not been saved
Delete Project	Delete the currently selected Project
Import Project Directory	Import a Project from a directory
Import From ZIP	Import a Project from a ZIP file
Export Project Directory	Export a Project to a directory
Export To ZIP	Export a Project to a ZIP file
New File	Add a new file to the currently selected Project
Delete File	Delete the currently selected file
Scan for Controllers	Displays the *Scan Controllers* pop-up window that enables you to connect GDS to a Guidance controller.

Edit Menu

Menu Item	Description
Undo	Standard undo and redo functions that reverse the effect of the previous editing operations or re-instate the effect of operations that were undone.
Redo	
Cut	Standard cut, copy, and paste functions that operate on the selected text within the GPL editor.
Copy	
Paste	

Controller Menu

Menu Item	Description
Show Controller Toolbar	Show or Hide the controller toolbar.
Find & Replace	Opens the Find and Replace window
Find Results	Opens the Find Results windows
Output	Opens the GPL Output Window. Displays all output generated by the controller in connection with the execution of GPL Projects and system messages.
Compilation Results	When you compile a project or scan it for errors, the system will provide you with notifications about any issues it detects.
GPL Projects	Opens the GPL Projects Window. Displays and enables management of projects residing in the controller's memory and in the flash disk and the PC's hard drive in the standard GPL Project areas.
Threads	Opens the Threads Window. Displays all threads currently executing and a list of procedures on the execution stack for each GPL thread
Console	Opens the Console Window Enables GPL Console Commands to be entered and executed.
Watch Window	Enables viewing and changing variables for a running application
Breakpoints	Display and manage all breakpoints set on the controller
Virtual Pendant	Displays the Virtual pendant window. Enables the robot to be manually jogged and displays the current robot position.
I/O	Display all digital and analog signals on the controller
File Management	Opens the File Manager Window. Displays and enables management of all folders and files on the controller's flash disk
GPL Object Browser	Displays the Object Browser Window. Provides a list of methods and properties of all GPL Classes. Automatically displays abbreviated help information as text is entered.

Menu Item	Description
Dialog Manager	Display the output of any applications invoking the Controller Show Dialog operation
Statement Browser	Display all statements available on the system
Custom Statements	Display the list of custom statements imported into GDS
GDS License	Displays the status of the GDS license

Vision Menu

Menu Item	Description
Show Vision Toolbar	Displays the Vision Toolbar
Vision Project	Displays the Vision Projects window
Cameras	Shows the Cameras connected to the vision system
Tool Properties	Displays theTool Properties for the selected tool in the vision projects
Camera Configuration	Enables you to change the resolution to the Camera Configurationon your system
Vision License	Displays the status of the Installing a Vision License
Classifier Models	Enables you to see, edit, and train the Classifier Models on the vision server
Stereoscopic Arm Camera	Performs a new stereoscopic calibration of the front-facing or downward-facing cameras of IntelliGuide Vision Grippers.
Manage Calibration	Manage Calibrationsenables you to copy calibration data between the controller and the vision server.

Window Menu

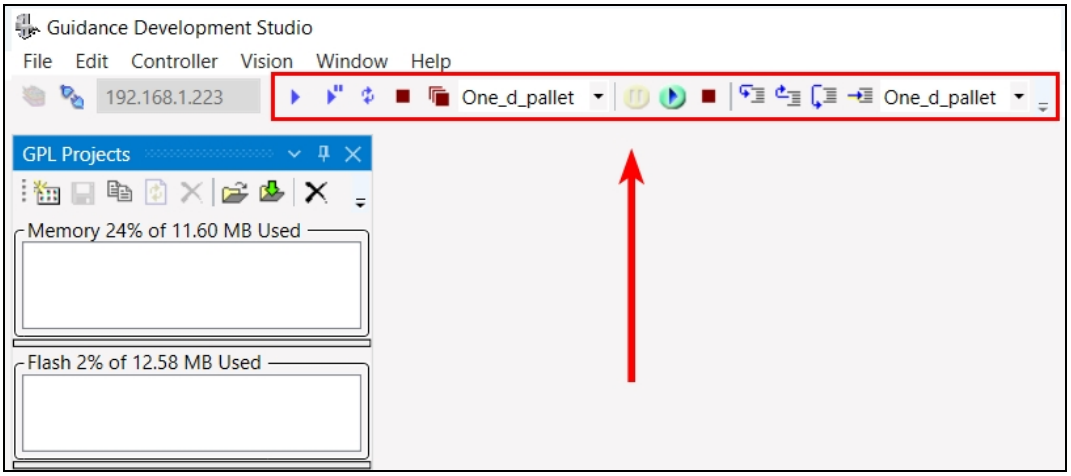
Menu Item	Description
Close All Documents	Closes all open documents
Cascade	Displays all open documents as tabs side by side
Tile Horizontally	Organizes and displays all open documents and windows horizontally
Tile Vertically	Organizes and displays all open documents and windows vertically

Help Menu

Menu Item	Description
Contents	Opens a searchable Help library
About	Displays the GDS version and ID information along with build information for key components of GDS.

Main Toolbar

The *Main Toolbar*at the top of the GDS window provides quick access to a number of commonly used functions.

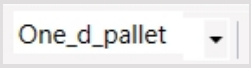


The *Main Toolbar* ends to the right at the purple **Connect/Disconnect** button (shown below), which applies to the vision toolbar.

Some toolbar icons may become available or unavailable depending on the state of operation the system is in. For example, while a Project is running, the **Thread** button becomes available.

The functions available on the main *GDS Toolbar* are displayed below.

GDS TOOLBAR BUTTONS

Button	Action
	Start selected Project
	Start selected Project with break
	Compile selected Project
	Stop the selected Project
	Stop all Projects
	The selected Project
	Pause selected thread
	Run/resume selected thread
	Stop selected thread
	Step Into on selected thread
	Step out on selected thread
	Step over on selected thread
	Set current line in selected thread
	Selected thread for toolbar functions

6. GPL

GPL Projects Overview

Definition of a Project

In GPL, rather than executing a "program," a "Project" is the basic executable entity. A Project consists of two or more text files stored within a single disk folder (i.e. directory). Each file is a standard human-readable ASCII file. The folder name and the Project name are synonymous. Project names conform to the standard GPL symbol name convention and therefore must start with an alphabetic character or an underscore (_) and cannot be a single underscore (_).

The first character can be followed by any combination of alphanumeric characters and underscore (_). Since Project folders can be stored on the flash disk, Project names are limited to 43 characters in length. Also, since flash disk names are case sensitive, the first alphabetic character in the Project name is always upper case and all other alphabetic characters are lower case, e.g. "Test_ project." Specific operations within GPL and GDS are provided for loading, compiling, and executing a Project.

File Types

Project.gpr

The file "Project.gpr" must always be present in each Project folder and is referred to as the "Project File." This file contains information on the other files within the Project. It is a sort of manifest. For example, the Project File stores the name of the procedure that is invoked when the Project begins execution. GDS automatically manages the contents of this file and so it is normally hidden within GDS. The contents of a "Project.gpr" file are viewable with a text editor. If you load a Project into memory and run it, GPL reads the "Project.gpr" file first before it starts.

.gpl Files

There can be multiple GPL source files within a Project. Each source file has a "gpl" extension. These files each can contain one or more program modules, which can contain multiple variable declarations and procedures.

.gpo Files

In addition, a Project can contain one, several, or no files with a "gpo" extension, a type of GPL code file. This type of file stores a global module that is used to define things like global Locations and Profiles. The .gpo file is convenient for saving taught robot Locations and general motion Profiles that are accessible by all procedures within the Project. Almost all of the work done within GDS involves the creation, debugging, and management of the .gpl and .gpo files for a given Project.

.gpp Files

A .gpp file is a password-protected.gpl file. A password is required to open, view, and edit the file. These files can be part of the Project structure.

.gsq Files

A .gsq file is a GP Flow sequence file that enables users to create high-level sequences of instructions without coding. It is part of the GP Flow process. When the **Generate** button is clicked, the system produces the corresponding GPL code for the Project to run. See GP Flow Programming Example: Pick and Place for more information.

Documents and Tools

Files such as .gpl and .gpo are considered "documents" to be created and modified. Tools such as those listed in the Controller drop-down menu are used to create and modify those documents. Many tool windows and toolbars can be moved around the screen and docked around the document for your convenience.

The Project as a Collection of Files

Since a Project consists of a collection of files within a disk folder, loading a Project into memory or copying a Project from memory or between disk units is equivalent to copying a file folder and all of its contents. So, Projects can be managed by dragging and dropping their associated files and folders onto the destination device. Although only one Project can be executed at any time, multiple Projects can be concurrently loaded into memory.

GPL Project Libraries

As a convenience when developing large software Projects or for sharing software modules, GDS supports GPL Project Libraries. This feature permits any Project to reference another GPL Project and utilize its public routines and data. No special operations must be performed to convert a Project into a Library. Any Project can be utilized as a Library. To reference a Project as a Library, the main Project must be modified to add the name of the Library into its Project File using the Project Window within GDS. A main Project can reference multiple Libraries and Libraries can reference other Libraries.

When the main Project is compiled, all the files in its referenced Libraries are logically included into the main Project. If two different main Projects refer to the same Library, the Library files are compiled separately into each main Project. This means:

- The use of shared Library Projects does not save memory.
- Global variables defined in the Library Projects are allocated separately for each main Project, so multiple main Projects cannot share data using global variables in the Libraries.

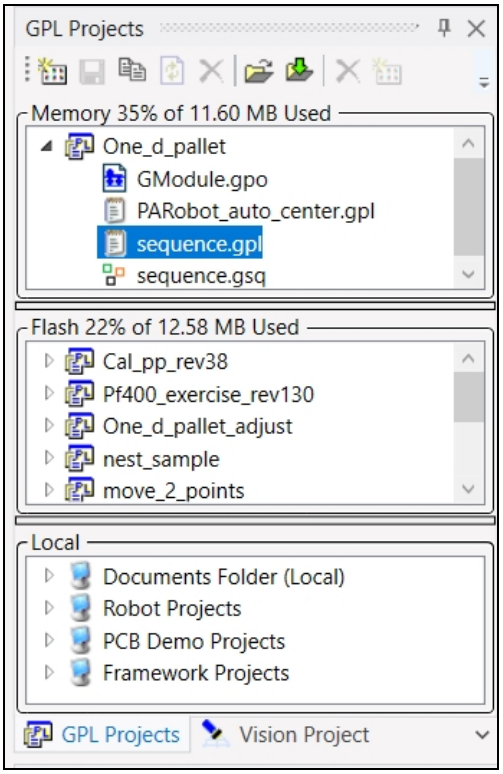
When GDS loads a main Project from flash, it automatically loads any referenced Library Projects from flash. If the Libraries are already loaded, the loaded Libraries are used. If you use GDS to load Libraries from the PC during development, the corresponding Libraries on flash are ignored. When you unload a main Project, the referenced Libraries are not unloaded. If you use GDS to load a main Project from the PC, you must also manually load the Libraries from the PC or from flash.

GPL Naming Conventions

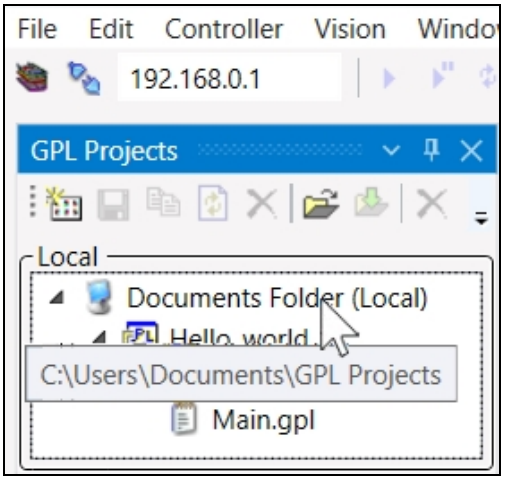
In order to standardize filename and Project-name conventions, GPL differentiates between certain characters and uppercase/lowercase letters in a filename or Project name. For example, GDS will automatically change a forbidden character such as a dash (-) to an acceptable filename character such as an underscore (_). GDS will also change an uppercase letter in a filename to a lowercase letter.

GPL Projects Window

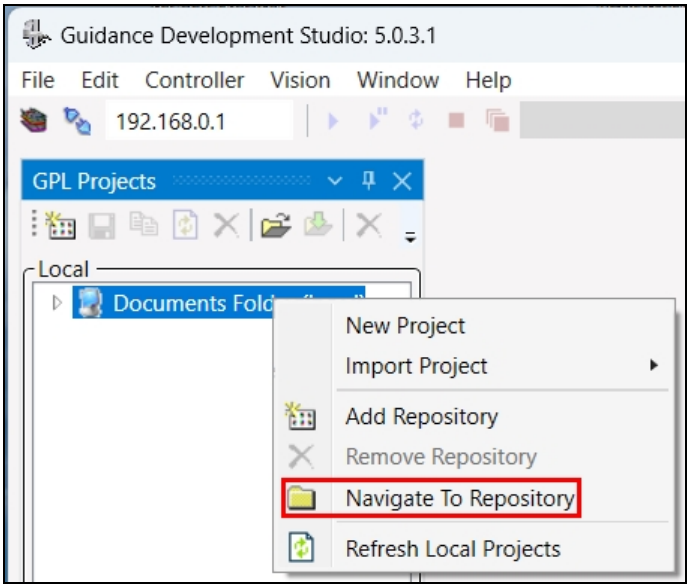
The *GPL Projects* window is one of the most useful windows in GDS. The *GPL Projects* window displays Projects that are loaded in the controller's memory as well as Projects stored in the controller's flash disk and in the PC's hard drive. This window enables Projects to be created, edited, selected for execution, deleted, and transferred between the controller's memory and the disk areas.



- The upper window, the *Memory* window, indicates all Projects loaded into the controller's memory. Once loaded into memory, these Projects are eligible to be executed. The amount of the controller's memory still available for use is displayed. To select a Project for execution or debugging, use the top GDS tool bar.
- The middle window, the *Flash* window, indicates the Projects stored in the controller's flash disk (/flash/projects/) directory.
- The lower window, the *Local* window, shows the Projects stored in GPL Projects areas on the PC's hard drive. If you hover your mouse over the "Documents Folder," the path to the file repository will display.



You could also right-click on "Documents folder," and select **Navigate to Repository** from the menu to go to the file repository.



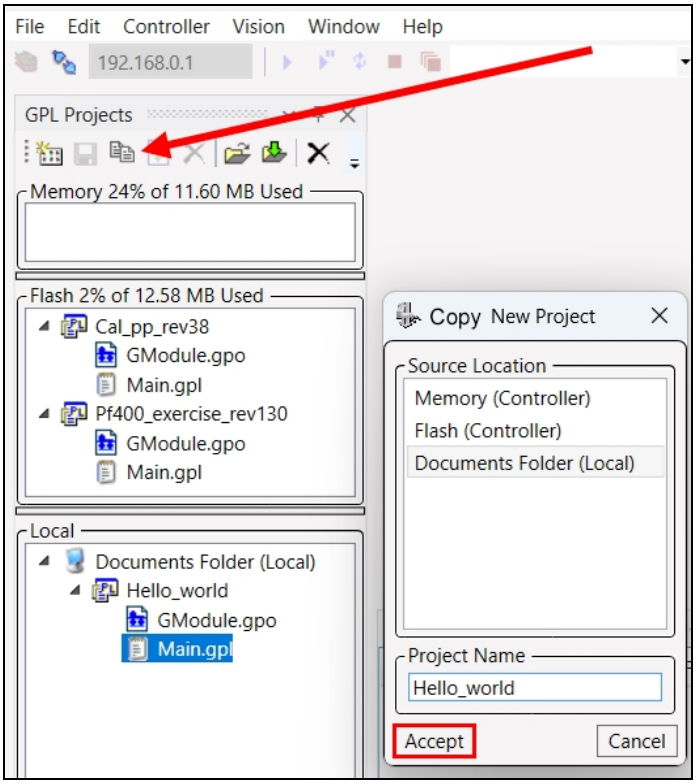
The PC hard drive folder that contains GPL Projects is located in the *my documents* folder. It is called “GPL Projects.” In addition to the local *GPL Projects* folder, users can identify additional folders to map to the local Project settings. This can be done by right clicking in the local Project area and selecting Add Repository. Each specified folder will appear as a top-level node in the lower *GPL Projects* window. This permits multiple hard drive folders to be simultaneously displayed and accessed. To edit a file within a Project, expand the contents of the Project and double click on the file of interest. You can edit Projects stored in memory, in the flash, or on the PC's hard drive.

To copy a Project between memory, the flash, or the hard drive, drag-and-drop the Project between windows. Projects that have been modified while in memory must be dragged-and-dropped to flash or the hard drive if you want to preserve the changes in the event that the controller is powered down.

> ***NOTE:*** If you drag and copy a file from the hard drive into memory, it is still only a *copy*. The file on the hard drive will not be affected by any changes you make to the copy in memory.

If you drag and copy a file from memory to the hard drive, it will overwrite the local copy of that file.

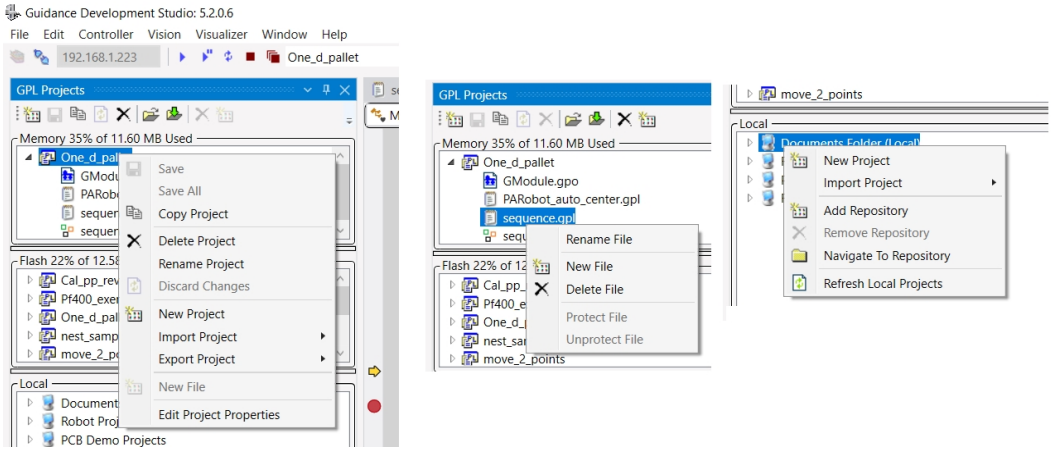
Another way to copy a file between *GPL Projects* windows is to click on the Copy icon in the *GPL Projects* toolbar. It will launch a *Copy New Project* pop-up window from which you can copy a source file and rename it. After you click the **Accept** button, it will leave the copy in memory.



The following table describes the operations available via the *GPL Projects* tool bar.

Icon	Tool Title
	New Project
	Save Project
	Delete Project Or Component
	Copy selected Project or file
	Discard unsaved changes in Project
	Export / Import Project

The image below shows the context menus available when you right-click on a Project, Project file, or local Project archive.

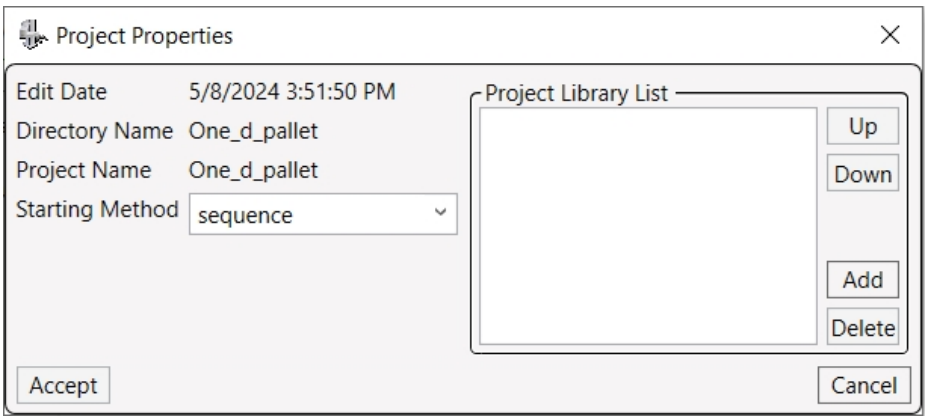


The following table describes the operations available by right-clicking in the *GPL Projects* window.

Right-Click	Description
Project Properties	Displays information from the Project File including the Project name and list of components. Enables the procedure, which is called when the Project is executed, to be changed. (See below)
Import Project Export Project	Permits a Project to be copied between a folder in the PC's file system and the controller's memory, the controller's flash drive, or the hard drive GDS Project area. These functions simplify sharing Projects in a common network drive and are convenient for exchanging Projects via email. In addition, Projects can be dragged from the PC's file system and dropped into the GDS Project area. However, the reverse process is not currently supported.
Edit	Same as the "Edit File" toolbar selection.
New Project	Same as the "New Project" toolbar selection.
Add New Item	Same as the "Add Item to Project" toolbar selection.
Copy Project	Copies or duplicates the selected Project. A pop-up window is displayed that permits the destination and new name of the copy of the Project to be specified.
Delete	Same as the "Delete Project or Component" toolbar selection.
Rename	Renames a single Project file. This function can be operate on files stored in Flash, Memory and on the PC. This operation does not support renaming an entire Project. ***NOTE:*** *Users should only rename files, not projects.*

Right-Click	Description
Protect	To prevent other users from seeing or modifying your GPL source code or other data, you can encrypt ("protect") a file within a Project. Protected files can be executed by anyone but cannot be edited. The protection requires a password that is embedded into the encrypted file. Selecting this operation will prompt the user for a password and a one-line description that is shown if the file is opened in the editor. ***NOTE:*** *Once a file is protected, it can only be unprotected using the same password. Keep a secure backup of the unprotected version of the file in case you forget the password.* This process can only be performed on individual files that are part of a Project that is stored on the PC, not an entire Project. Once a file is protected within a Project, the Project can then be transferred to a controller's Flash or Memory.
Unprotect	Enables a developer to unprotect a previously protected file that is part of a Project. The unprotect function can only operate on files that are stored on a PC and requires the password originally used to protect the file. If a file resides on a controller, it must first be transferred to the PC and then unprotected. This operation cannot be performed on an entire Project, only individual files of a Project.

When **Edit Project Properties** is selected, the *Project Properties* pop-up window displays, which contains information on the files contained within the selected Project.

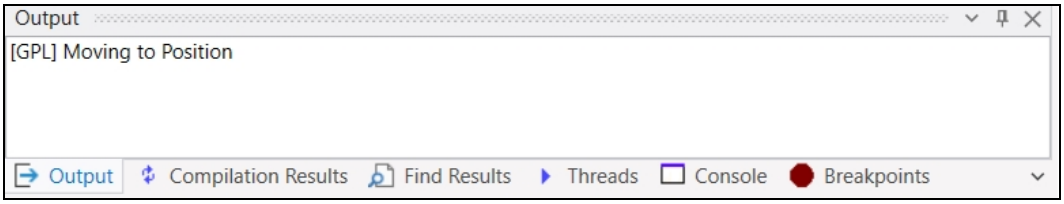


The **Starting Method** specifies the name of any subroutine within the Project that will be executed first when the Project execution begins.

The files contained within each Project Library are compiled into the main Project as though they were explicitly named in the Project File List. If a GPL Project is loaded from the Flash Disk to a controller's memory, any referenced Libraries are automatically loaded. If a GPL Project is loaded from the PC, any referenced Libraries must be manually loaded into a controller's memory.

GPL Output Window

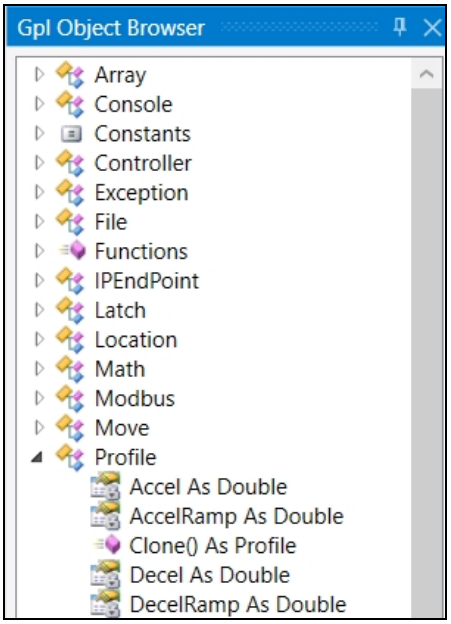
The *GPL Output Window* displays all output from the controller that is generated in connection with executing a GPL Project or output from a user application. For example, when you compile a GPL Project, the output of the compiler including any error messages will be displayed in this window. In addition, if your GPL program generates any text output, e.g. by executing a "Console.WriteLine" method, this output will also be displayed in the *GPL Output Window*.



If you want to **Clear** the contents of this window or **Copy** the contents to a file, right-click anywhere in the window to get a pop-up menu to execute these operations.

GPL Object Browser Window

The Object Browser Window displays syntax and information for all of the GPL statements and class methods and properties. This dockable window operates in a fashion similar to the .NET Object Browser.

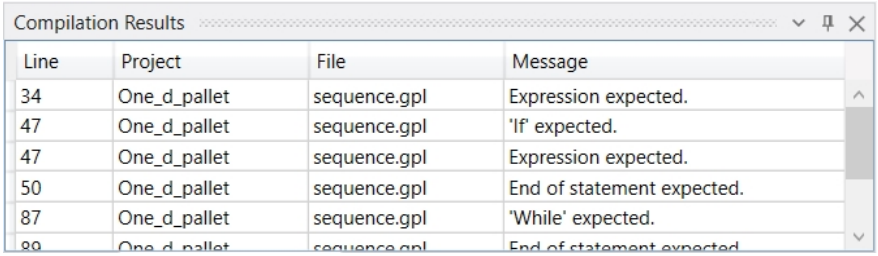


You can browse the tree view in the upper panel of the Object Browser for syntax information on specific statements, methods or properties. An icon to the left of each line provides a quick visual queue to indicate the type of the language element. When you select an item in the top panel, a short description of the language element is displayed in the lower panel.

To access the help page for the item, double-click on an item to navigate to the help topic for that item. This will open the *PreciseFlex Library* at the dictionary page for the selected item. The Object Browser is an information window and source of help information only, so there are no tools associated with this window.

Compilation Window

When a project in memory is compiled, any errors detected will be displayed in the compilation window. You can double-click on any line in the list, and the cursor in the Editor Window will jump to the text line that generated the error.



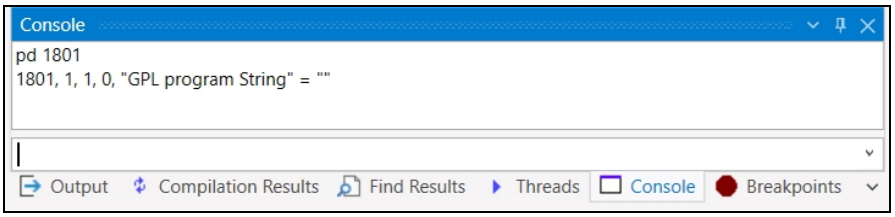
Line	Project	File	Message
34	One_d_pallet	sequence.gpl	Expression expected.
47	One_d_pallet	sequence.gpl	'If' expected.
47	One_d_pallet	sequence.gpl	Expression expected.
50	One_d_pallet	sequence.gpl	End of statement expected.
87	One_d_pallet	sequence.gpl	'While' expected.

Console Window

The *Console* window provides access to the GPL console commands. This window is equivalent to connecting to the serial port of the controller. Console commands are simple, non-graphic text commands that perform rudimentary operations such as displaying the current memory usage.

During normal operation and software development, you should not need to issue console commands, since their functionality is provided by the browser interface and GDS. However, the *Console* window is provided in GDS for completeness.

The *Console* window example below shows where a command has been issued.

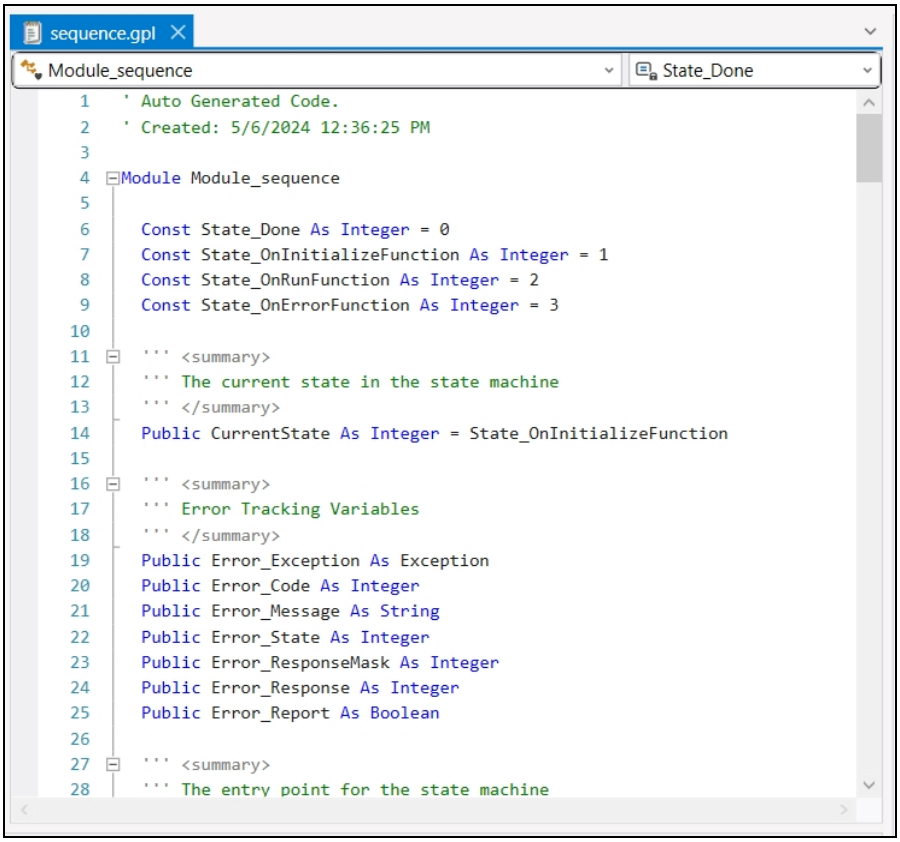


For more information on GPL console commands, see the "Software Reference" section of the *PreciseFlex Library*.

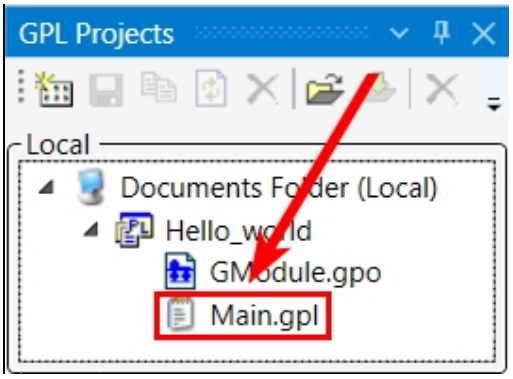
Editor and Debugger Window

Editing a .gpl File

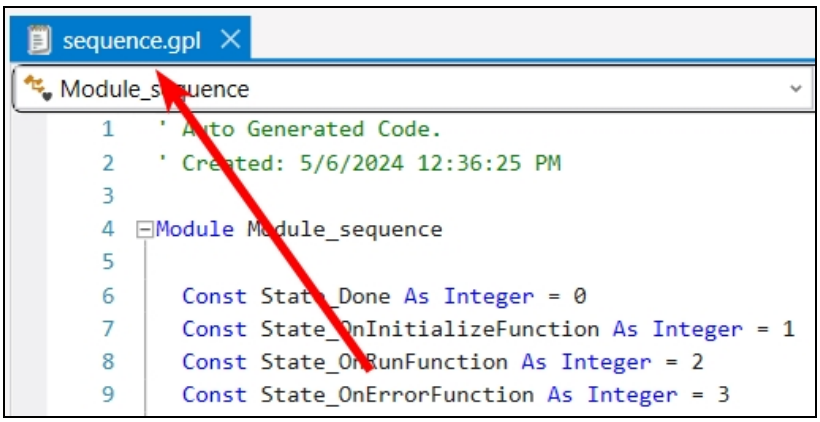
The *Editor and Debugger* window is the primary focus of GDS and occupies all of the space not utilized by the displayed dockable windows. This window enables you to create and modify GPL source files and global GPL motion and program variables, and to debug GPL procedures by single stepping, setting breakpoints, displaying variable values, etc. In its normal source code editing mode, the window will look like this:



To open a file for editing, double-click on a ".gpl" file in the *GPL Projects* window.

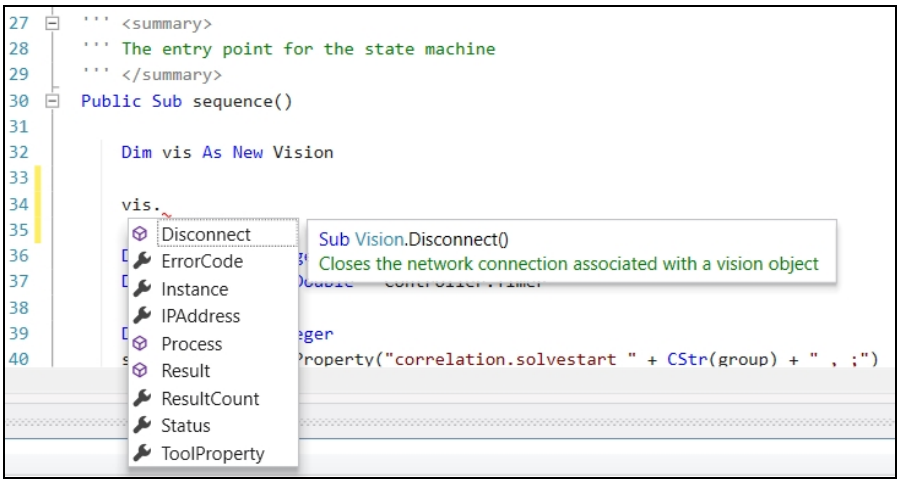


As each file is opened, at the top of the editor window, tabs with the names of opened files are dynamically created to enable you easily switch between files.



You can edit files located in the controller's memory, in the controller's flash disk or in the PC's hard drive. Any files that have been modified and not saved are displayed in bold on the GPL Projects display. Just below the file tabs, two drop-down menus are available for quickly positioning the cursor at a specific module and procedure within a file.

The editor operates in the typical manner for inserting, deleting, cutting, and pasting text. In addition, context sensitive help is available. For example, in the screen shot below, a pop-up window displays when the editor recognizes you attempt to enter a property or method for the GPL vision class.

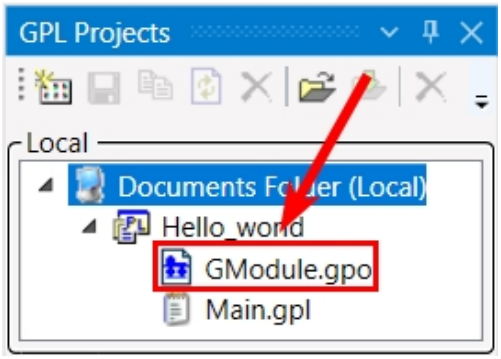


The editor automatically capitalizes keywords and built-in system classes, methods, and properties and color codes the text for greater readability.

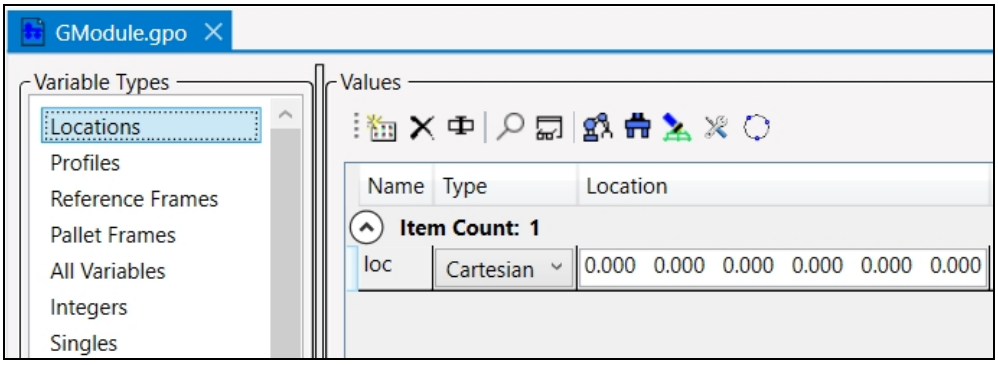
7. GPO

Editing a .gpo File

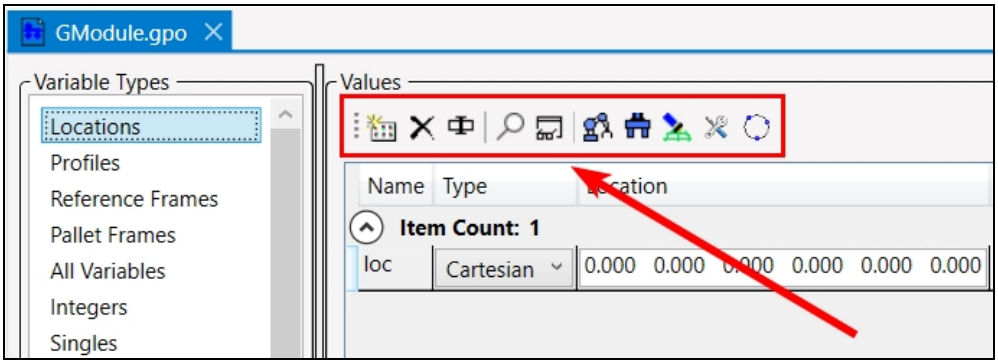
The .gpo file is a global modules file that contains motion objects and/or global GPL variables. It is an object file, a type of GPL code file. Double-click the .gpo file to open it.



The editor screen will display *Variable Types* in the left panel and their *Values* on the right.



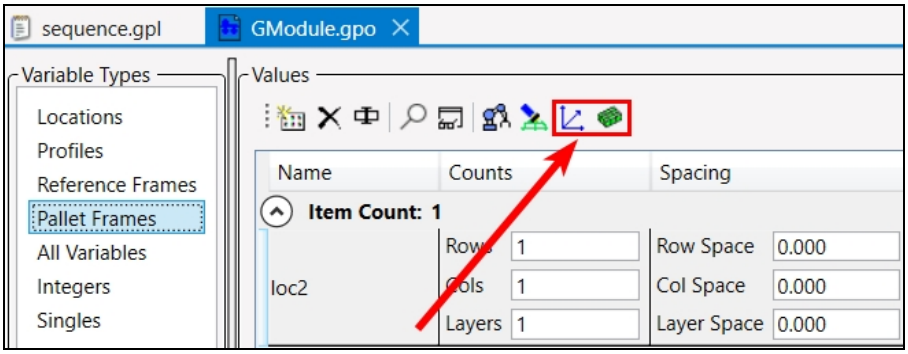
The toolbar at the top of the *Values* window will display icons that provide functions to interact with the GPL data. Some of the tool icons are dynamic, and they display only as needed depending on the variable type.



Here are the toolbar icon definitions:

Icon	Description
	Add
	Delete
	Insert new item into an array
	Append a new item into an array
	Delete the selected array item
	Find the reference to the variable in the Project
	Add selected item to the watch window
	Record the current position of the robot into the selected location
	Teach a vision pick offset and assign to the value of a location
	Use the vision system to locate an object and assign to the value of a location
	Apply the current location as a TOOL offset for the robot
	Run a process to define a tool offset via a tech sequence.

If you select *Pallet Frames*, the following icons will display.

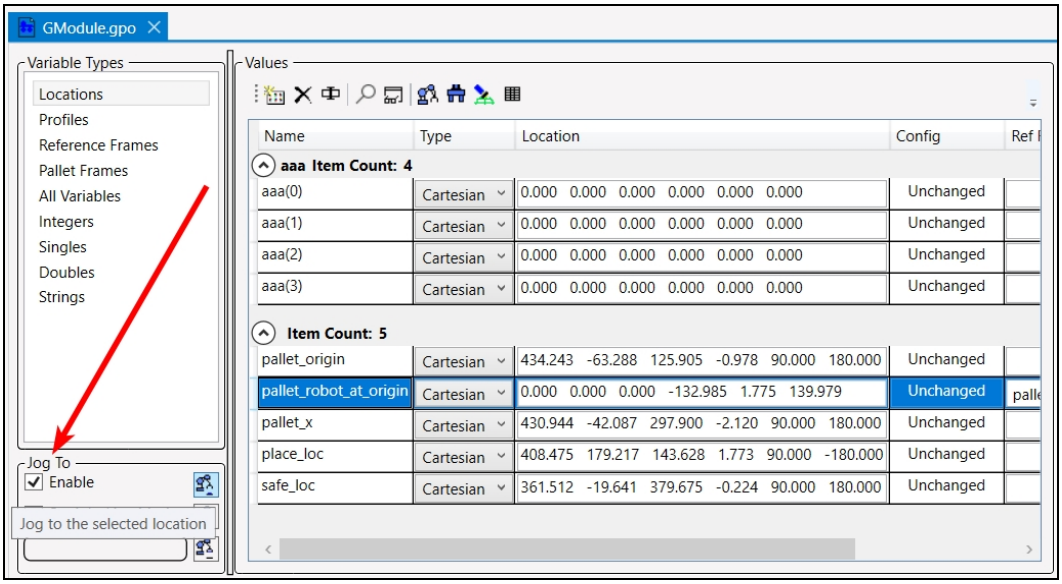


Icon	Description
	Teach a frame using 3 robot points
	Teach a pallet using the robot

Additional GPO Editor Toolbar Commands

- **Add**: Creates a single item or an array of global **Location, Profile** and **RefFrame** motion objects as well as single or arrays of **String**, **Integer**, **Single** and **Double** GPL global variables.
- **Delete**: Deletes a selected global variable or array.
- **Record**: Sets the position of the selected global **Location** equal to the current location of a robot. The number of the robot is specified by the drop-box to the right of the **Record** button.
- **Jog To** : Manually moves the robot to the selected location. See the image below for the jog being enabled.

When GDS is connected to a controller, the **Jog To** function assists in modifying the position and orientation of global **Locations** by enabling the robot to move back and forth between its current location and the value of a selected global **Location**. To activate this function, select a **Location** value, select a **Location** object, and check the **Enable** box in the *Jog To* panel. The following panel will then be displayed. The jog control is built into the GPO editor.



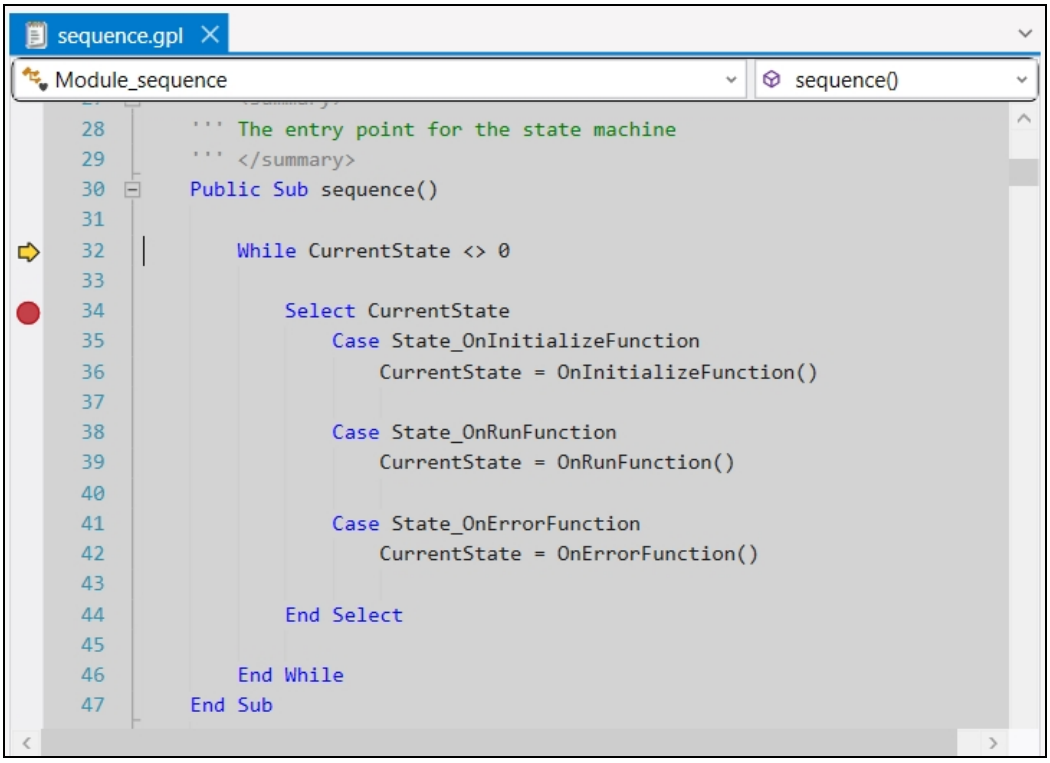
To move the robot between its current location and the position and orientation of the selected global **Location** object or its approach point, click and hold down the **Jog To**, **Jog Back** or **Jog Above** buttons. The robot will only move when the buttons are clicked.

In the GPO editor, these are the buttons:

Icon	Description
	Jog to location
	Jog above location
	Jog away

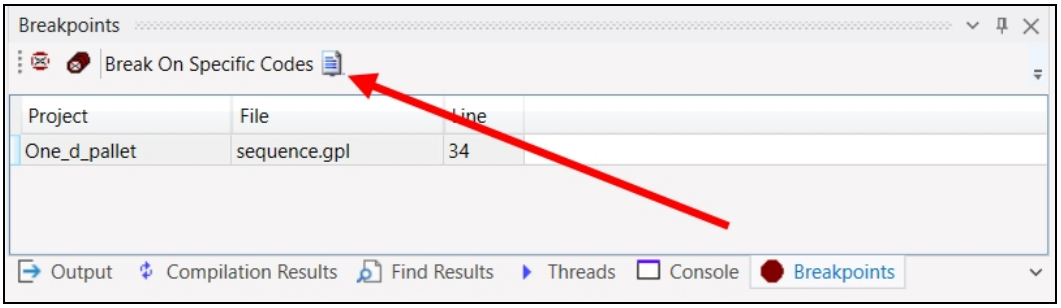
The Jog Speed for the operation is located at the bottom of the Virtual Pendant Window

While a Project is executing and is paused by a breakpoint or other means, the Project will enter the debugger operation. A GPL file being controlled by the debugger will appear in a tab in the same manner as files being edited.



As a visual cue, files being debugged are displayed with a gray background. Also, in the left margin, the next step to be executed is indicated by a yellow arrow. Any breakpoints that are set are indicated by a red dot as in line 34 above. Breakpoints can be set or cleared by clicking in the left margin.

The *Toggle Breakpoints* and *Clear All Breakpoints* functions are in the breakpoint window (displayed with the menu selection **Controller > Breakpoints**), shown below. Click on the **Break on Specific Codes** button, indicated by the arrow below, to change how GPL handles exceptions. By default GPL will not automatically break on an exception. But it can be useful to change this behavior to break on any exceptions or on exceptions with a specific set of error codes.



If you hover the cursor over a variable, the variable's name, type, and current value are displayed for the current context. Both the standard source code editor and the debugger can have their screens split. So, multiple editors and debuggers can be simultaneously active.

8. Vision

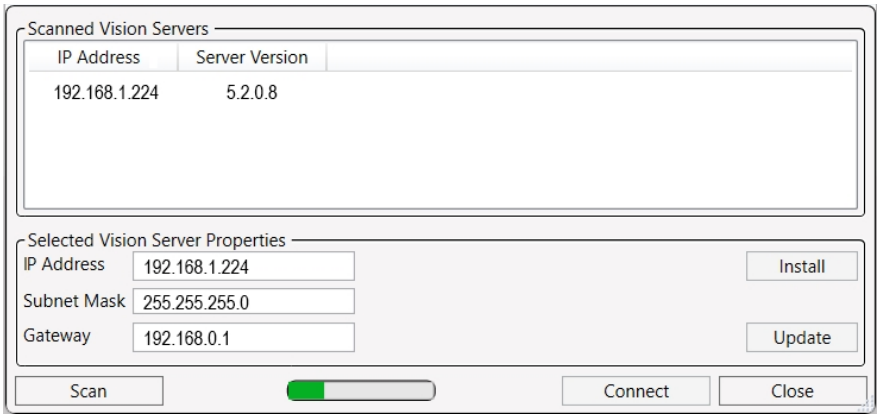
Connecting to the Vision Server

If your PreciseFlex robot is equipped with an IntelliGuide Vision Gripper, this section outlines how to connect to the internal vision server via GDS.

In the top center of the desktop, on the vision toolbar, click the purple **Connect/Disconnect** button to connect to the vision server if the correct IP address is displaying.

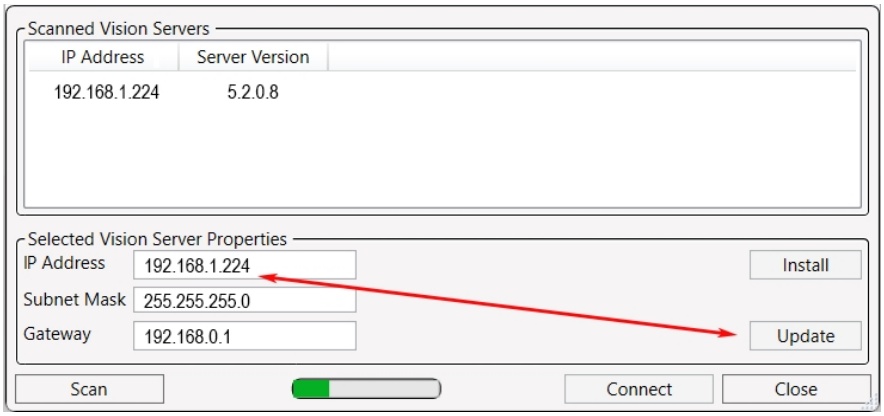
Alternatively, click on the blue **Scan for Vision Servers** button .

The *Scanned Vision Servers* window will display the available IP addresses of the vision servers. Select a vision server, and click **Connect**.

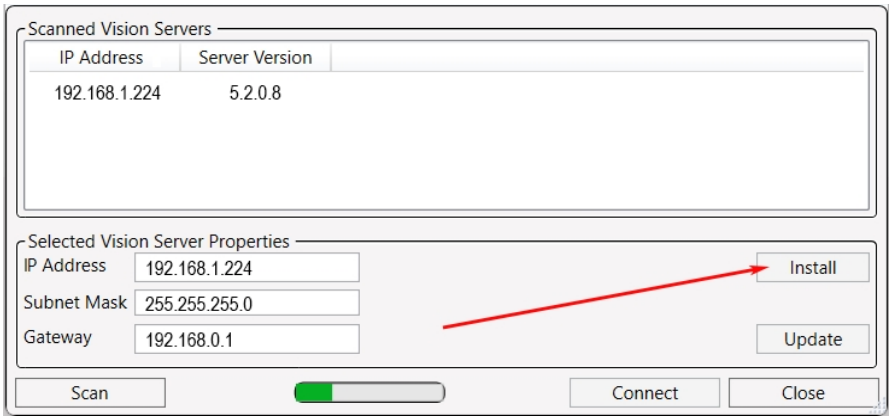


To change the IP Address, type in a new IP address, and click the **Update** button as shown below.

> ***NOTE:*** This step is used to change the IP address of the vision server. Do it if you need to change the address before connecting. Otherwise, select a different address from the list above to use as the connection address.



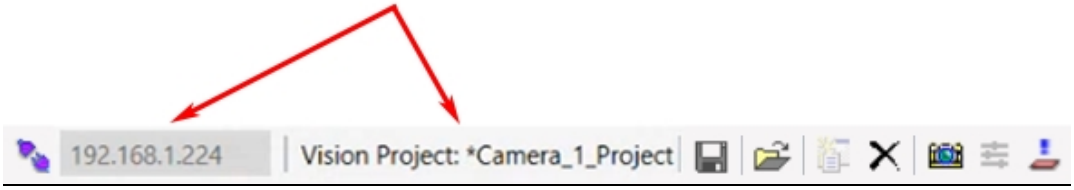
To install a new version of vision server software, click the **Install** button.



In the pop-up window that the **Install** button generates, locate the appropriate executable file for installation

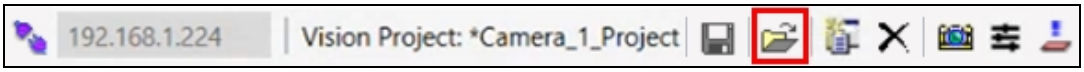
Vision Toolbar

The vision toolbar displays the vision server to which you are connected and the project that is loaded into GDS.

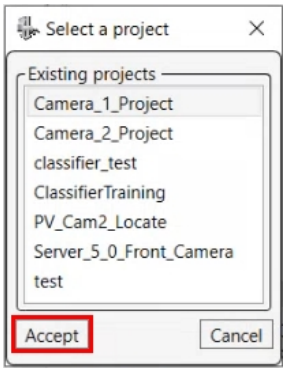


Click **Save** to save the vision project. Only the vision project will get saved, not the GPL project.

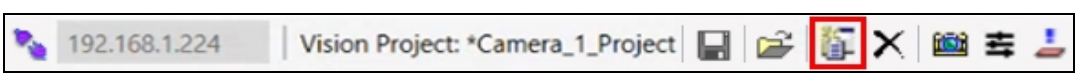
Click **Load a New Project** to select another project.



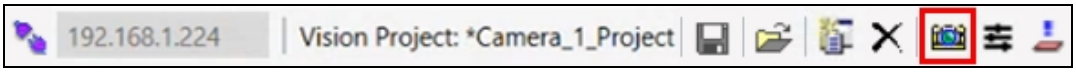
The *Select a Project* window will display. Select a project from the list, and click **Accept**.



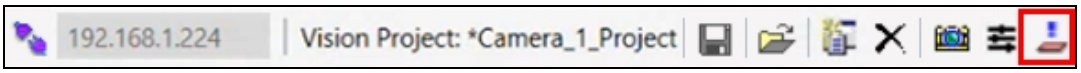
Clear **Clear the Current Project** to clear it.



Take a Picture takes a picture using the camera that is currently selected. (see also the Camera Display Window window).



Live Video Mode executes a continuous steam of images from the currently selected camera.

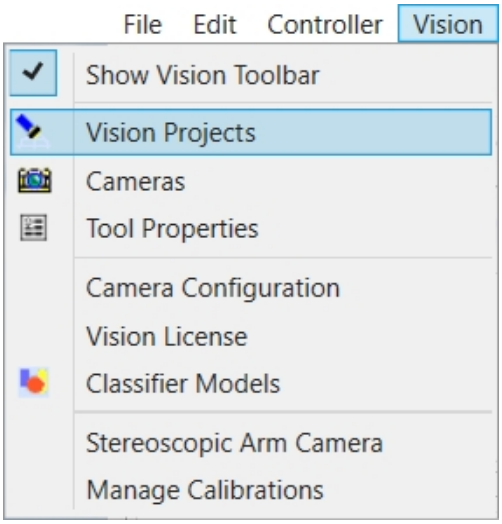


Vision Drop-Down Menu

In the main menu bar, the *Vision* drop-down menu links to the following windows and resources:

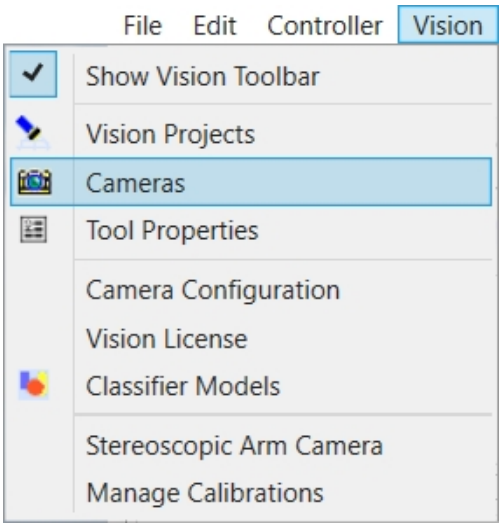
Vision Projects

Vision Projects opens the Vision Projects window, which contains the vision tools.



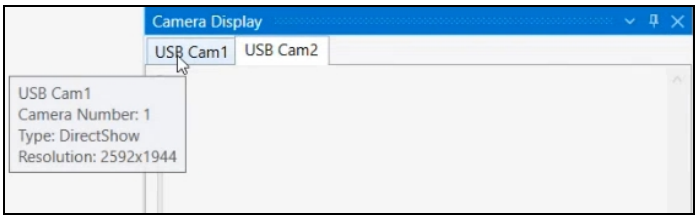
Cameras

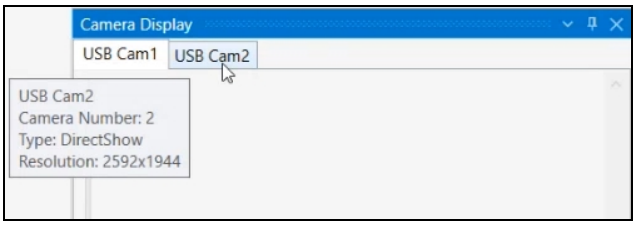
Cameras opens the *Camera Display* window, which contains camera resources.



Use the resources in the **Camera Display** window to interact with the forward-facing camera (USB Cam 1) and the downward-facing camera (USB Cam 2).

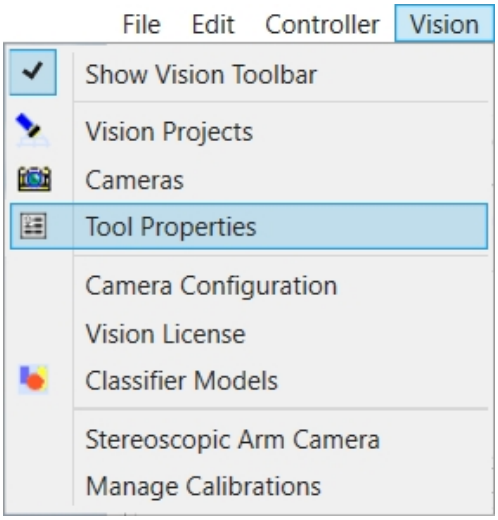
Downward-facing camera (USB Cam 2) looking at worktable surface





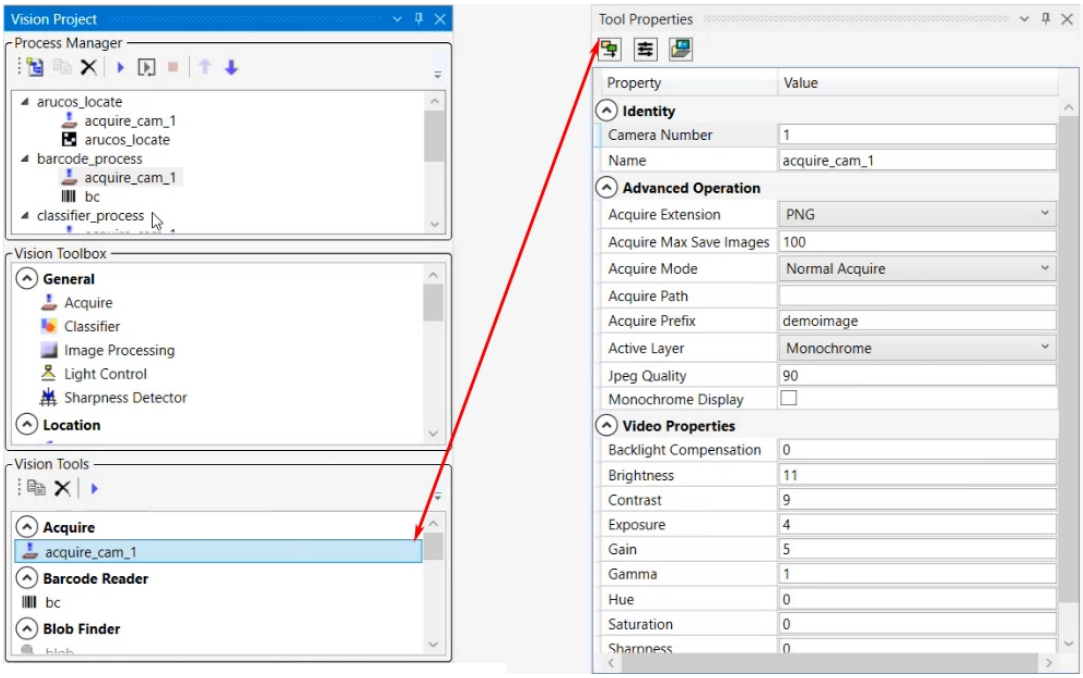
Tool Properties

Tool Properties will display whatever vision project and vision tool is selected at the moment.



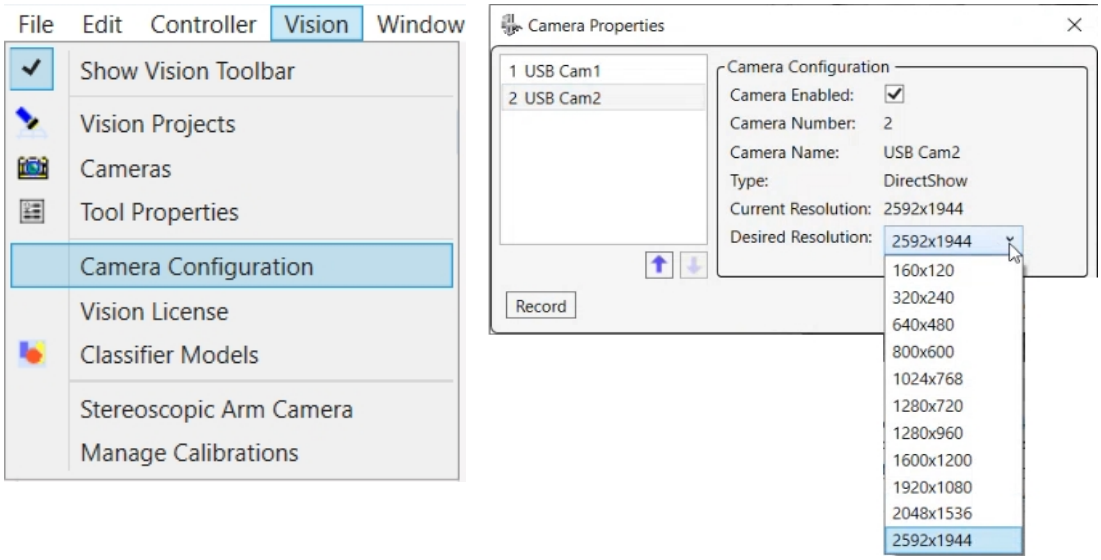
In the example below, the "acquire_cam_1" *Vision Tool* is selected, and its properties display in a window on the right side of GDS.

Click the camera icon  in the upper left of the *Tool Properties* window to switch from Camera 1 (forward facing) to Camera 2 (downward facing).



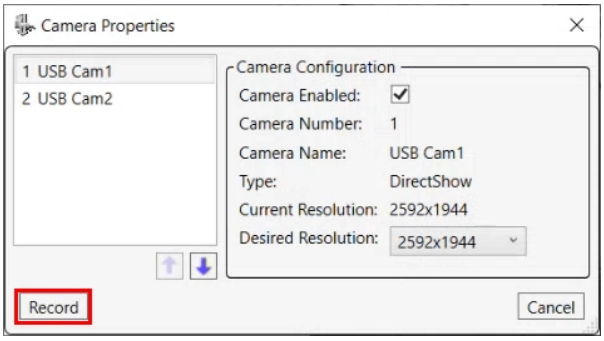
Camera Configuration

Camera Configuration opens the *Camera Properties* window, from which you can modify the cameras' pixel resolution.



- The higher the resolution, the higher the image quality, but the slower the pictures get taken.
- The lower the resolution, the lower the image quality, but the faster the pictures get taken. It's best to not use a resolution lower than 640 x 480, though it depends on the application.

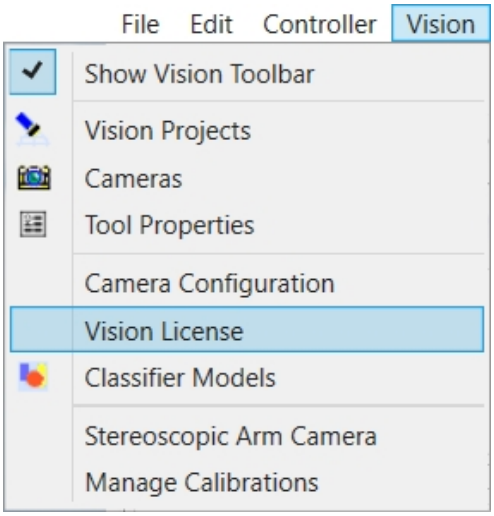
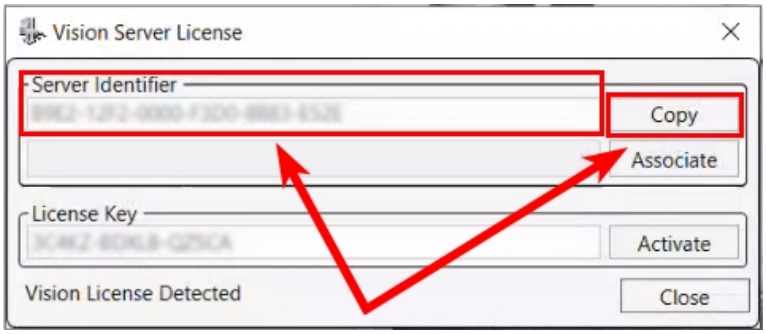
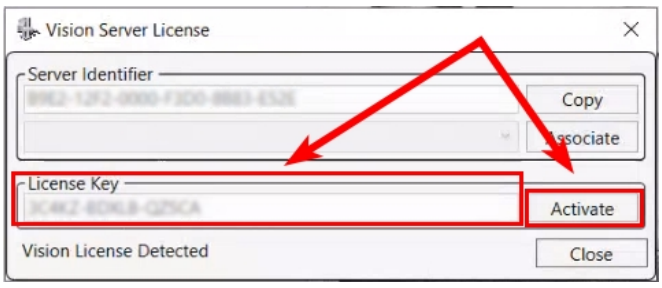
After changing the camera resolution, click the **Record** button.



Installing a Vision License

The vision license is pre-installed on every IntelliGuide Vision Gripper. You only need to install it if the vision processing is being run on an external device. To get a vision license, send your PC information to sales_preciseflex@brooksautomation.com.

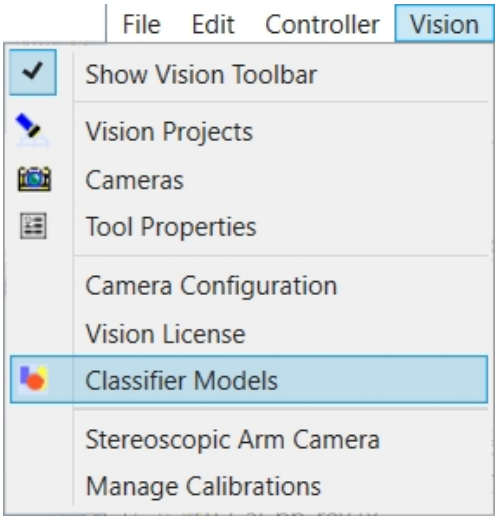
Follow the procedure below to acquire and activate a vision license.

Step	Action
1.	In the main title bar, select the **Vision** drop-down menu, then select **Vision License** to display the vision server license. 
2.	Select and copy the *Server Identifier* number. 
3.	Email the *Server Identifier* number to sales_preciseflex@brooks.com to start the order process for a vision server license key.
4.	When you receive the vision server license key, enter it into the *License Key* field and click **Activate**. 

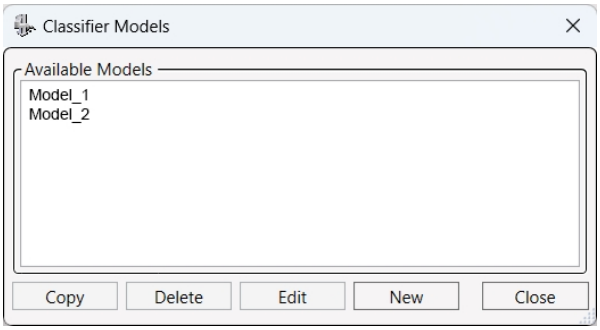
NOTE: You must run the application in administrator mode in order for the license to be successfully applied.

Classifier Models

Classifier Models contain *positive conditions* and *negative conditions* attributed to a region of the camera's field of view. They are used for classification – positive or negative – and not for object location.
- *Positive conditions* represent the presence of the specific characteristic or condition the classifier model is trying to identify.
- *Negative conditions* represent the absence of the specific characteristic or condition the classifier model is trying to identify.

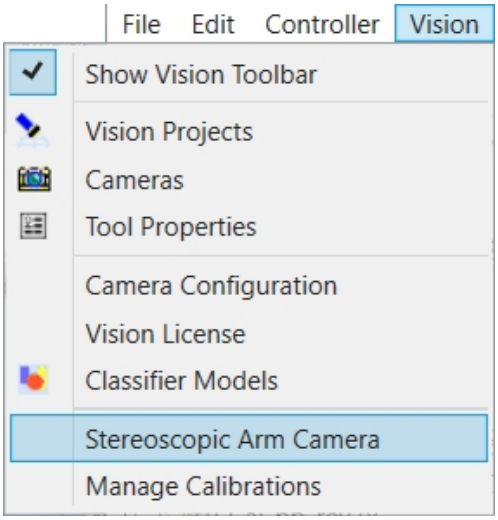


Classifier Models must be trained, and that training information is saved in a *Model* file. The menu selection **Classifier Models** displays the available Classifier Models you've trained. See Selecting and Training Classifier Models.



Stereoscopic Arm Camera

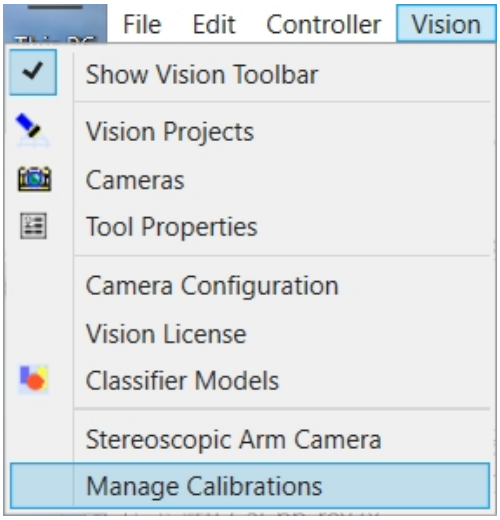
Stereoscopic camera calibration determines the relationship between the robot coordinate system and the camera coordinate system. See the *IntelliGuide Vision* user manual instructions on how to perform the **Stereoscopic Camera Calibration**.



Manage Calibrations

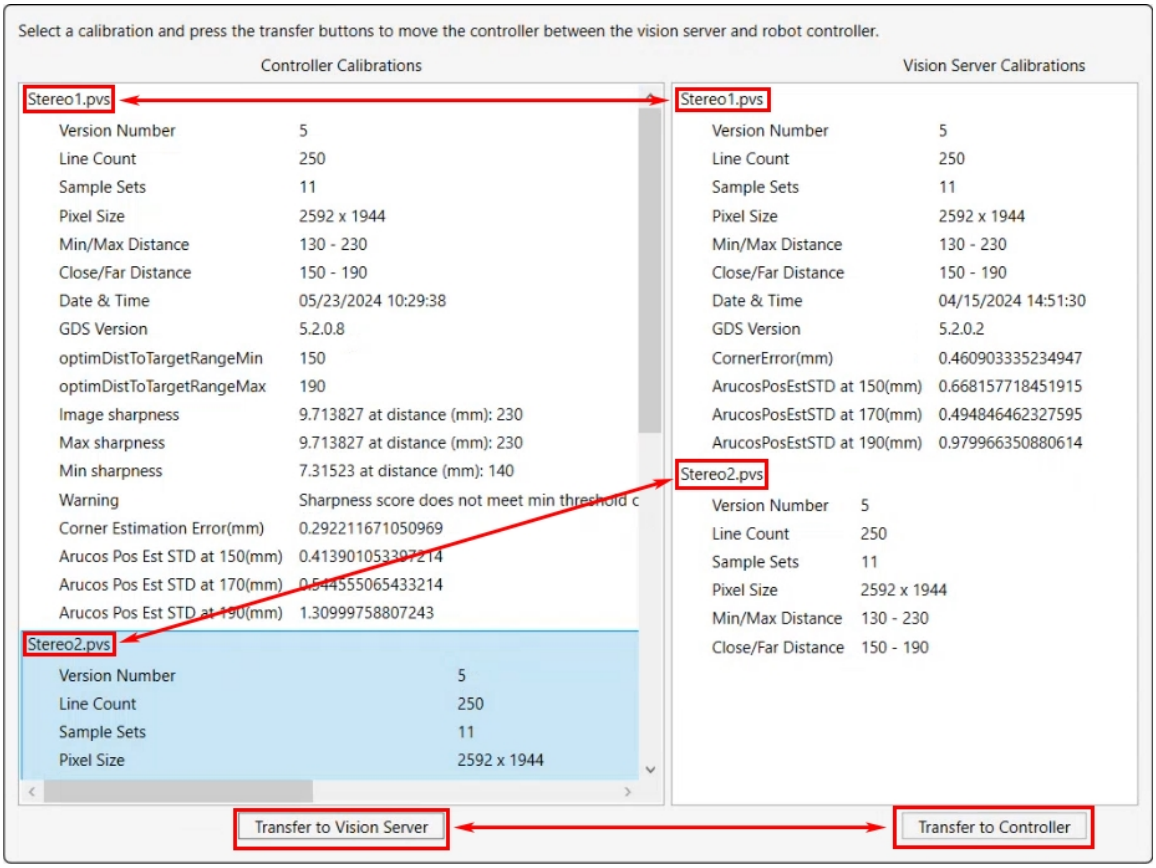
Stereoscopic Calibration is when a robot takes multiple photos of a target and creates a Calibration Model. The photo calibration model data is stored on the robot's controller.

The **Manage Calibrations** tool helps transfer a copy of that data from the robot's controller to the IntelliGuide gripper. If you detach that IntelliGuide gripper and attach it to another PreciseFlex robot, you can transfer the stored calibration files from the IntelliGuide gripper to the new robot's controller.



In the image below, the *Manage Calibrations* window shows that two calibration files reside on the robot controller ("Stereo1.pvs" and "Stereo2.pvs"), and two calibrations reside on the vision server (also "Stereo1.pvs" and "Stereo2.pvs"). You can select and transfer data from one to the other by selecting the data files and clicking the **Transfer to Vision Server** button or the **Transfer to Controller** button.

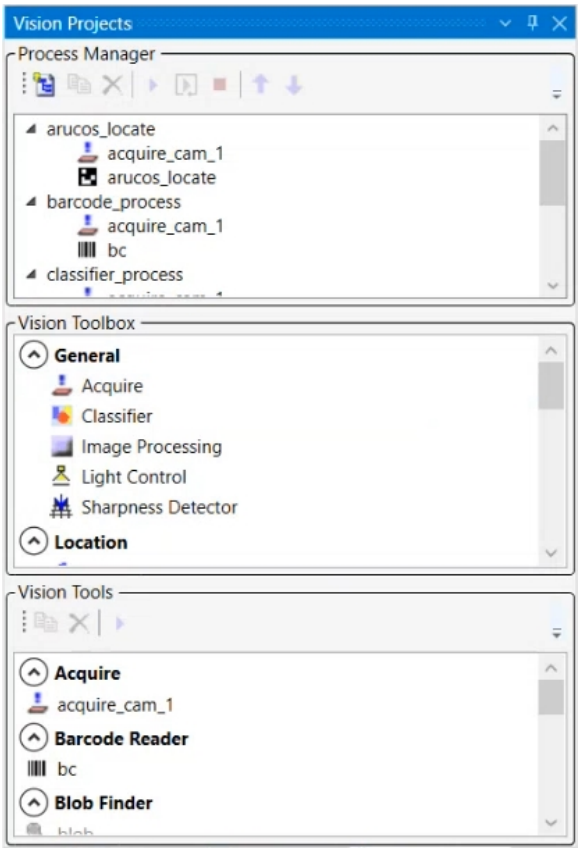
> ***NOTE:*** When you click in a ".pvs" file in a widow, it and its contents highlight in blue, as shown below in the *Controller Calibrations* window, the file titled "Stereo2.pvs."



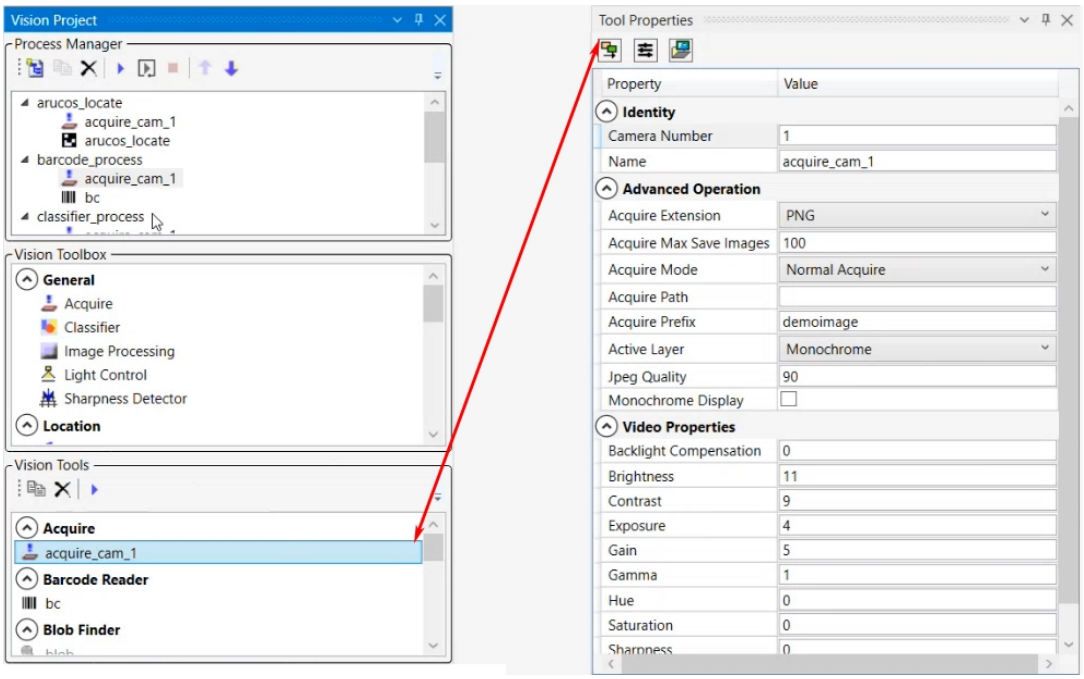
Vision Projects

Vision Projects

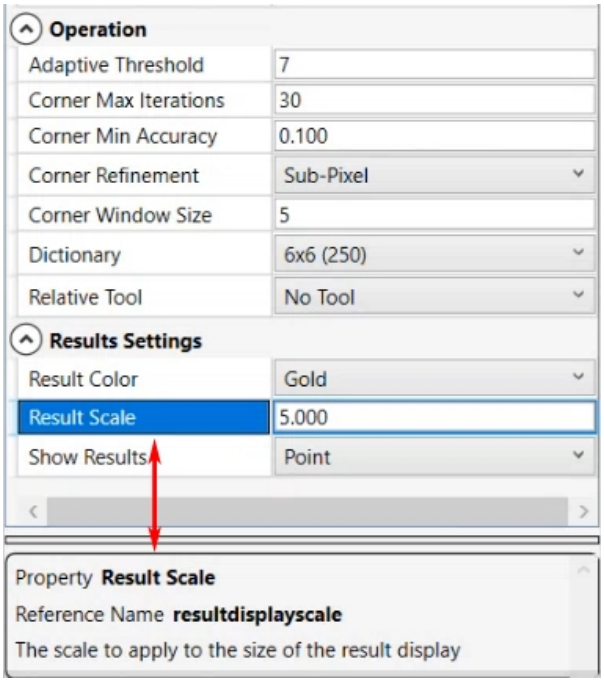
When GDS connects to the Vision server, the *Vision Projects* windows displays. It contains windows for *Process Manager*, *Vision Toolbox*, and *Vision Tools*.



When you select a tool from the *Vision Tools* or *Vision Toolbox* window, its properties display in a *Tool Properties* window. You can change properties there.

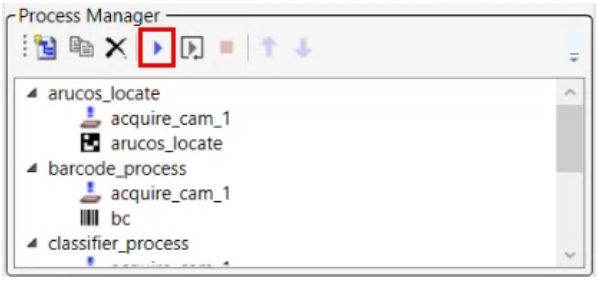


Detailed information about each selected property displays, at the bottom of the *Tool Properties* window.



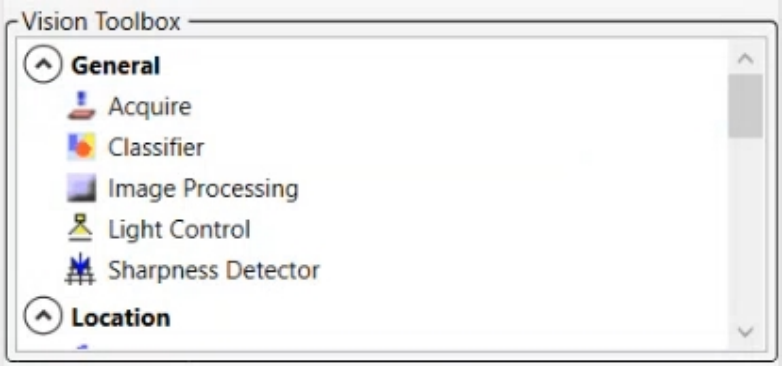
Process Manager

In GDS, a "process" is a collections of tools dragged from the *Vision Toolbox* and *Vision Tools* windows to the *Process Manager* window to be run sequentially. Initially, the vision system tools – such as camera acquisition and lighting adjustments – are configured individually. These tools are then integrated into processes in the *Process Manager*. In the example below, the process *arucos_locate* runs and executes the vision system tools "acquire_cam_1" and "arucos_locate" sequentially. The **Run the Process** arrow is highlighted in red.

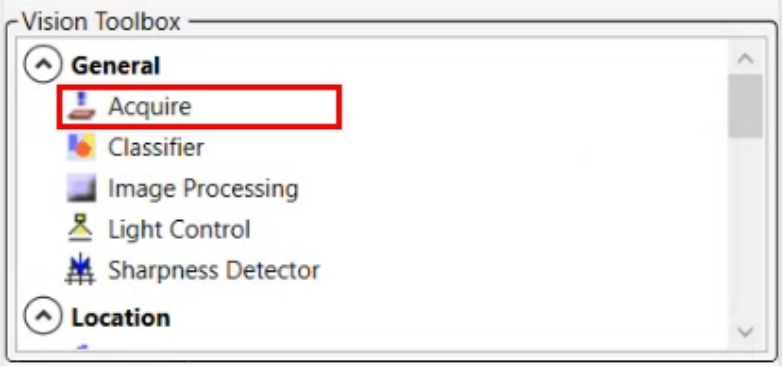


Vision Toolbox

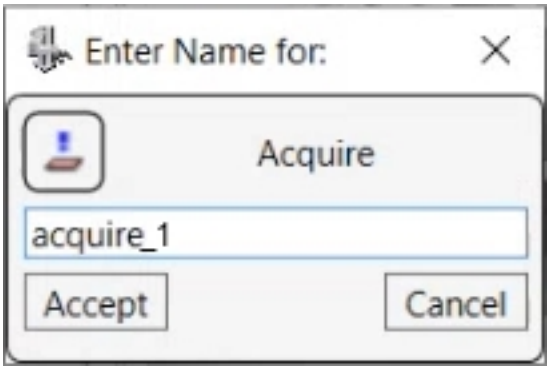
The *Vision Toolbox* contains all the tools available on the vision server.



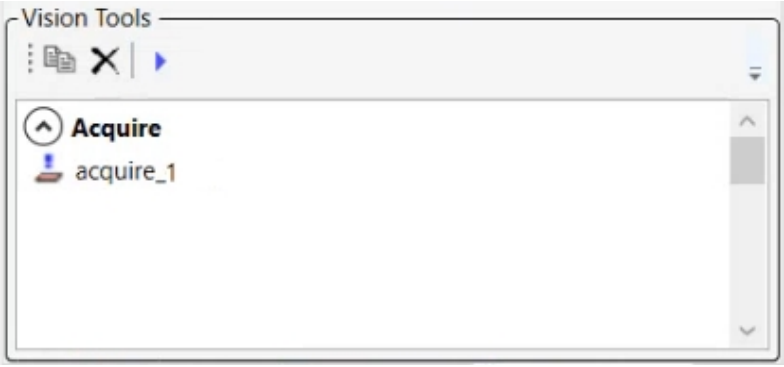
Acquire gets an image for the project. Double-click **Acquire**.



In the pop-up window, enter a name for the Acquisition Tool, and click **Accept**.



The Acquisition Tool will display in the *Vision Tools* window.



When you create a vision tool, it is associated with the camera that is selected. In the image below, *USB Cam2* – Camera 2 – was selected, so the tool **acquire_1** is associated with Camera 2. When you click on a vision tool, the *Camera Display* window will display what camera is associated with the vision tool

Click **Run** to run the tool.

To create new instances of a Vision Tool, drag the desired tool type from the *Vision Toolbox* to the *Vision Tools* window.

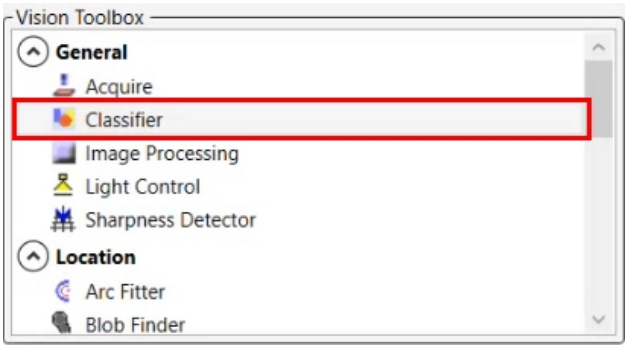
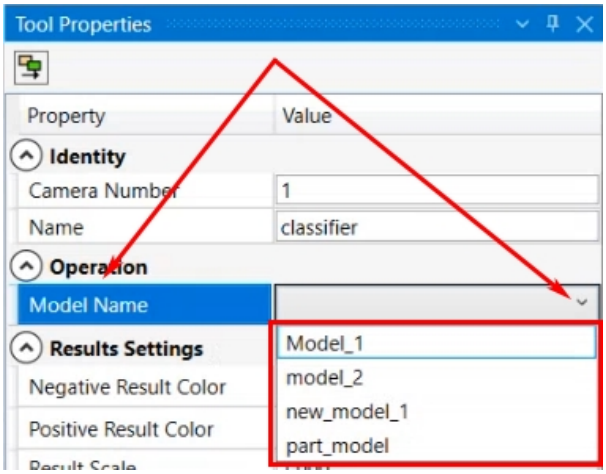
Selecting and Training Classifier Models

Selecting from the Vision Toolbox

The *Vision Toolbox* contains the Classifier tool. If you add a Classifier tool to a project, you also need to select a Classifier Model, which the Classifier tool uses for reference.

> ***NOTE:*** You need to train each Classifier Model. See Training Classifier Models for training instructions.

To select a Classifier Model for a Classifier tool, you can either select **Classifier Model** from the Vision drop-down menu or follow this procedure:

Step	Action
1.	In the *Vision Toolbox* window, click on **Classifier**. 
2.	The **Classifier** *Tool Properties* window will display on the right side of the screen. Open the **Operation** > **Model Name** drop-down window and select a model. 

Training Classifier Models

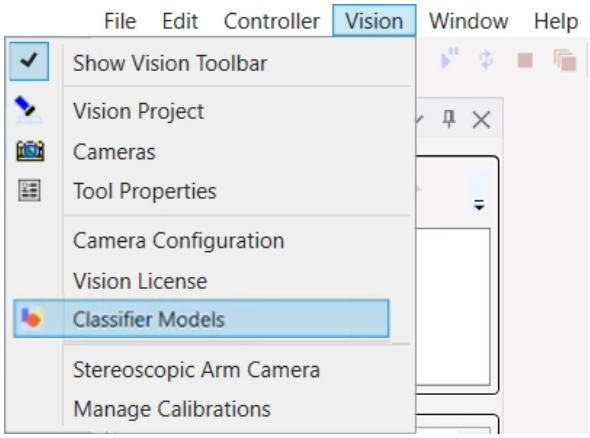
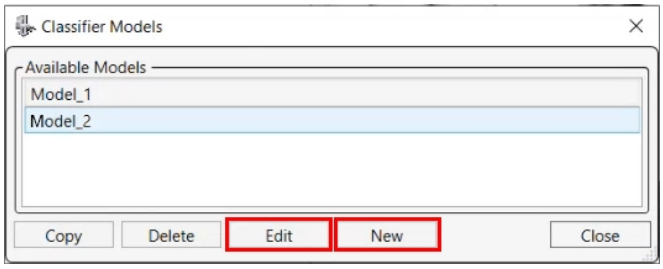
Each Classifier Model must be trained with a set of instructions about what the vision system must focus on and what the vision system must avoid.

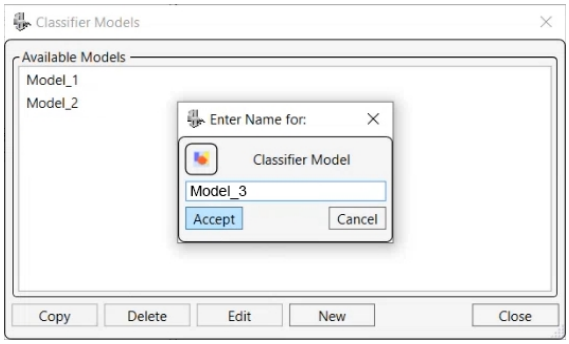
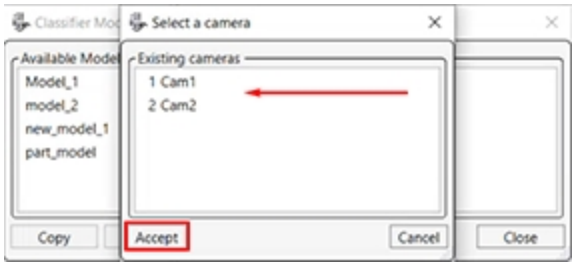
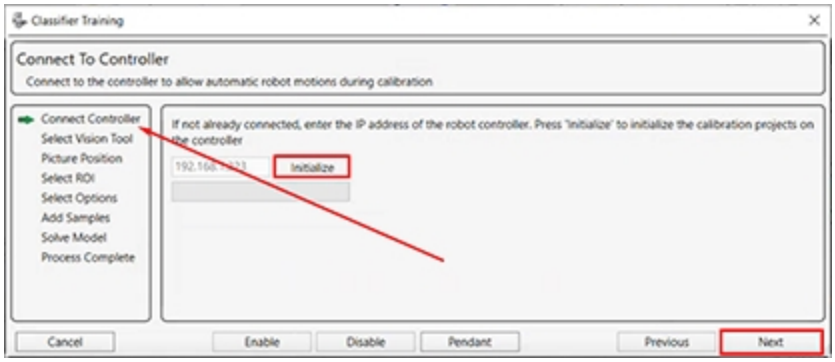
> ***NOTE:*** If there are no existing Classifier Models, you must train an initial Classifier Model.

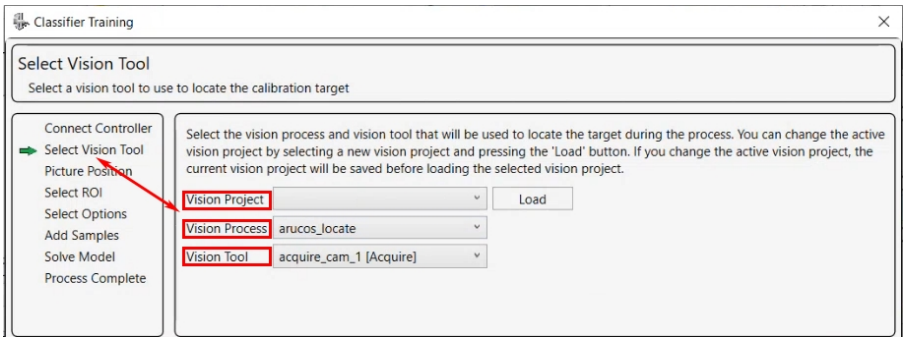
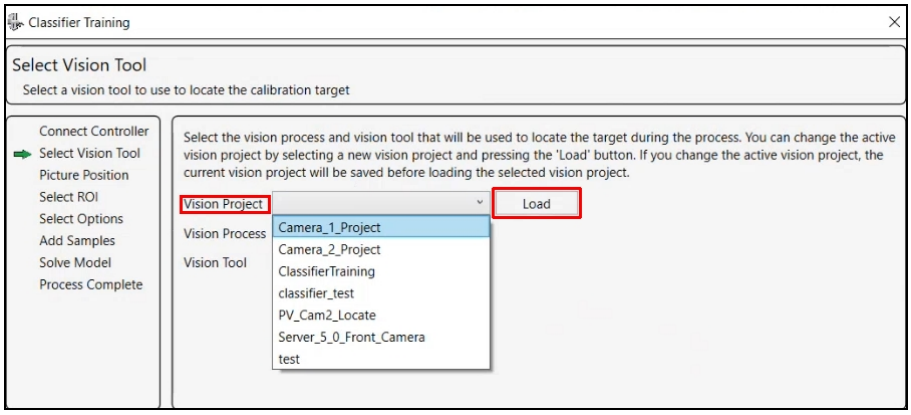
You can train a new Classifier Model or copy and modify an existing model. To train a Classifier Model, follow the procedure below.

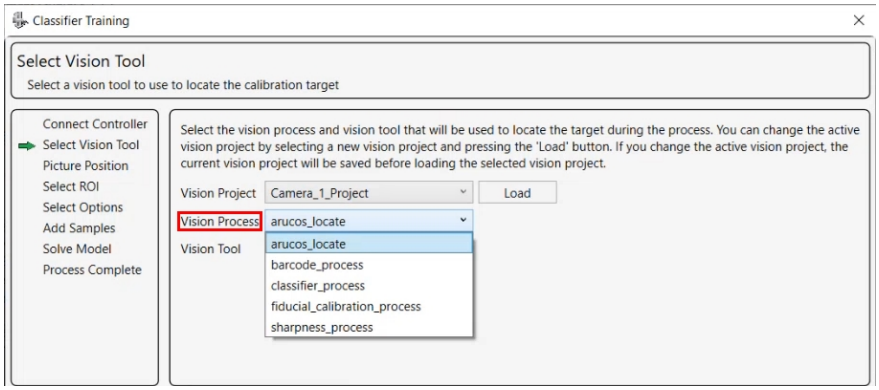
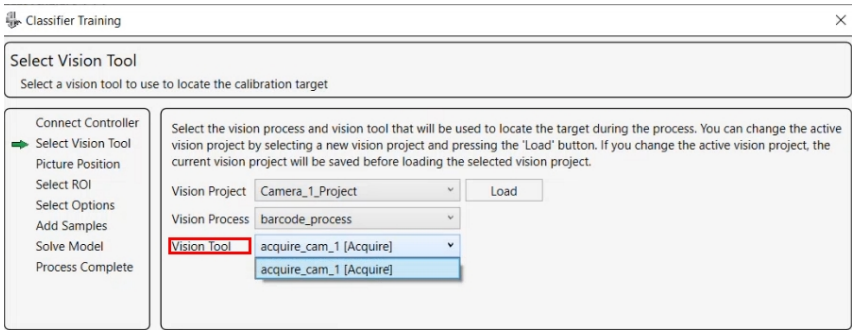
Before training a Classifier Model, select the Vision Process and Acquisition Tool that gives you a good image:

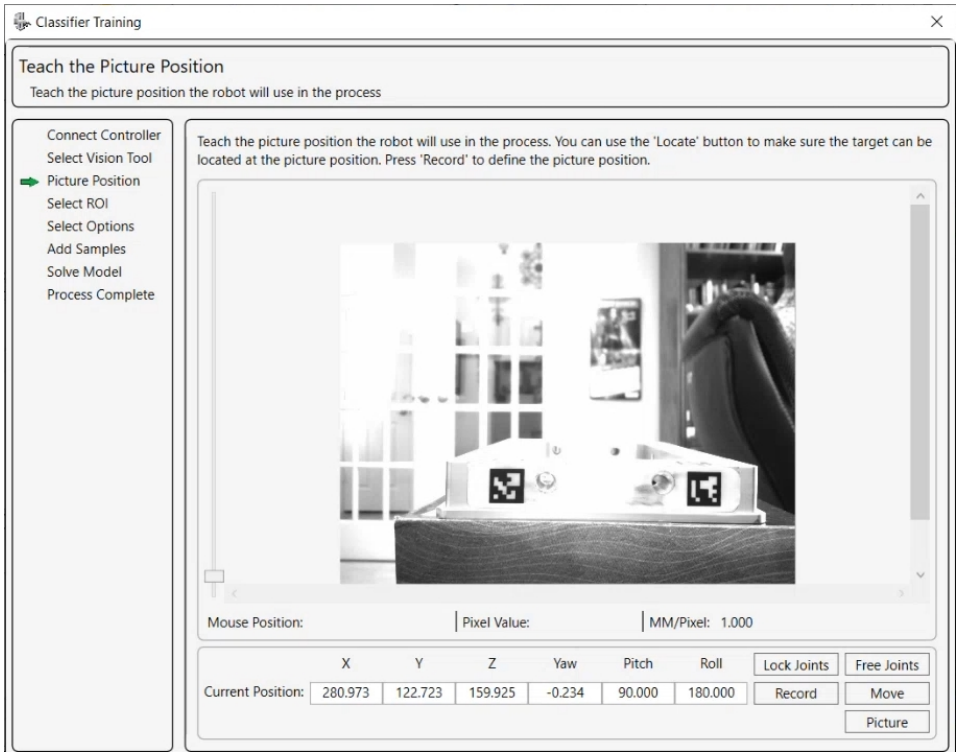
- To create a Vision Project (see the Vision Projects section)
- Make sure it contains a Vision Process (see the Process Manager) that at least has an Acquisition Tool (see the *Vision Toolbox*) for the camera you're using.

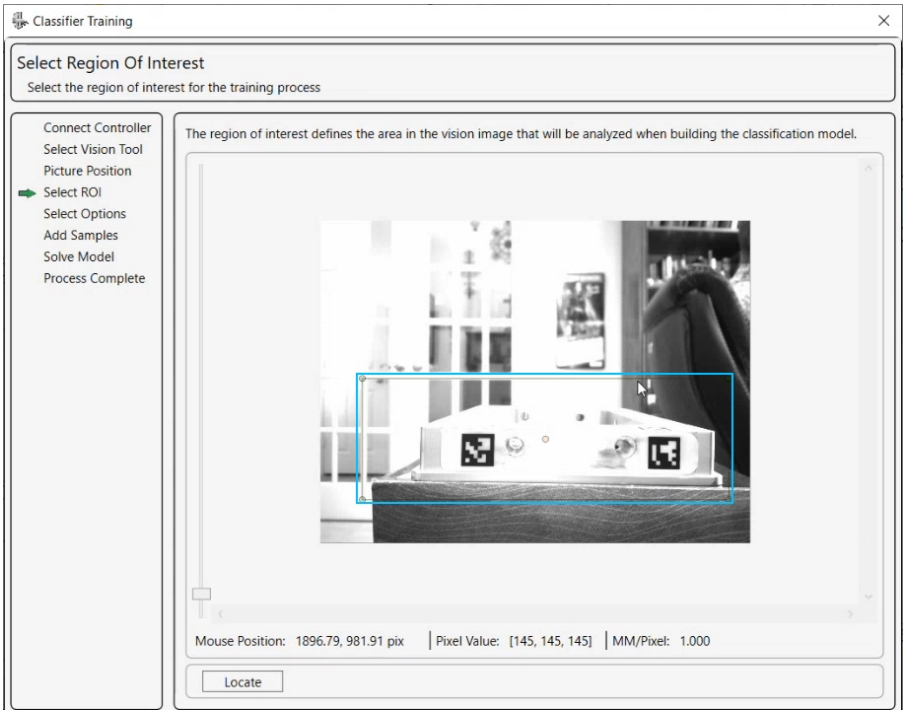
Step	Action
1.	Open the **Vision** drop-down menu and select **Classifier Models**. 
2.	When the *Classifier Models* window displays, select a model and click **Edit** to edit an existing Classifier Model, or select **New** to train a new *Classifier Model*. ***NOTE:*** *If you select* ***Edit****, the Classifier Training wizard, shown in Step 5 below, will open.* 

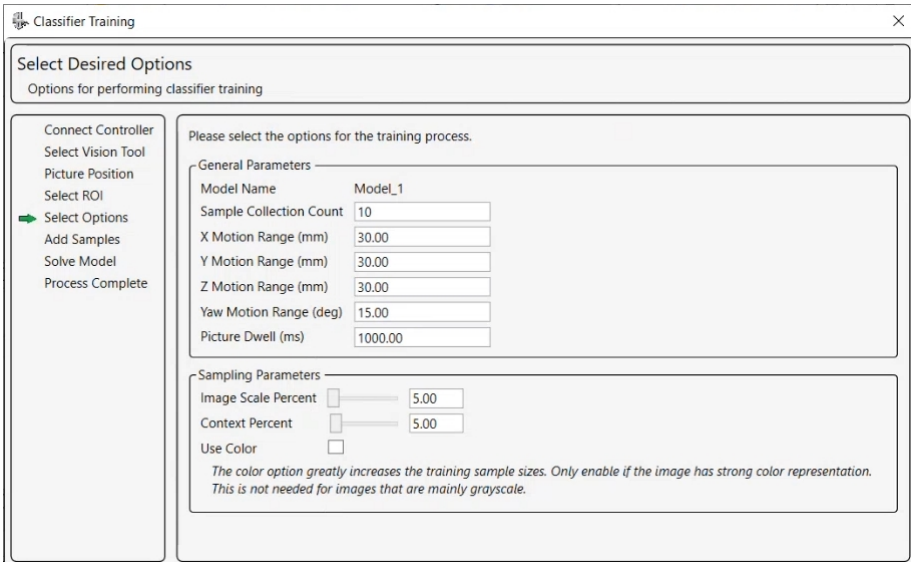
Step	Action
3.	If you select, **New**, an *Enter Name for:* window will display. Enter a name for the new Classifier Model, and click **Accept**. 
4.	In the subsequent *Select a Camera* window, select a camera that you want to apply the Classifier Model to, and click **Accept**. 
5.	The *Classifier Training* wizard window will open. Select the first option in the left panel, **Connect Controller**. ***NOTE:*** *If the controller is not already connected, enter the controller IP address in the main window and click* ***Initialize****.* When the controller is connected, click the **Next** button in the lower-right corner. 

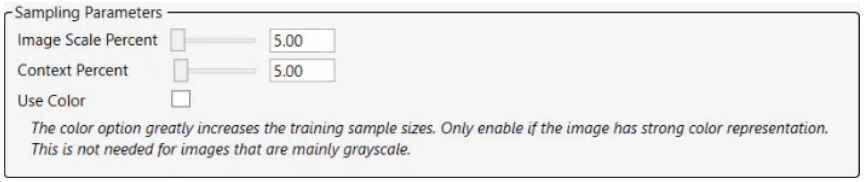
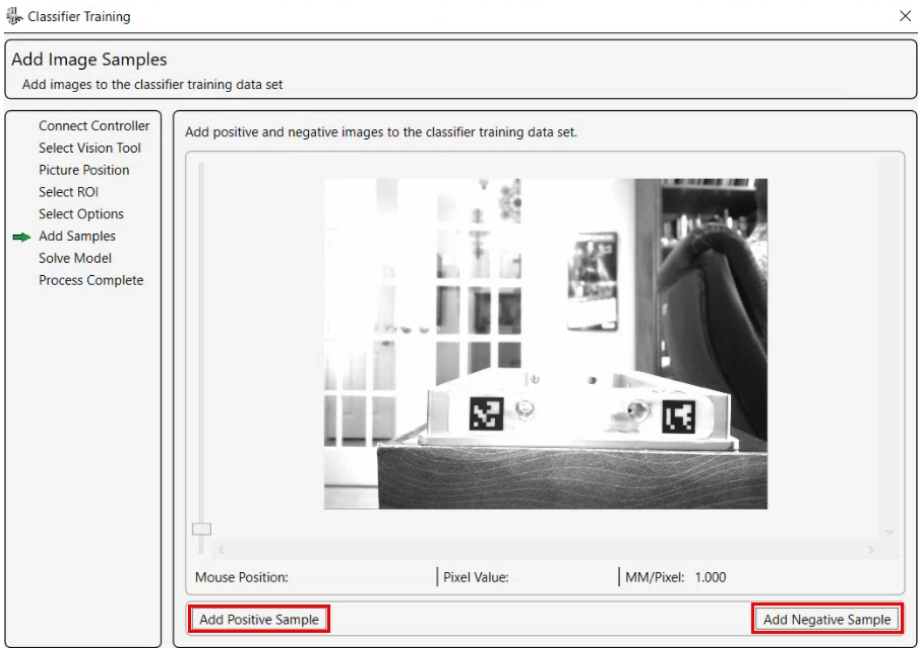
Step	Action
6.	Click on the **Select Vision Tool**, which displays drop-down menus for • *Vision Project* • *Vision Process* • *Vision Tool* 
7.	Open the **Vision Project** drop-down menu and select a project. If there is no project available, create a Vision Project in the Vision Projects window. After selecting a Vision Project, click the **Load** button. 

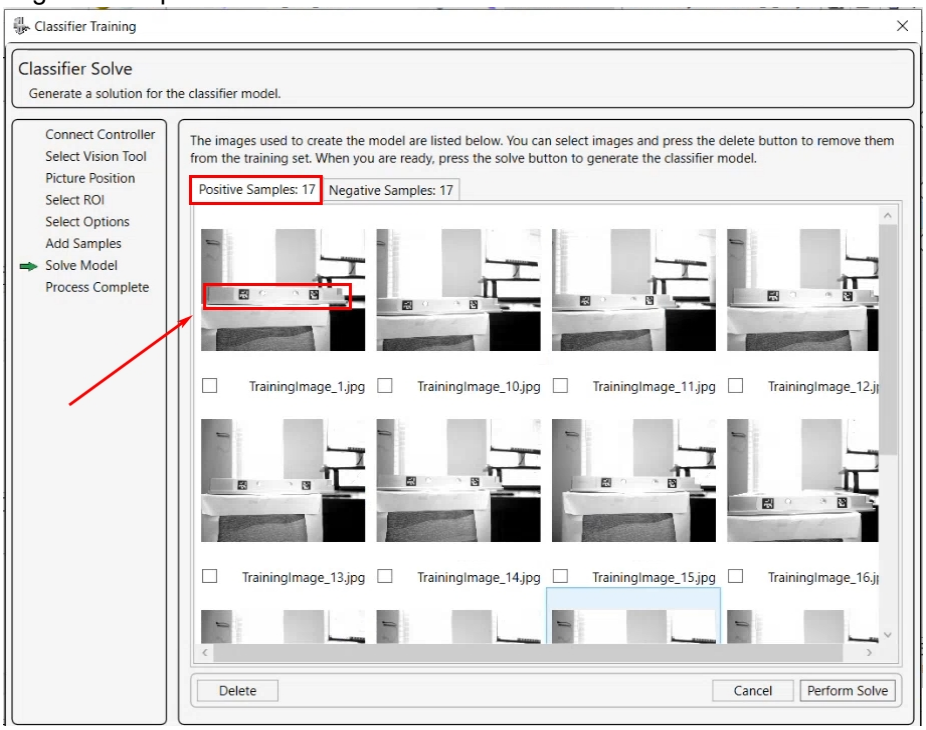
Step	Action
8.	Open the **Vision Process** drop-down menu, and select a Vision Process. If there is no Vision Process, create one. For more information, see the section on the Process Manager. 
9.	Open the **Vision Tool** drop-down menu and select a tool from the To create new instances of a Vision Tool, drag the desired tool type from the Vision Toolbox to the Vision Tools window. window. After selecting a Vision Tool, click **Next**. 

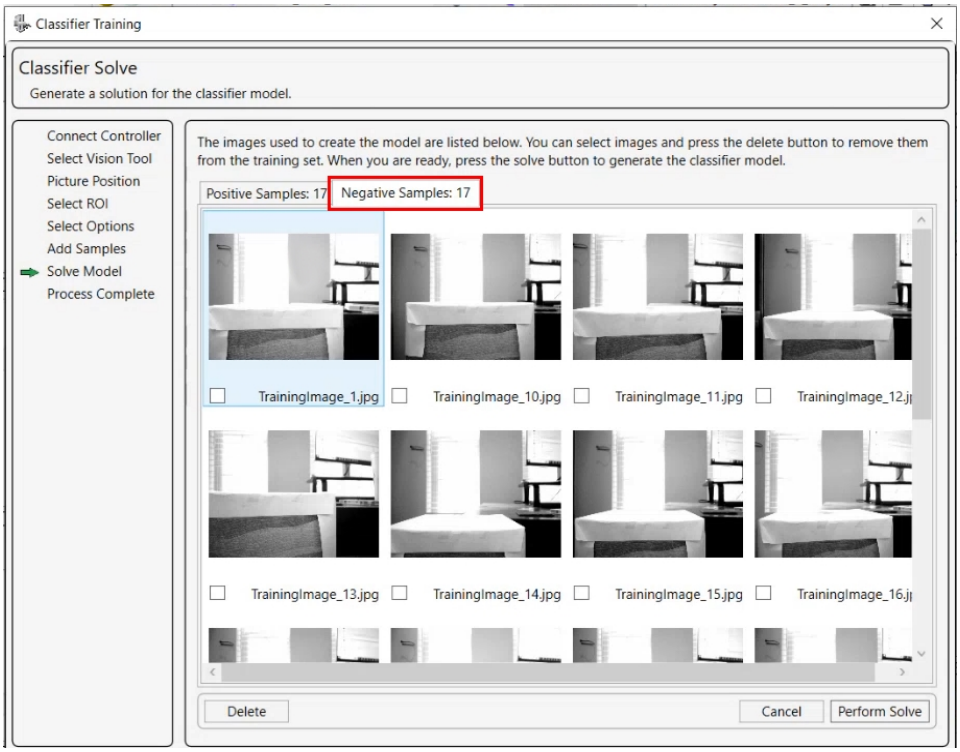
Step	Action
10.	In the left column, select **Picture Position**. Define a picture position that represents where you want the robot to start from when it's training the Classifier Model. In the wizard, the robot will move around to take pictures at different positions in the scene when training the model. This represents the starting location. Click **Record** when the robot is in the desired position, then click **Next**. 

Step	Action
11.	In the left column, select **Select ROI** (Region Of Interest), which is a specific area of the photo that you want to analyze. The Region of Interest defines the area in the picture that will be used when building the Classifier Model. Click and drag the mouse around the Region of Interest. Click **Next**. 

Step	Action
12.	The *Select Options* section contains parameters you can set to add variables to the camera location. When Vision analyzes sample sets, the more variability that you add, the better Vision will be able to determine if the image it sees is "good" or "bad" or "positive" or "negative." This section is usually pre-filled with default settings. 
13.	In the **Select Options** *General Parameters* window, *Sample Collection Count* indicates that the robot will move to 10 different locations. From the starting point, the robot will move: • ± 30 mm in the *X Motion Range* • ± 30 mm in the *Y Motion Range* • ± 30 mm in the *Z Motion Range* • ± 30 degrees in the *Yaw Motion Range*. • and it will dwell (*Picture Dwell*) for 1000 ms before taking a picture sample. 

Step	Action
14.	The **Select Options** *Sampling Parameters* window contains controls for scaling the image for analysis. Analyzing an image at 100% of its size will take a lot of time on a large image. The lower the percentage number is, the faster the analysis of the model will execute; the higher the number, the slower the analysis. Whatever the default number is – usually 5% to 10% – is a reasonable number. *Use Color* will increase training data size. 
15.	In the *Add Samples* section, you collect images to help train the vision system. Select *Add Positive Sample* or *Add Negative Sample*, and those images will be added in the next section, *Solve Models*. 

Step	Action
16.	In step 16, you can review the Positive samples. The images below are examples of Positive Samples for the Vision system to focus on. The samples were taken according to the *General Parameters* as shown in Step 13: pictures were taken at 10 locations from the X, Y, and Z range of ± 30 millimeters, and yaw of ± 30 degrees. Below, and in Step 17, are examples of all images associated with the positive and negative samples. 

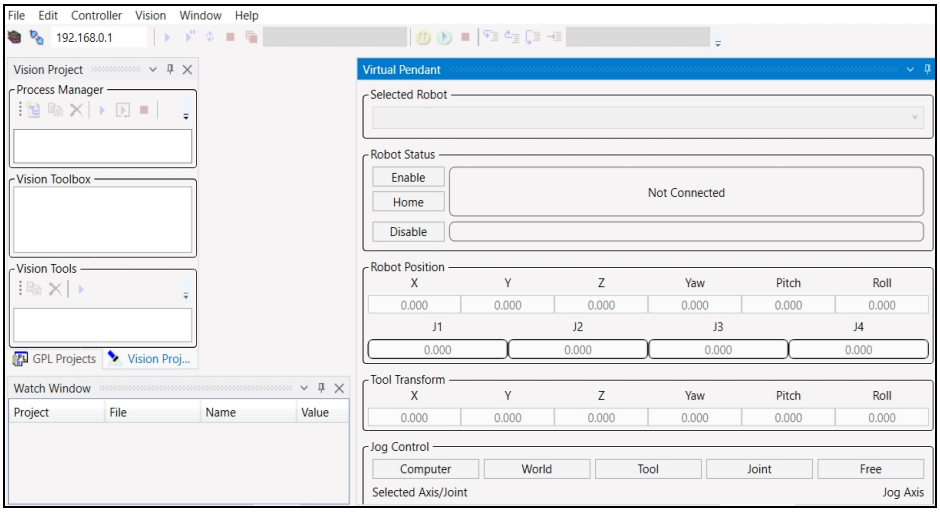
Step	Action
17.	Step 17 shows Negative Samples for the robot to avoid. When the robot is finished, you can select and delete photos. When you are satisfied with both the Positive Samples and the Negative Samples, click the **Perform Solve**button. 
18.	In the left column, select **Process Complete** when finished.

Appendices

Appendix A: GDS Programming Example - Hello World

In this exercise, you will learn how to create a Project, write a simple procedure that outputs the text "Hello World," loads the Project into the controller, and executes it.

GDS should be connected to your controller and look approximately like the image below. Since you will not be moving the robot, the power to the robot need not be enabled.

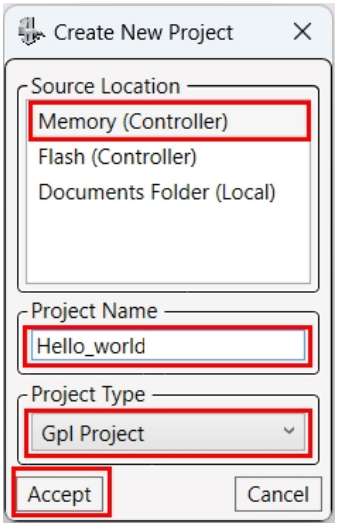


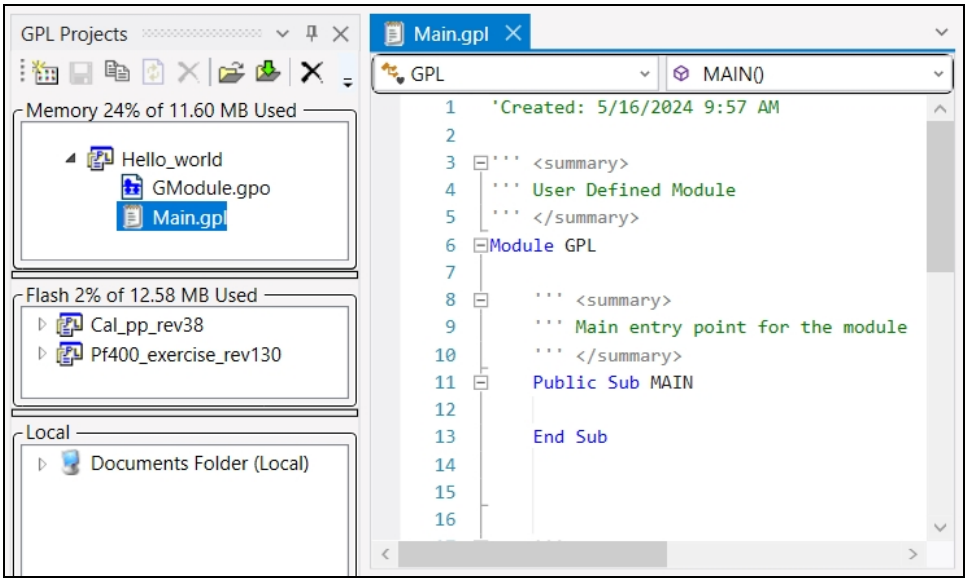
Creating a Project

The first step is to create a new Project. Follow the procedure below.

> ***NOTE:*** Rather than create a Project in the documents folder, create it in Memory, which requires you to be connected.

Step	Action
1.	Click on **File > New Project**.

Step	Action
2.	In the *Create New Project* pop-up window: • Enter "Hello_world" in the *Project Name* field> • Select **GPL project** from the *Project Type* drop-down menu. • Select **Memory**. • Click **Accept**. ***NOTE:*** *If you are not connected, the Create New Project window will only display the Documents Folder.* 
3.	When creating the **Hello_world** Project and any new Project, GDS automatically adds into *GPL Projects* a global modules file ("GModule.gpo") and a main source-code file ("Main.gpl").  Also, the "Main.gpl" file already contains the definition for the "MAIN" public procedure. By default, this is the procedure that will be started when executing the Project, although this can be easily changed by accessing the Project properties.

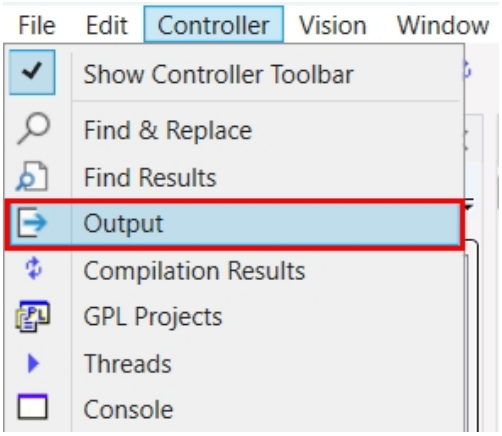
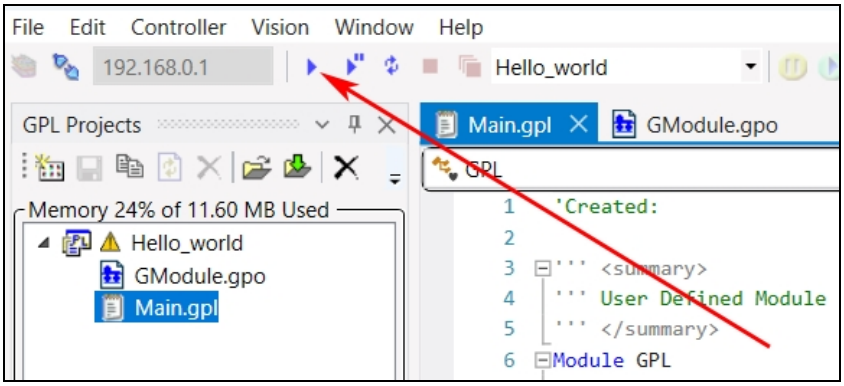
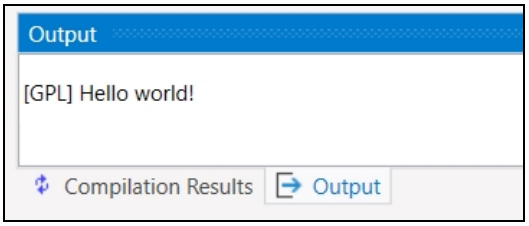
Step	Action
4.	In the GPL Projects window, expand the new **Hello_world** Project folder and double-click the **Main.gpl** file. The GPL editing window and script will display. 

Outputting Text

Follow the next set of instructions to edit the main procedure and add statements to output text.

Step	Action
1.	Double-click on the "Main.gpl" file to open the GPL editor window.

2.	In the editor window, under "Public Sub Main," insert the following lines of text:
	`console.writeline("")`
	`console.writeline("Hello world!")`
	`console.writeline("")`

Step	Action
3.	Open the Controller drop-down menu and select Output. 
4.	In the GPL main window toolbar, click "Start the Project Running." 
5.	"Hello World" should display in the *Output* window. 

Appendix B: GPL Programming Example - Point to Point Movement

A point-to-point operation moves the robot's tool tip to a position to pick up a part and to a second position to drop off the part. In this exercise, you will develop a Project that performs a simple simulated point-to-point operation.

In order to clear any possible obstacles and avoid dragging the part, the tool tip retracts after picking up the part and approaches the place location slightly above the placement position.

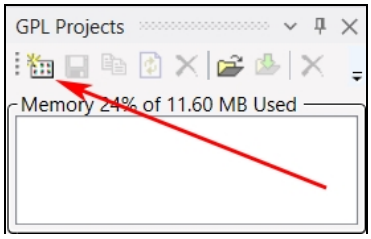
> ***NOTE:*** For this operation, the robot power must be enabled, and the robot must be homed.

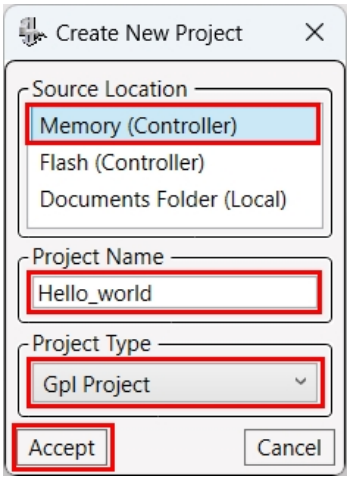
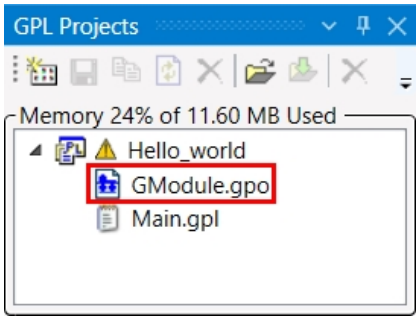
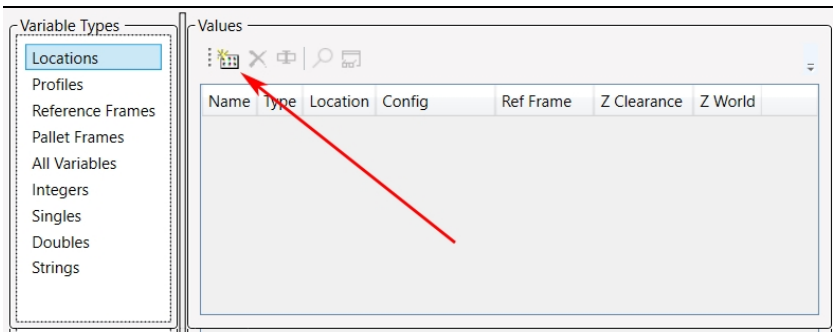
Locations

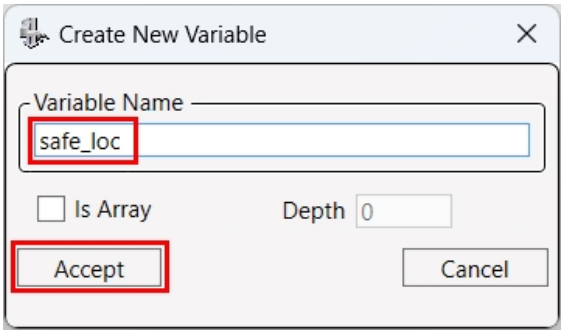
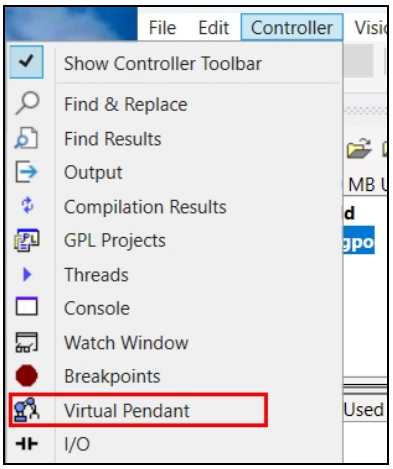
The following procedure will be for defining and teaching three motion locations to the robot:

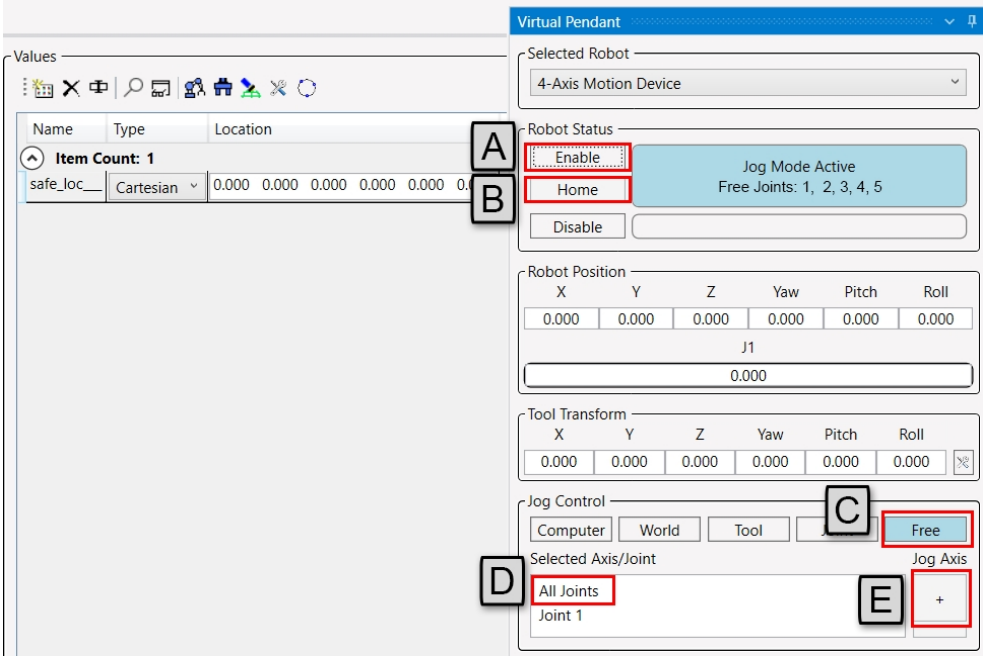
- Safe position
- Pick-up position
- Placement position.

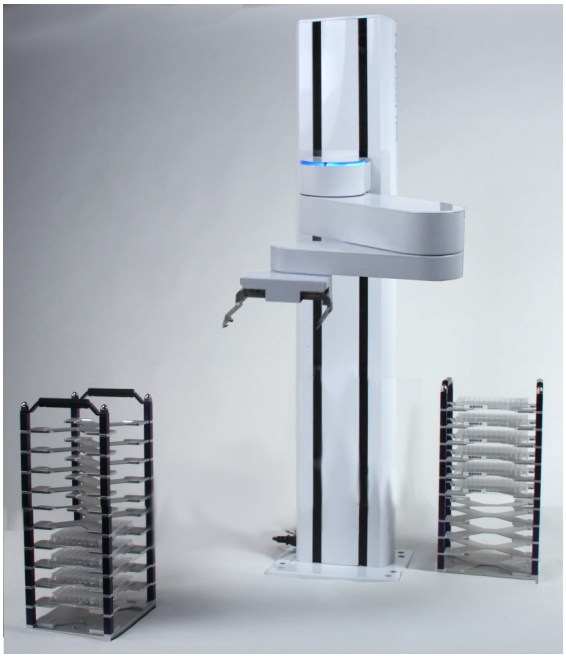
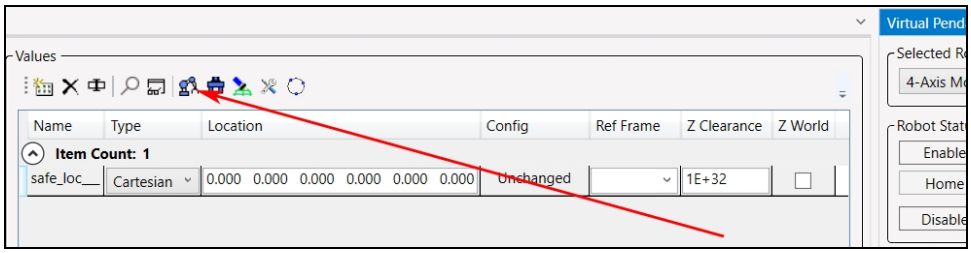
For these positions, select places in the workspace that are clear of obstacles, a minimum of 20-40 mm above the work surface.

Step	Action
1.	In the *GPL Projects* window, click **Create New Project**. 

Step	Action
2.	In the *Create New Project* window, type "Hello_world" into the *Project Name* field, select **Memory** in the *Source Location* field, and click the **Accept** button. 
3.	In the *GPL Projects* window, expand "Hello_world," and double-click on "GModule.gpo" to open the *GPO Editor*. 
4.	In the *Variable Types* window of the *GPO Editor*, select **Locations**, then, in the *Values* toolbar, click on **Add a New Variable**. 

Step	Action
5.	In the *Create New Variable* pop-up window, type "safe_loc" (for "Safe location") and click **Accept**. 
6.	Open the *Controller* drop-down menu and select **Virtual Pendant**. 

Step	Action
7.	Follow the GPL Editor steps below.  • A - Click **Enable** to enable power. • B - Click **Home** to perform the homing sequence after enabling power. If the robot has a 60 N or dual gripper, the homing sequence is performed automatically during the power-enabling phase. • C - Click **Free** to be able to move the robot arm freely. • D - In the *Selected Axis/Joints* window, select **All Joints** • E - In the *Jog Axis* window, click the plus (+) sign.

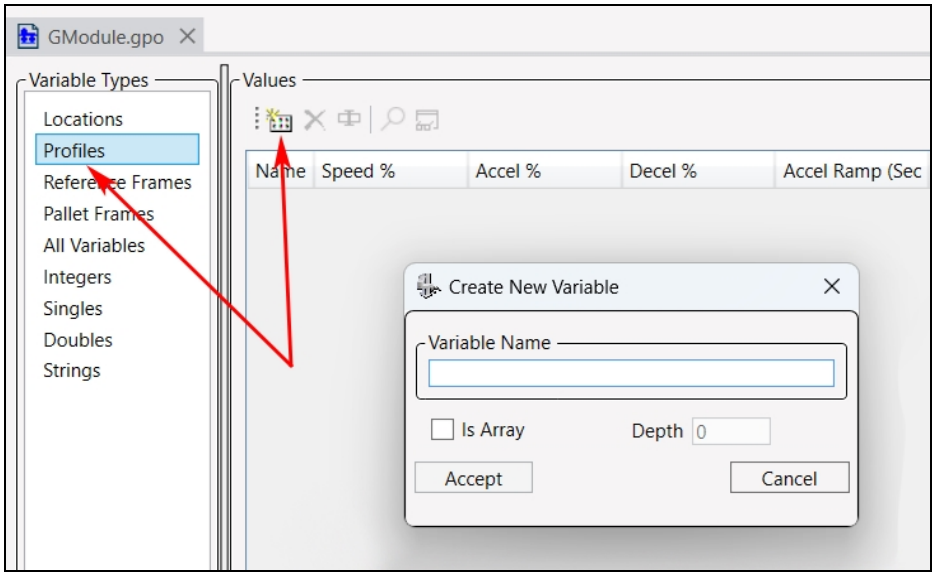
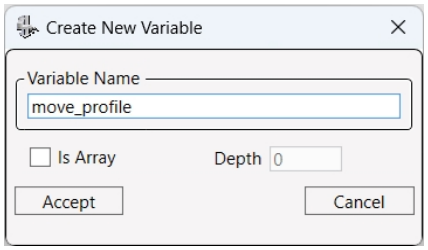
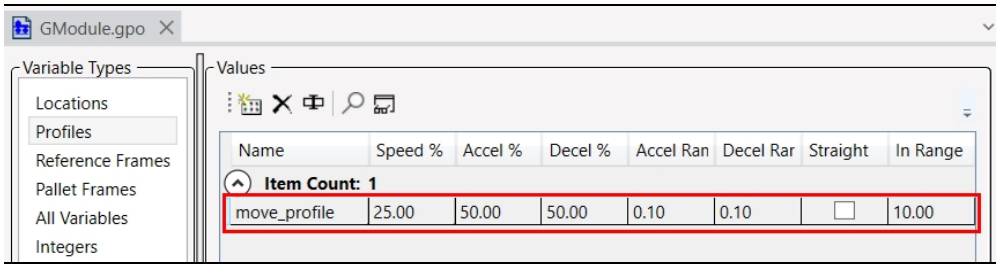
Step	Action
8.	Lift the robot arm and position it to a location above the work surface, one that can be safely reached from most positions in the work envelope.  **Safe Position**
9.	On the *Values* window toolbar, click **Record the Current Location**:  

Step	Action
10.	Now that you have recorded the "safe" location to which the robot arm will return after executing instructions, repeat the last few steps with two new variables and positions. *pick_loc* • Create a new variable called "pick_loc." • Use the *Virtual Pendant* to prepare the robot arm for positioning • Position the arm and gripper to pick up an object. • Record the position.  **Pick-Up Position**

Step	Action
11.	*place_loc* • Create a new variable called "place_loc." • Use the *Virtual Pendant* to prepare the robot arm for positioning • Position the arm and gripper to pick up an object. • Record the position.  **Placement Position**
12.	The locations get listed in the *Values* window. Values Name | Type | Location | Config | Ref Frame | Z Clearance | Z World Item Count: 3 pick_loc | Cartesian | 93.222 32.852 88.857 50.000 90.000 180.000 | Unchanged | | 1E+32 | ☐ place_loc | Cartesian | 152.000 39.852 8.857 45.000 90.000 180.000 | Unchanged | | 1E+32 | ☐ safe_loc | Cartesian | 256.060 123.526 146.234 -8.899 90.000 180.000 | Unchanged | | 1E+32 | ☐

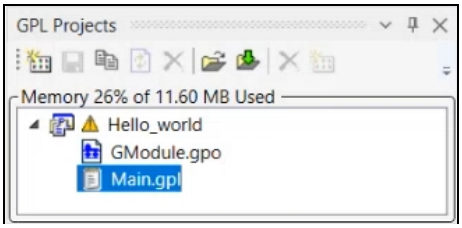
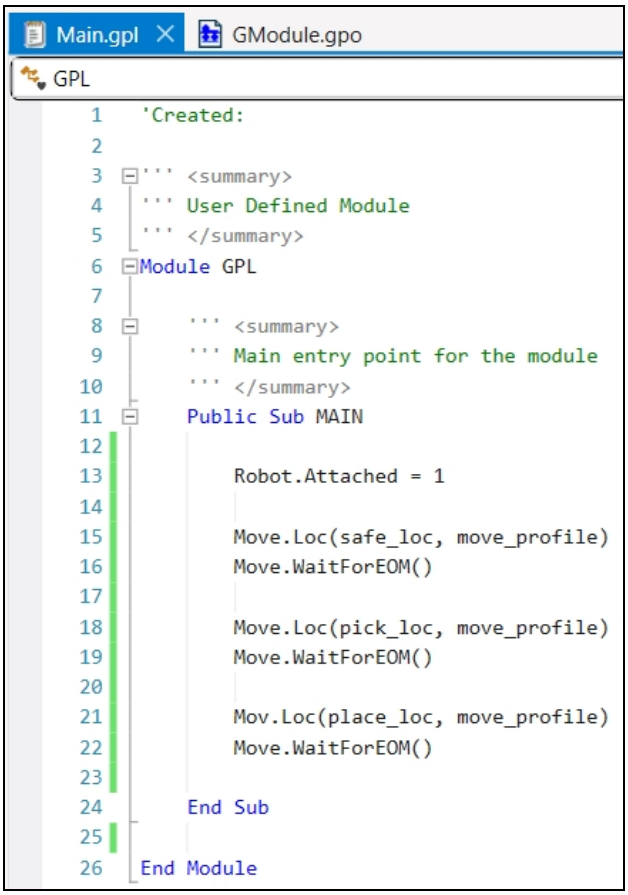
Profiles

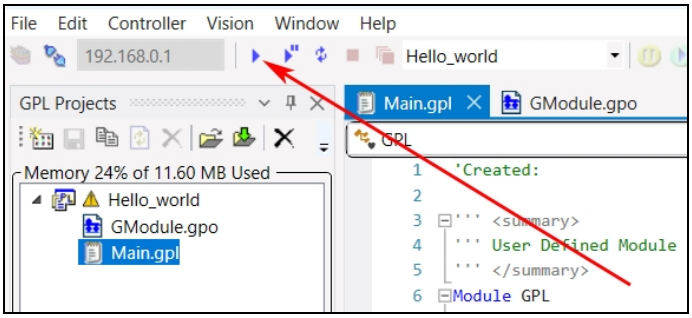
In the following procedure, create a *Motion* profile. It will be used to control the speed of the robot during the various types of motions and determine when the robot stops.

Step	Action
1.	In the *Variable Types* window, select **Profiles**. In the Values window toolbar, select **Add a New Variable**, 
2.	Type in "move_profile," and click **Accept**. 
3.	The "move_profile" display in the *Values* window along with default values. 

Write and Execute the Project

To write the GPL program that will make use of this information, complete the following procedure.

Step	Action
1.	In the *GPL Projects* window, double-click "Main.gpl" to open the *GPL Editor* window. 
2.	In the editor window, below the "Public Sub Main" statement, insert the following lines of text. ***NOTE:*** *This programming example doesn't include gripper action. Depending on your gripper, the program in this exercise may change or be different.* 

Step	Action
3.	In the GDS toolbar, click **Start the Project Running**. 

At this point in the example, the robot will move directly between the points that were created. You can modify the GPL program to include Approach motions and gripper operations based on the specifics of your application configuration.

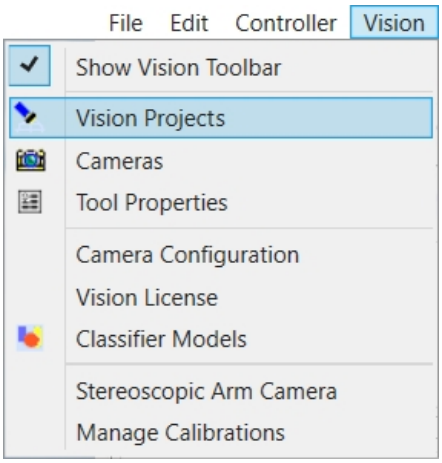
If the robot is moving safely in the workspace, the overall speed can be gradually increased in Profiles.

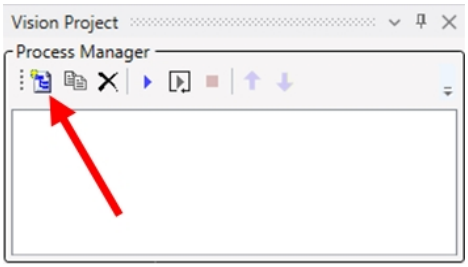
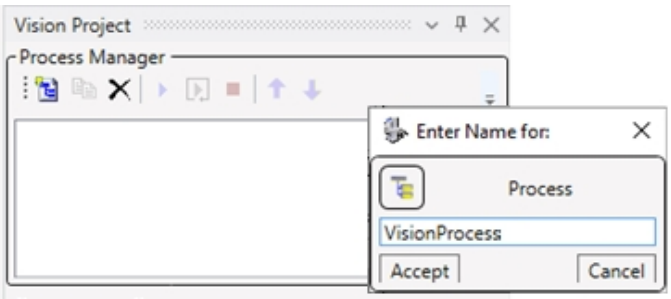
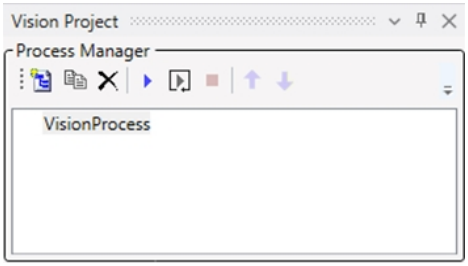
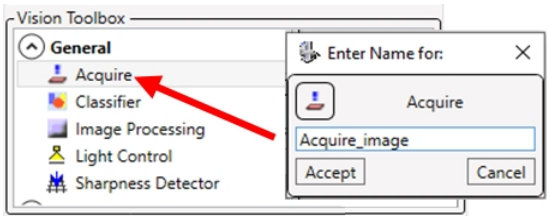
Appendix C: Vision Example - Creating an IntelliGuide Vision Project

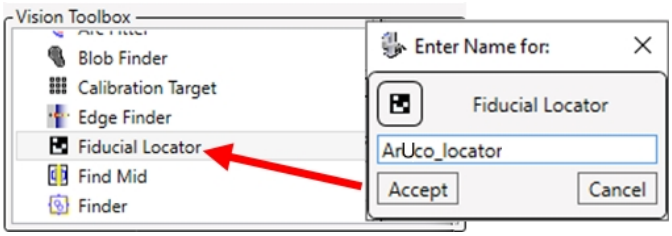
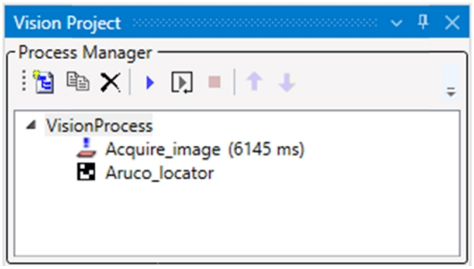
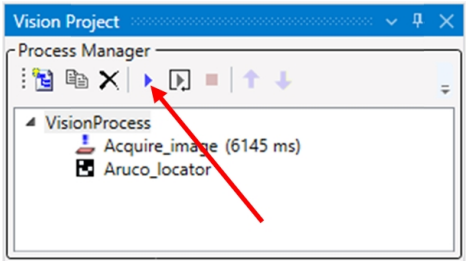
This section will take you through the steps of creating an IntelliGuide vision project and process. The procedure involves acquiring images and analyzing ArUco data, if it is present in the image, to test vision processes before integrating them into a robot's operational workflow. It's a preparatory step to ensure the vision system functions correctly before deploying it in practical robot operations.

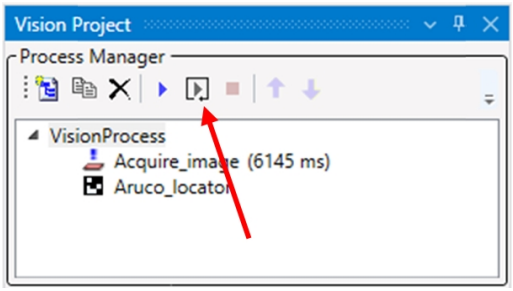
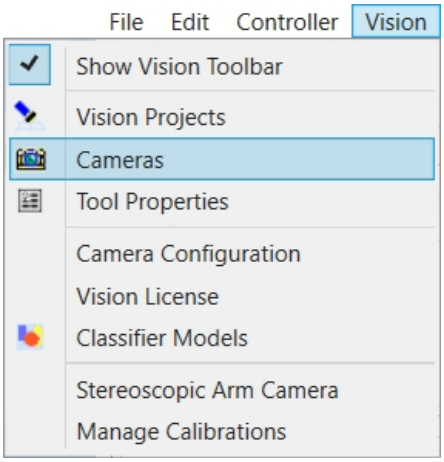
In most cases, only a single vision process is executed in order to perform the complete machine vision task. Typically, this vision process will take a picture and then utilize vision tools to locate a part and validate some key features or dimension. However, if a more complex machine vision operation is required, you can execute multiple vision processes, which can be stored in a *Vision Project*.

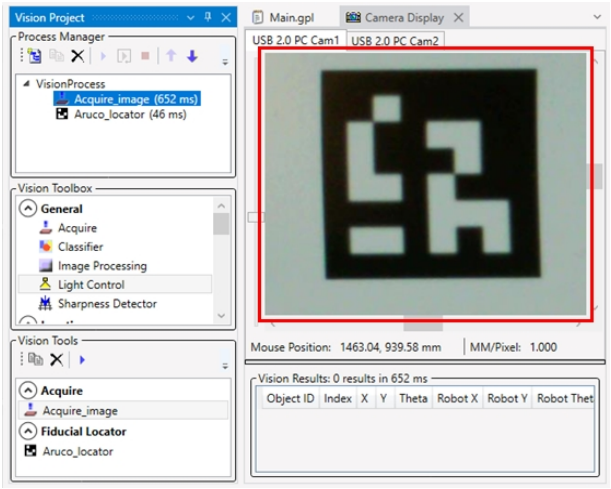
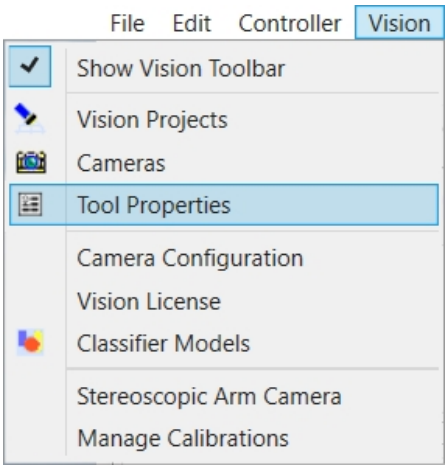
To create an IntelliGuide vision project, perform the following procedure.

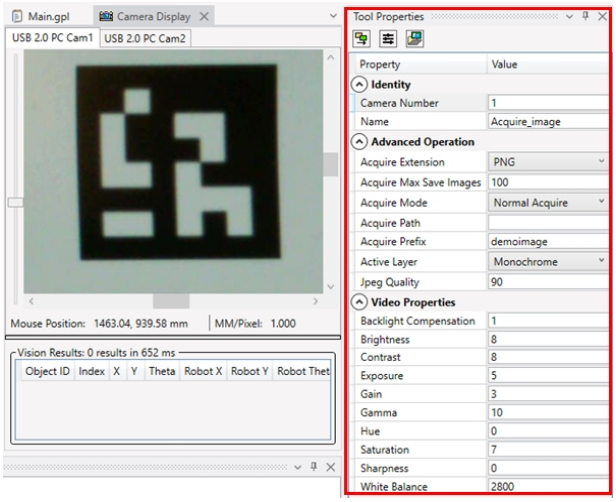
Step	Action
1.	In GDS top menu, open the **Vision** drop-down menu and select **Vision Project** to display the Vision Project window. 
2.	The Vision Project section will contain three windows: • **Process Manager**: Build and run processes. • **Vision Toolbox**: Select from various Vision Tools. • **Vision Tools**: User specific vision tools for your processes.

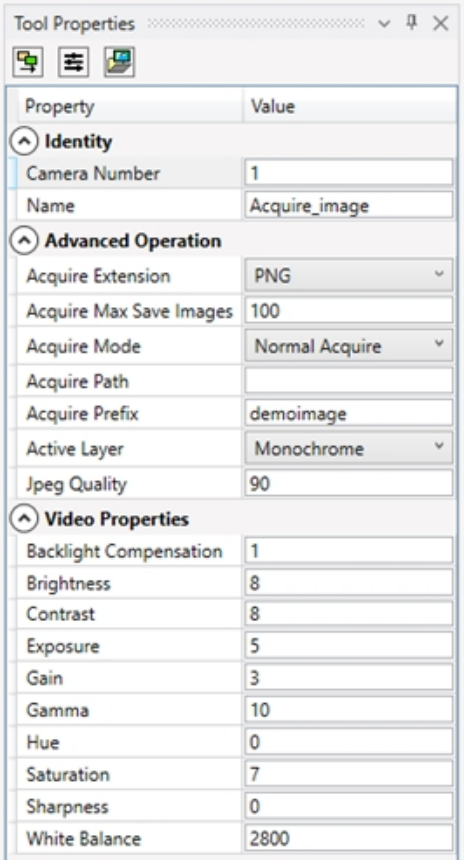
Step	Action
3.	In the **Process Manager** window, click **Create a new process** . 
4.	In the **Enter Name For:** popup window, enter a process name, any name you want, and click **Accept**. 
5.	The process name will then display in the **Process Manager** window. 
6.	In the **Vision Toolbox** window, double-click the **Acquire** vision tool to create a new vision tool. In the popup window enter any name and click **Accept**. This tool will enable the camera to take a snapshot of whatever it sees and display this image. 

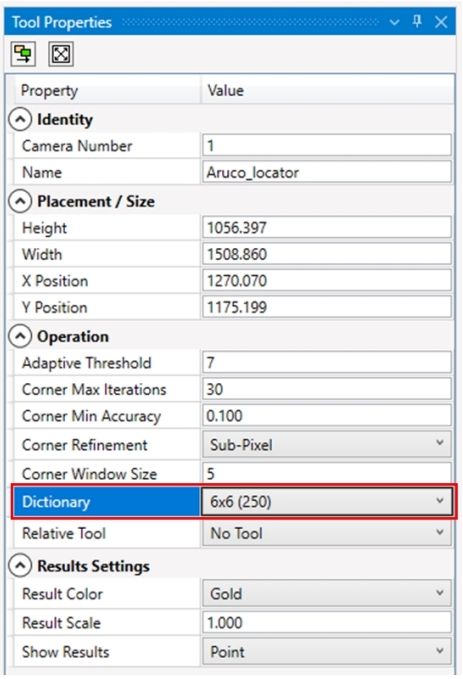
Step	Action
7.	In the **Vision Toolbox** window, double-click **Fiducial Locator**, and enter any name into the popup window. A fiducial locator detects a marker in the camera's field of view. It is used as a point of reference to help determine location. Click **Accept** when you are finished. 
8.	Drag the two newly created vision tools into the **Process Manager** window. 
9.	In the **Process Manager** window, click**Run the selected process** to test the image acquisition and the ArUco locator tools. This will execute the vision process by acquiring a new image and applying the associated vision tools. The process will run through once. ***NOTE:*** *The tools will run sequentially in the order they are listed, from top to bottom.* 

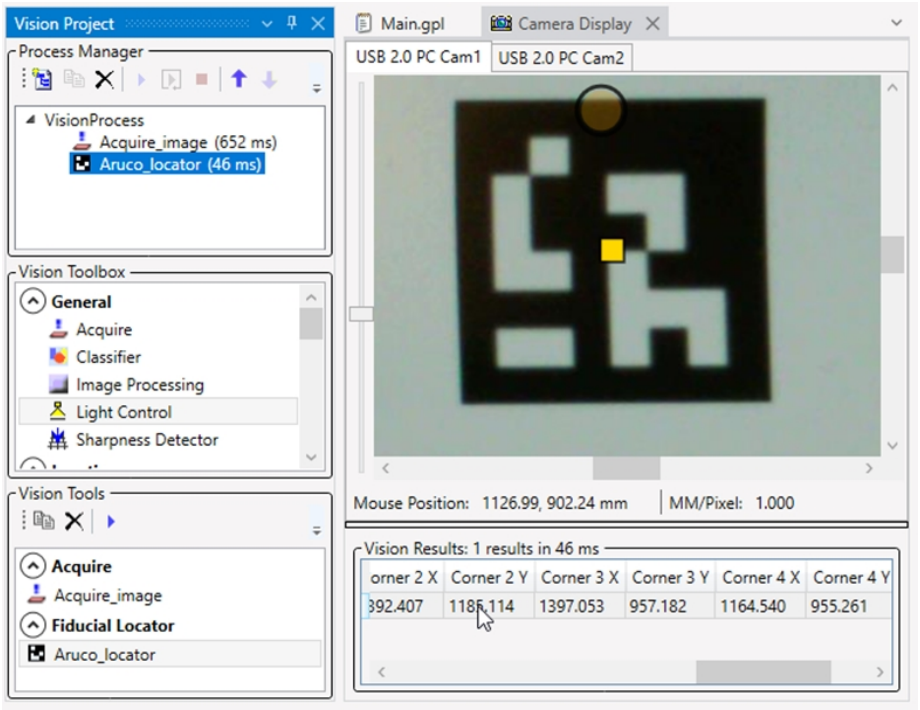
Step	Action
10.	You can also select **Run the selected process in continuous mode**, which will loop the process until you stop it. 
11.	Open the **Vision** drop-down menu and select **Cameras** to display the acquired image and the processed results. ***NOTE:*** *The acquired image is a snapshot the camera took of what it sees.* 

Step	Action
12.	The image displays. In this example, the ArUco displays what the camera sees. The results, camera coordinates, and robot coordinates, are displayed in the window below the image. 
13.	To adjust the vision tool properties, open the **Vision** drop-down menu and select **Tool Properties**. 

Step	Action
14.	On the right side of the screen, the **Tool Properties** window will display the properties of the acquired image. 

Step	Action
15.	You may need to adjust the acquisition parameters in the Tool Properties window based on the environment where the robot is located. The default parameters are only used for reference and a starting point when setting up the vision application.. The default settings for the Image Acquisition tool are shown below.  Tool Properties Property / Value **Identity** Camera Number: 1 Name: Acquire_image **Advanced Operation** Acquire Extension: PNG Acquire Max Save Images: 100 Acquire Mode: Normal Acquire Acquire Path: Acquire Prefix: demoimage Active Layer: Monochrome Jpeg Quality: 90 **Video Properties** Backlight Compensation: 1 Brightness: 8 Contrast: 8 Exposure: 5 Gain: 3 Gamma: 10 Hue: 0 Saturation: 7 Sharpness: 0 White Balance: 2800

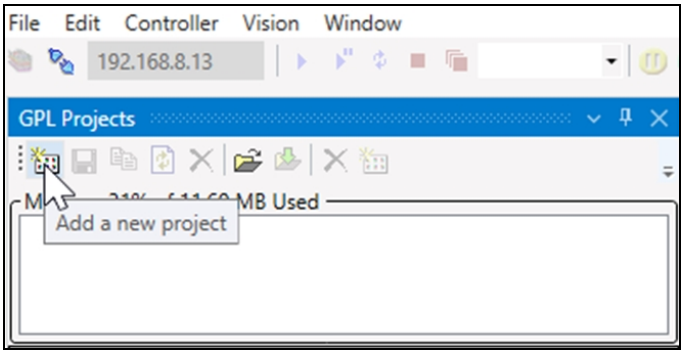
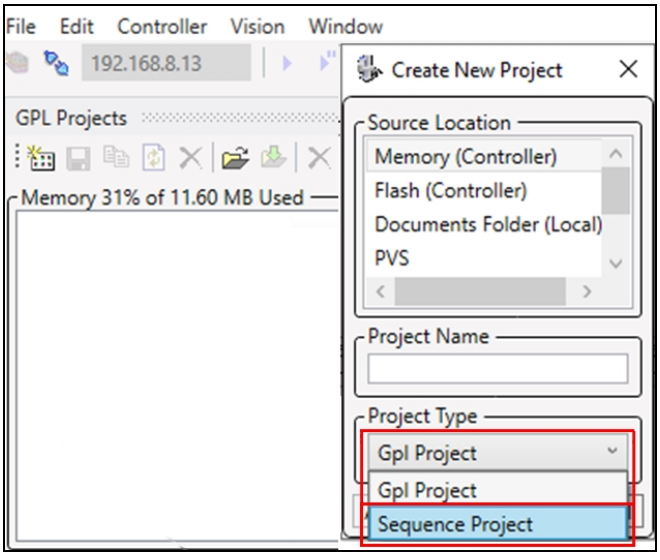
Step	Action
16.	Use the correct **ArUco Dictionary** for setting up the Fiducial Locator tool. For each teach plate that Brooks supplies, a 6x6 ArUco dictionary is used. ***NOTE:*** *If you change the dictionary, make sure to select the correct one. There are few different ArUco dictionaries, including:* • *4x4 - contains 50 ArUco markers* • *5x5 - contains 100 ArUco markers* • *6x6 - contains 250 ArUco markers* • *7x7 - contains 1000 ArUco markers* 

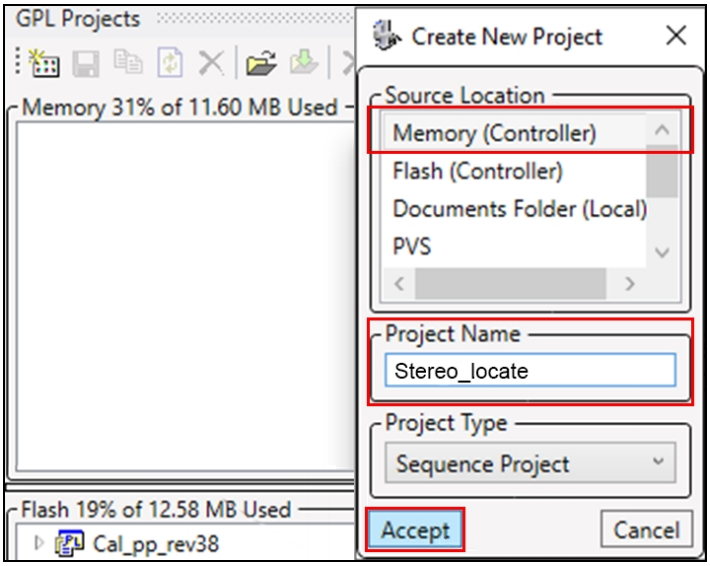
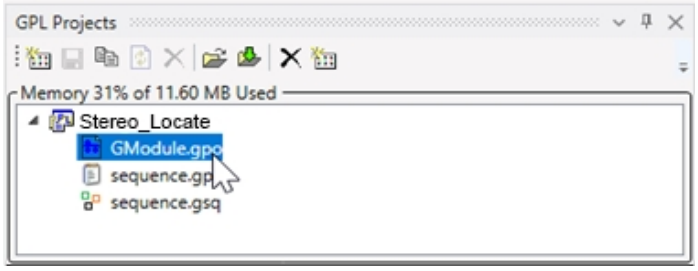
Step	Action
17.	If the **Acquire** and **Fiducial** tools are configured correctly, the image should look clear and the results from the tool will be displayed when the user selects the vision tools. 

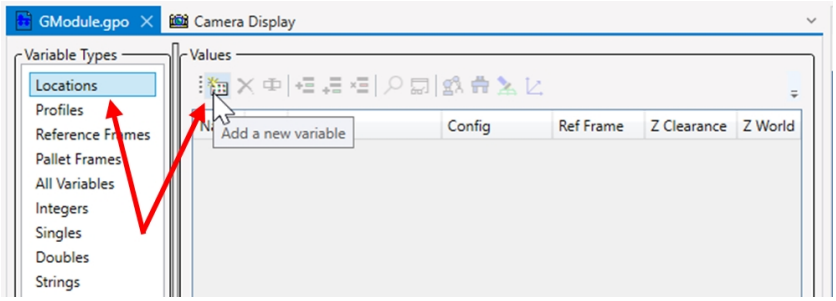
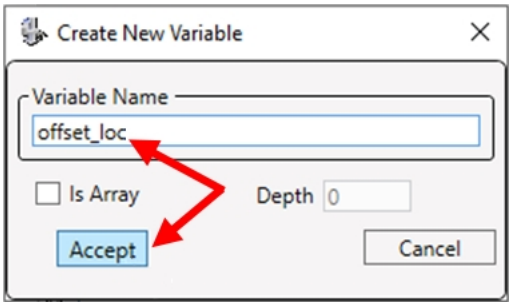
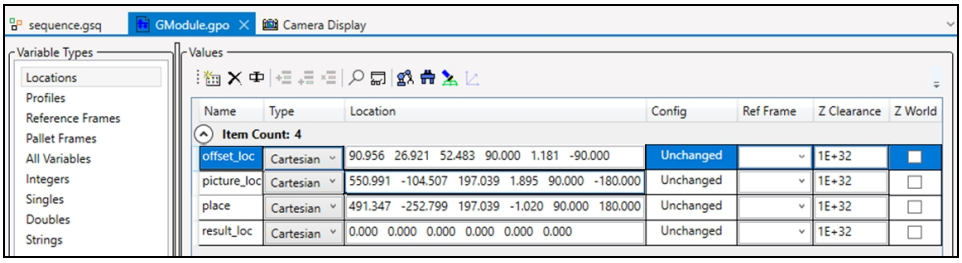
Appendix D: Vision Example - Creating an IntelliGuide Vision Offset

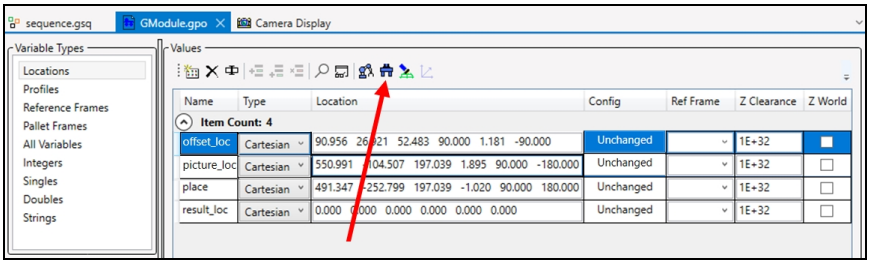
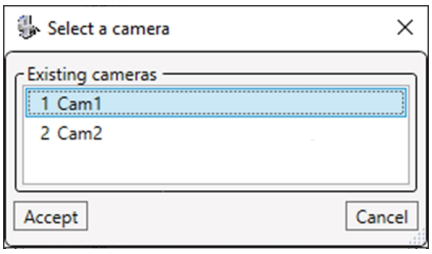
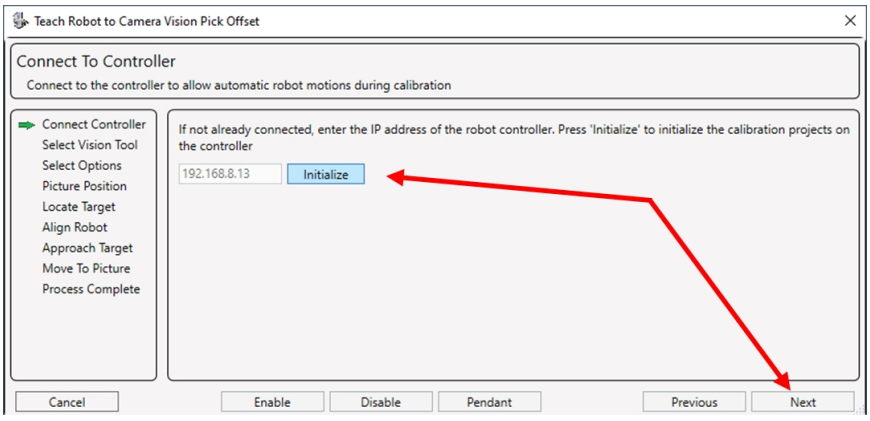
A vision offset guides the robot's movement from the midpoint to pick up the object, ensuring accurate interaction.

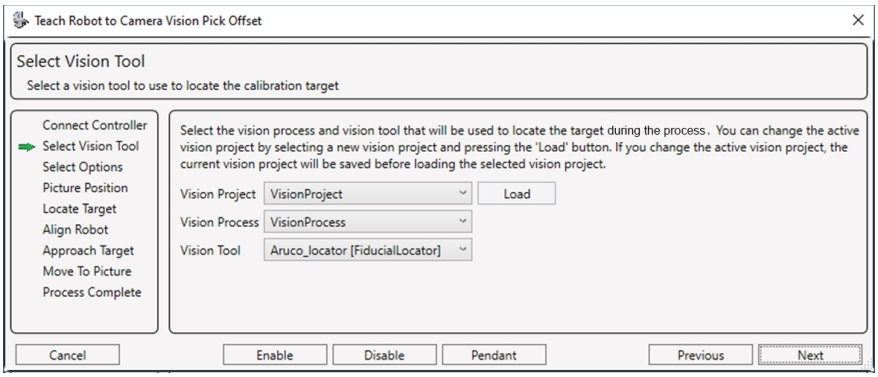
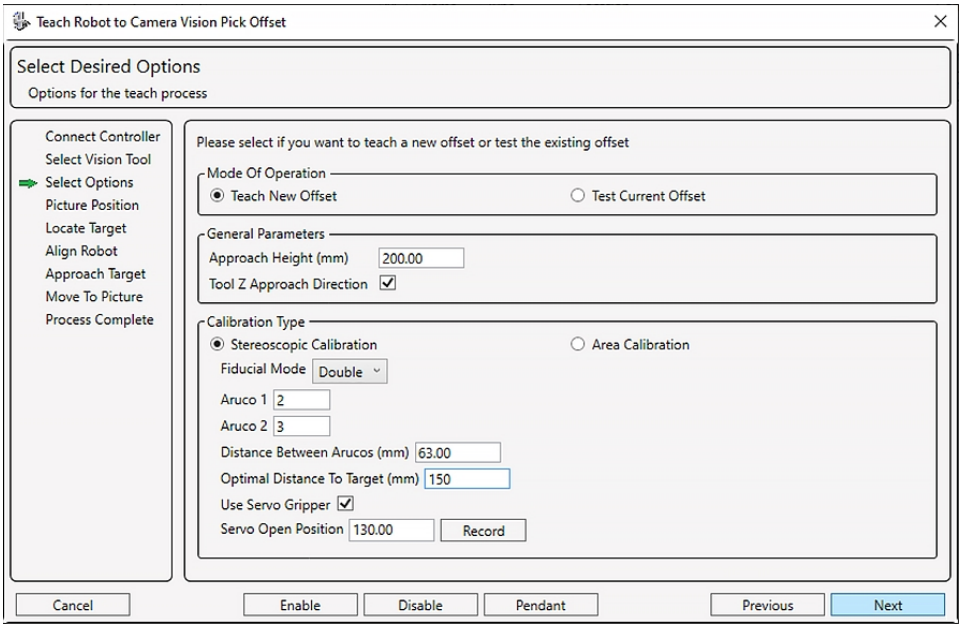
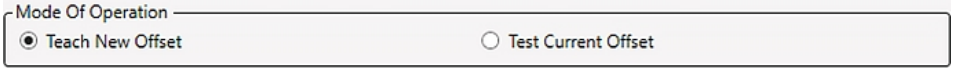
Perform the following procedure to create an IntelliGuide vision offset.

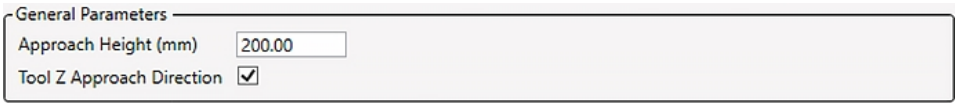
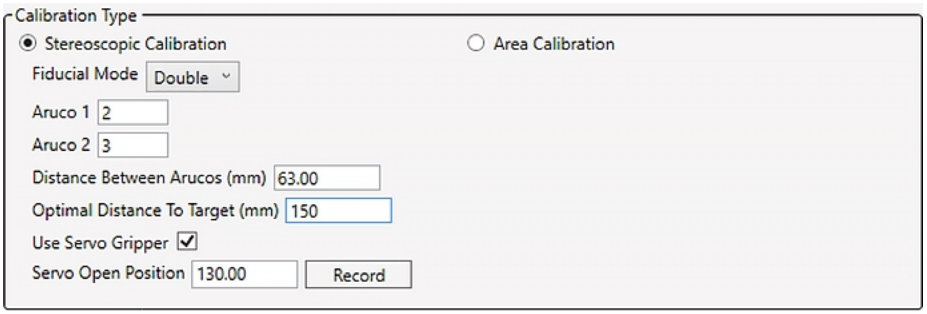
Step	Action
1.	In the **GPL Project** window, click **Create new project**. 
2.	In the Create New Project pop-up window, open the **Project Type** drop-down menu and select **Sequence Project**. 

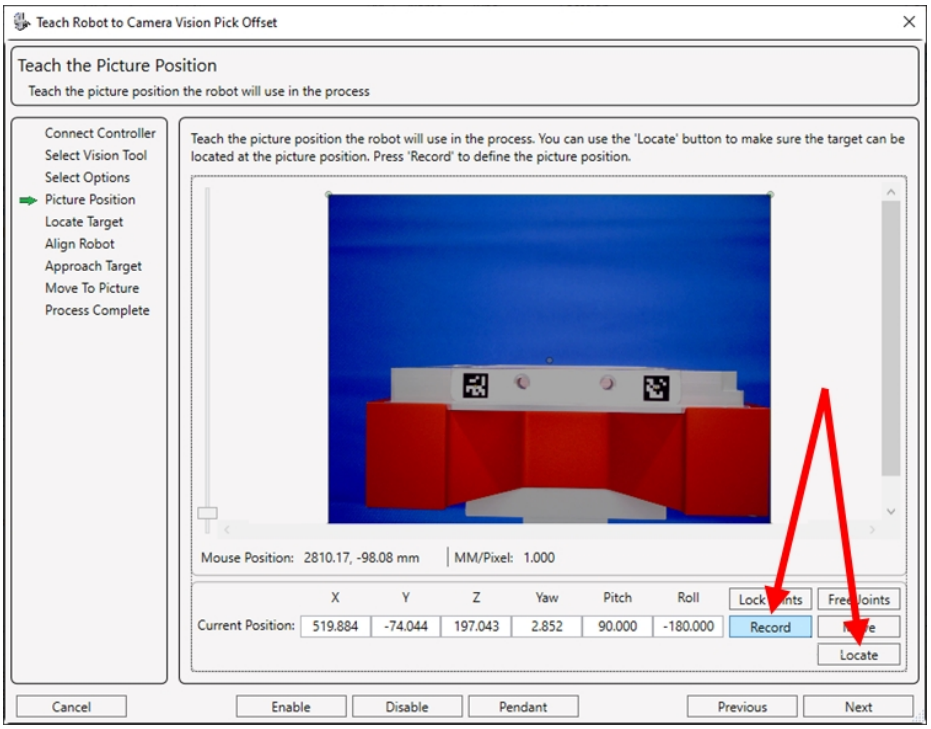
Step	Action
3.	• Select a destination window for the project: *Memory*, *Flash*, or *Local*. For this example, we selected the *Memory* window. You can click and drag the project from one window to another. • Type in a **Project Name**. In this example, we named the project "Stereo_ locate." • Click **Accept**. 
4.	Expand the **Stereo_locate** project, double-click on **GModule.gpo** to add and edit location. 

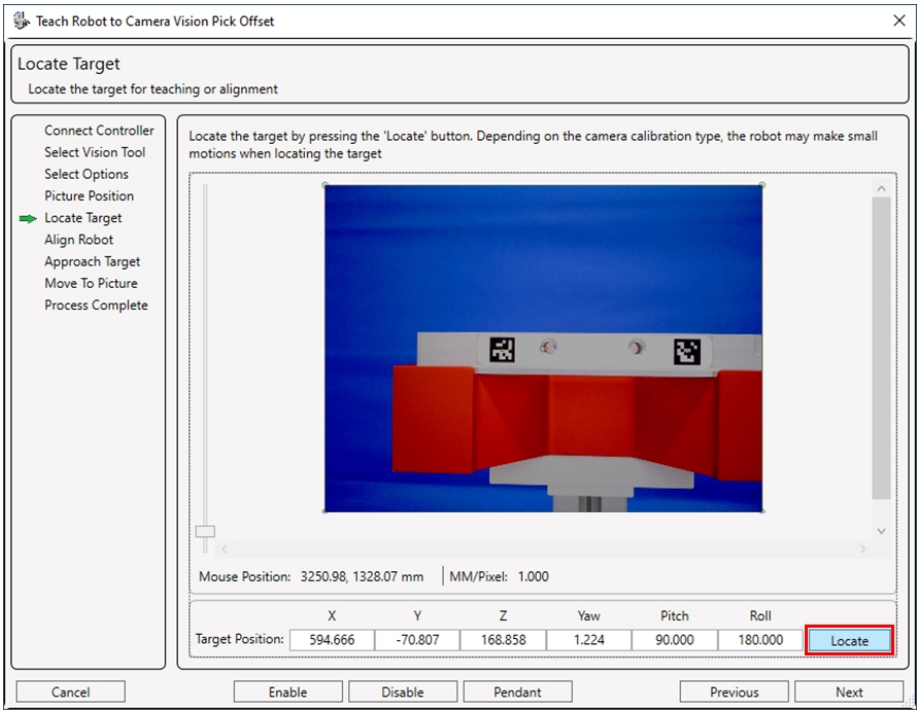
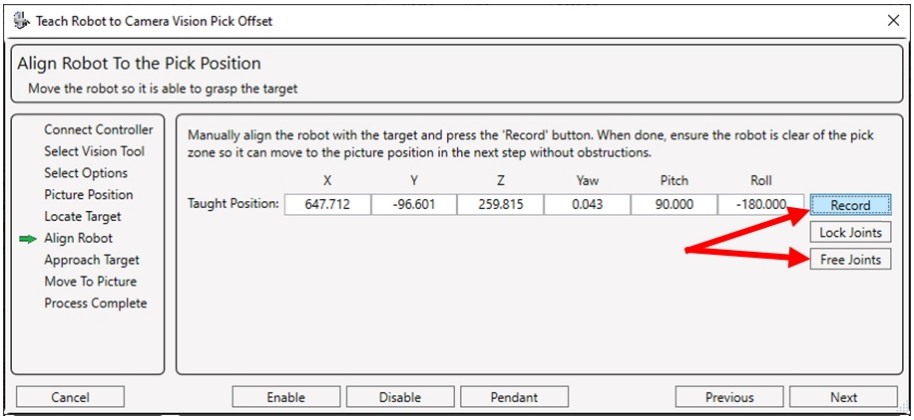
Step	Action
5.	• In the **GModule.gpo** window, in the *Variable Types* , **Locations**. • With Locations selected, in the *Values* window toolbar, click **Add new variable** to add a new location. 
6.	• In the **Create New Variable** pop-up window, add a **Variable Name** to the location. In this example, we named the Variable Name "offset_loc." • Click **Accept**. 
7.	Repeat the preceding steps to add a few more locations. 

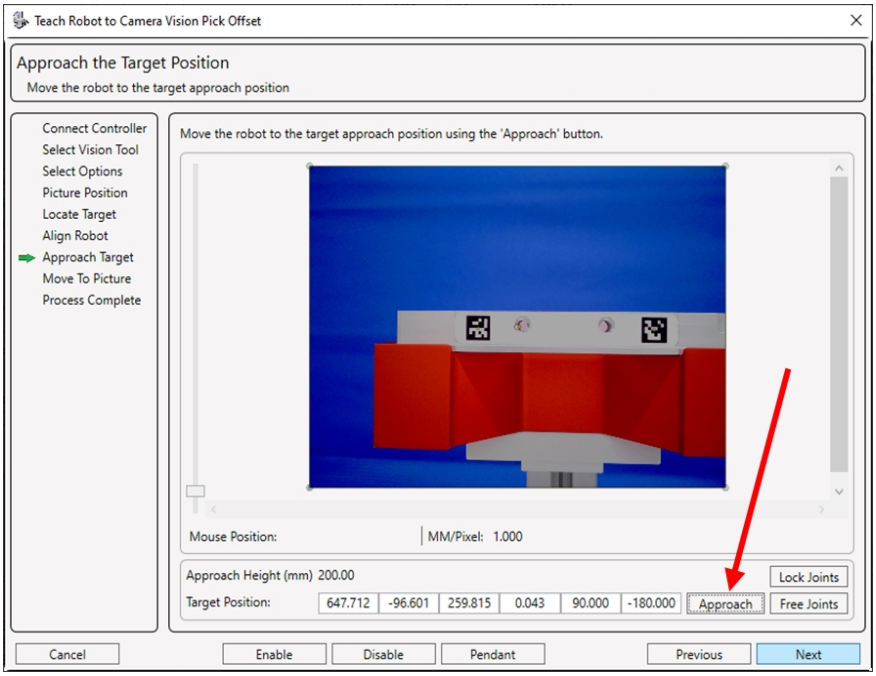
Step	Action
8.	To access the **Vision offset wizard**, select the desired location that will become your vision pick offset. This location will be overwritten by the wizard. Click the **Teach vision pick offset** button in the **GModule.gpo** window. 
9.	In the **Select a camera** popup window, select the camera number (for this example, you will use camera 1, the front camera), and click **Accept**. 
10.	In the next window, click the **Initialize** button to load the required system files and initialize the vision system, then click **Next**. 

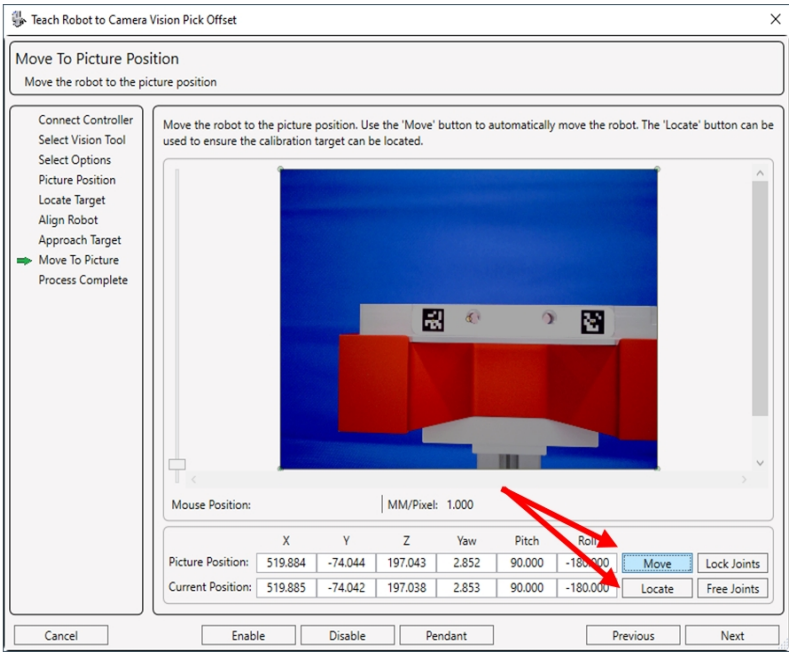
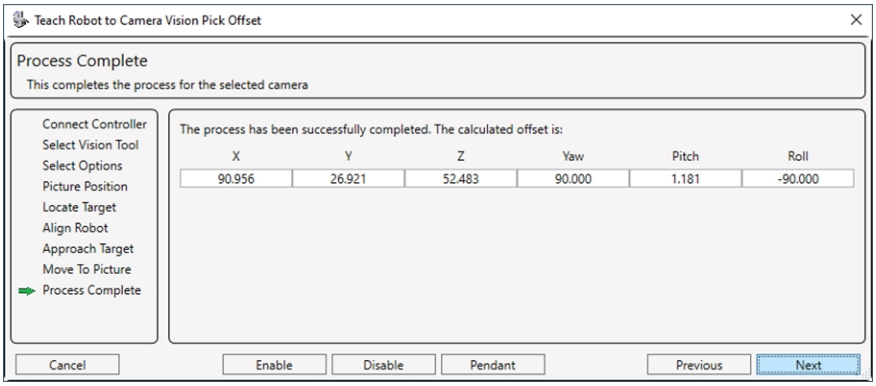
Step	Action
11.	In the Select a Vision Tool window, load and select the related **Vision Project**, **Vision Process**, and **Vision Tool**. Click **Next** when you are finished. ***NOTE:*** *Be specific. If you have multiple vision projects, the chosen tools are crucial for the offset's accuracy. In this case, for example, the ArUco locator tool is required to detect the offset effectively.* 
12.	Make selections on the **Select Desired Options** page. 
13.	In the **Mode of Operation** section of the options page, select the option for **Teach New Offset** or **Test Current Offset**. 

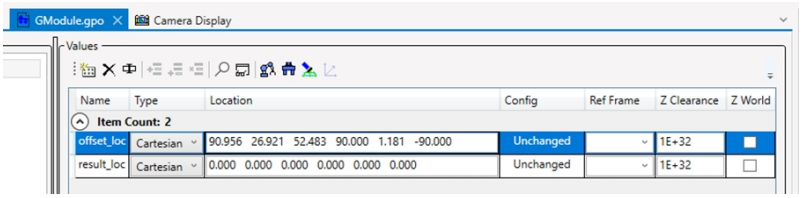
Step	Action
14.	In the **General Parameters** section, define the safe approach height. ***NOTE:*** *Safe approach height refers to the distance between the robot's gripper and the target object, ensuring that the gripper's fingers do not collide with the target during image acquisition. It allows the robot to have clear visibility of the target while avoiding any potential collisions during the picking process.* 
15.	In the **Calibration Type**, specify the Fiducial mode, single or double. This example shows double ArUco detection. Define the ArUco numbers, the distance between the fiducial markers, and the optimal distance to the target. Define the servo open position, how much the gripper fingers should open in order for the gripper to safely go around the target. When you are finished, click **Next**. 

Step	Action
16.	In the Teach the Picture Position window, click **Record** to save the current location that will be used to take a picture of the teach plate with the two ArUcos labels. The location should be recorded so the two ArUcos are in the field of view, and the distance matches the originally configured optimal distance of 150 mm. Click **Locate** to take the image and confirm that the location is appropriate for the operation. Click **Next** when you are finished. 

Step	Action
17.	In the Locate Target window, click **Locate**. This is an automated operation, and the robot will enable power and move so that it can take stereo images of the target. Click **Next** when you complete the operation. 
18.	In the Align Robot to the Pick Position window, free the joints by clicking **Free Joints** and aligning the end effector with the target to be picked. Click **Record** when the end effector is perfectly aligned. This is critical step for setting up a proper offset. Click **Next** when you are finished. 

Step	Action
19.	In the Approach the Target Position window, click **Approach** to approach the target. The robot will move and approach the target. Click **Next** when you are finished. 

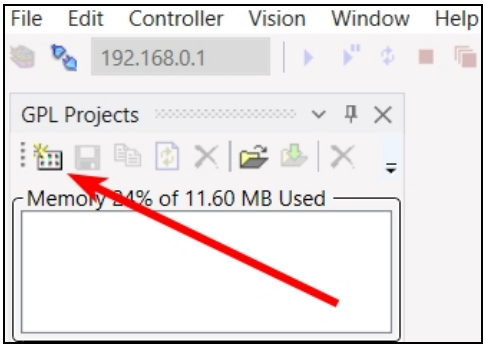
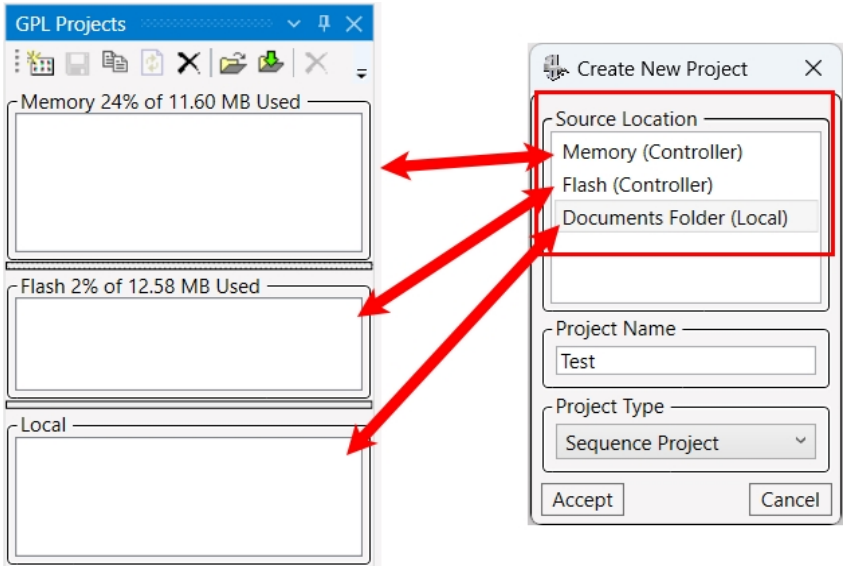
Step	Action
20.	In the Move to Picture Position window, click the **Move** button to move to the picture location. At this stage, you can use **Locate** to ensure the target can be located and the label is in the camera's field of view. Click **Next** after you complete the motion. 
21.	After you click Next, the Process Complete window will display. 

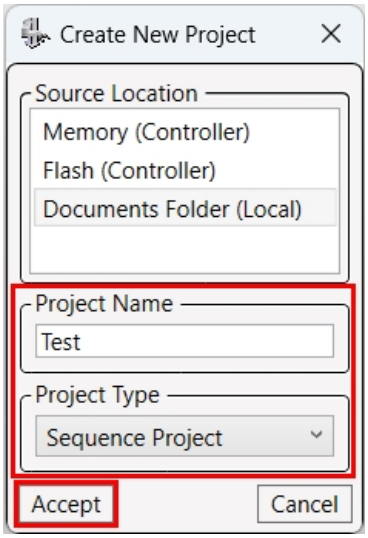
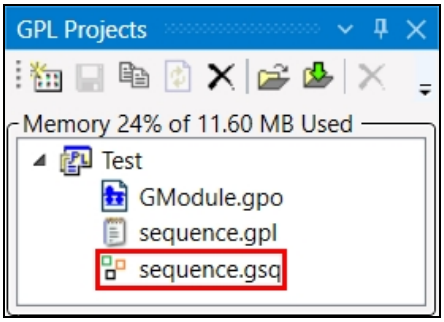
Step	Action
22.	Confirm that the offset was recorded in the selected location variable in the GModule.gpo window. 

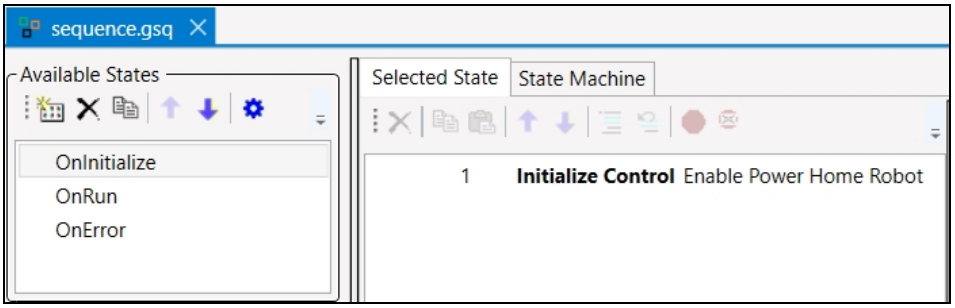
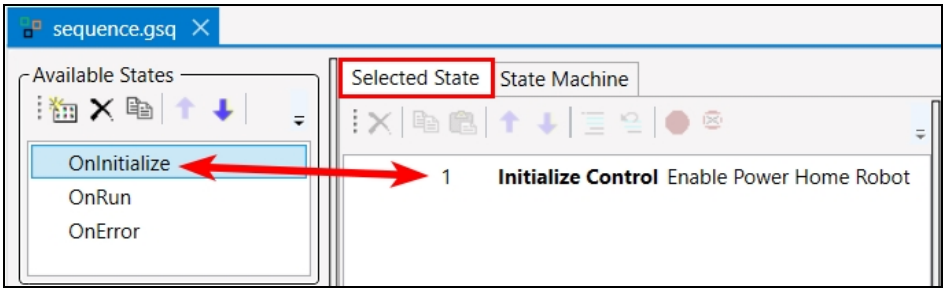
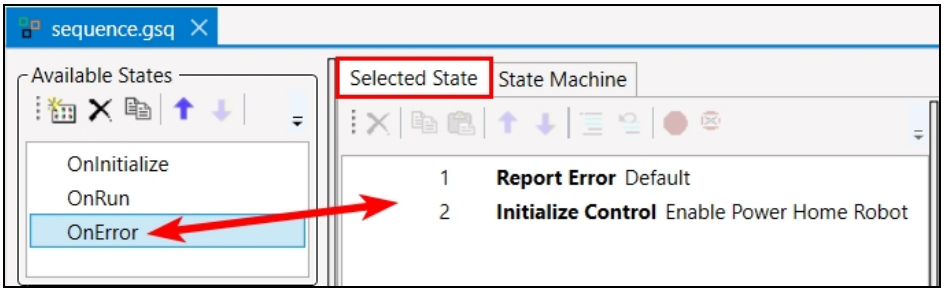
Appendix E: GP Flow Programming Example: Pick and Place

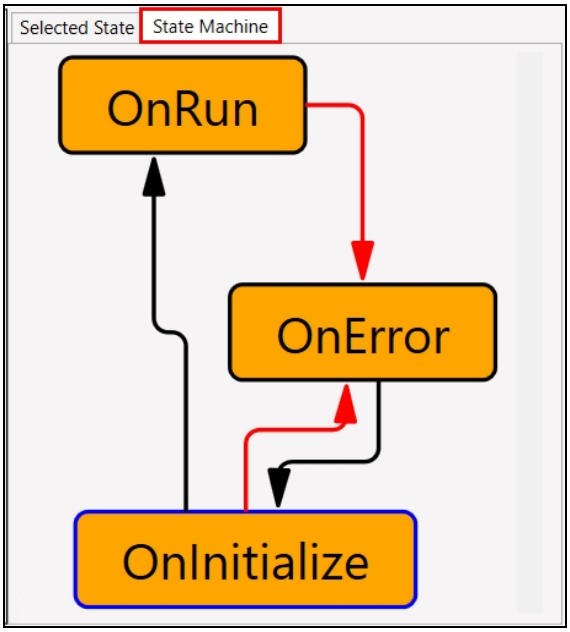
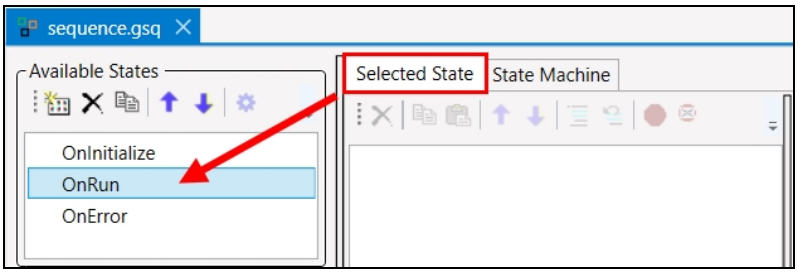
GP Flow is a graphical programming language for programming the robot without using structured text such as Python or C#. The graphical programming is compiled into GPL code, which then runs on the controller.

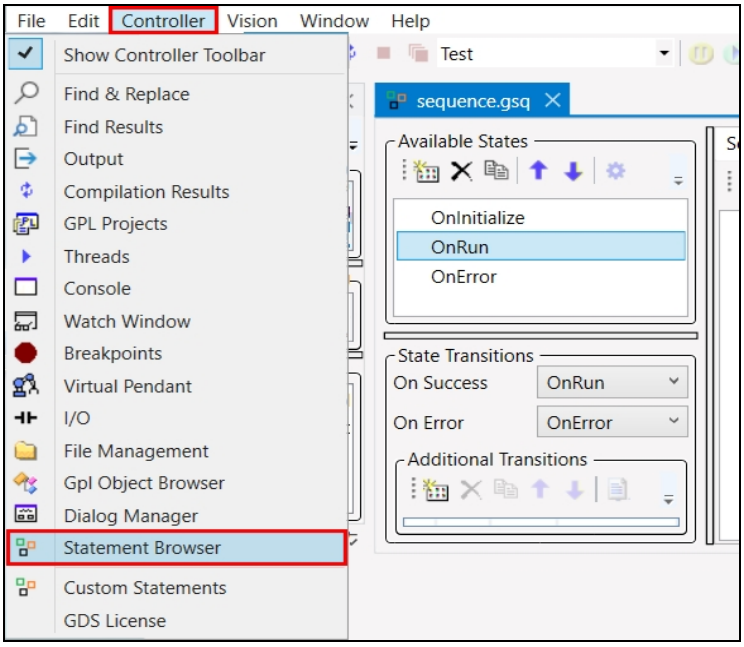
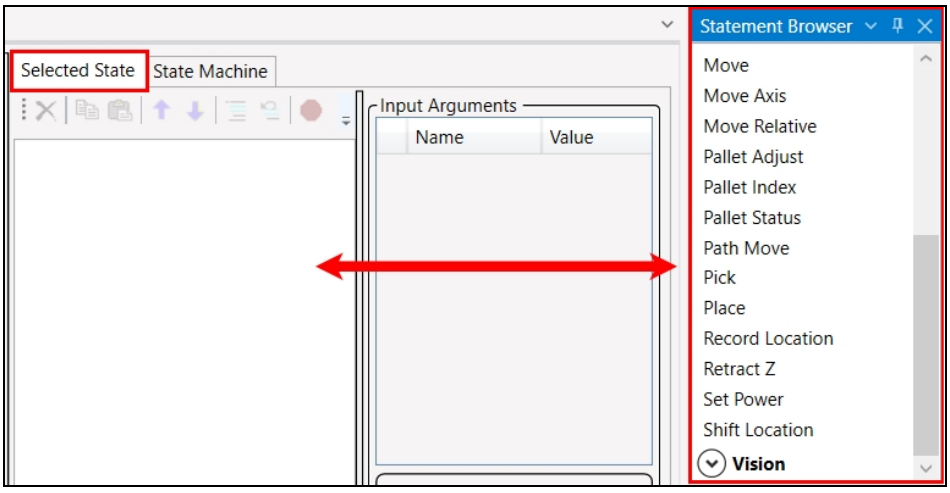
This is an example GP Flow exercise that shows the basic steps for programming the robot to pick up an object from one location and place it somewhere else.

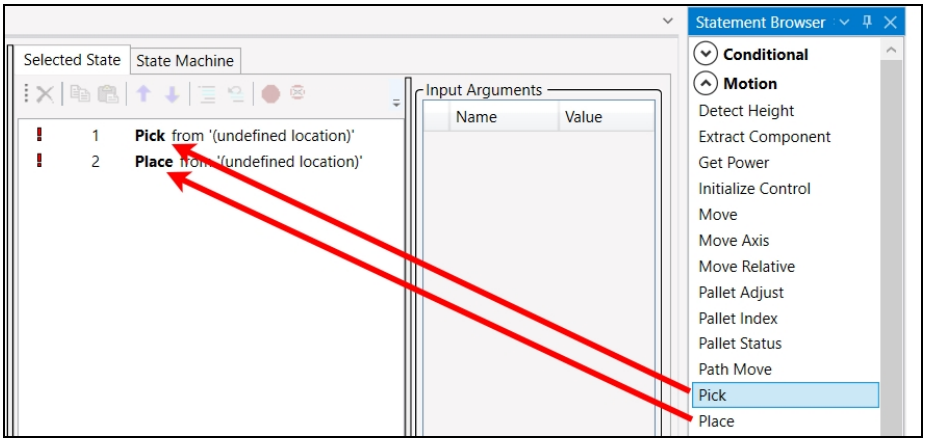
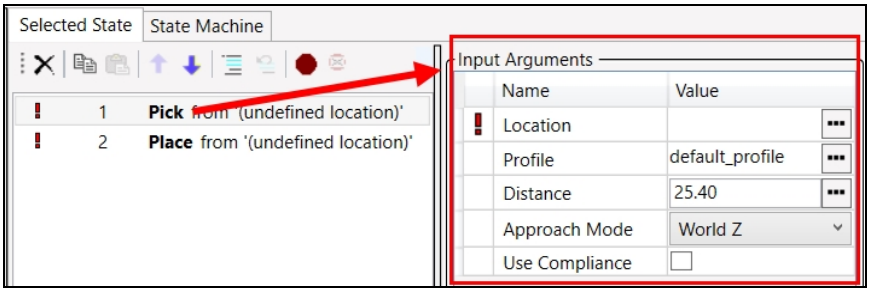
Step	Action
1.	In the *GPL Projects* toolbar, click on **Create New Project**. 
2.	The *Create New Project* window will display. Select a *Source Location* window to put the new project: *Memory*, *Flash*, or *Local*. 

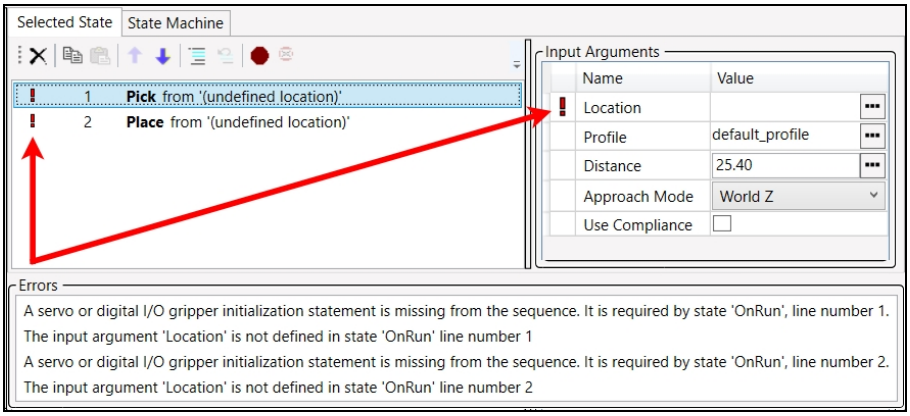
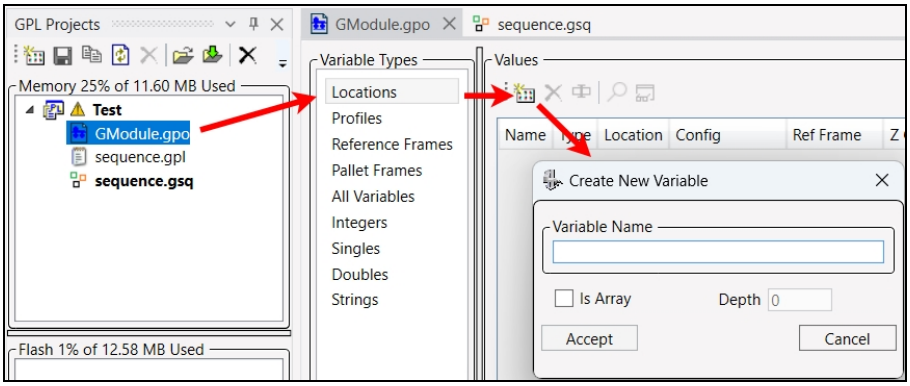
Step	Action
3.	• In the *Project Name* field, type in a name for the project. For demonstration purposes, title this project *Test*. • From the *Project Type* drop-down menu, select **Sequence Project**. • Click **Accept**. 
4.	The *Test* project will contain three files: • GModule.gpo • sequence.gpl • sequence.gsq Double-click on sequence.gsq. 

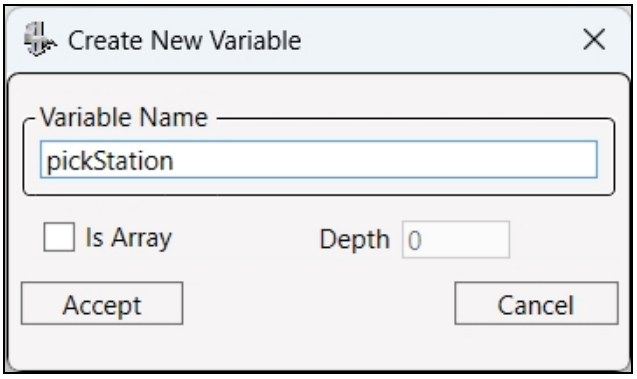
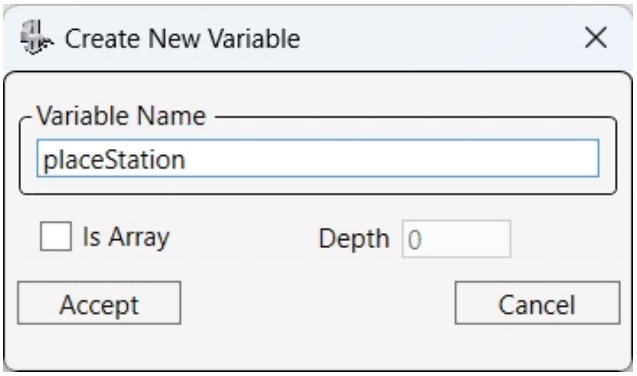
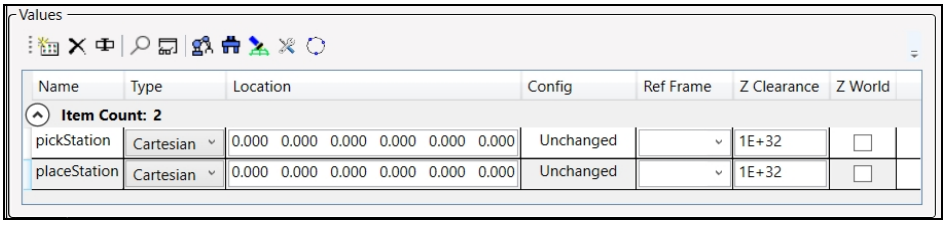
Step	Action
5.	The main sequence editor window will open. 
6.	In the *Available States* window (shown here on the left), when you click on **OnInitialize**, **OnRun**, or **OnError**, the contents of each will display in the *Selected State* window.  

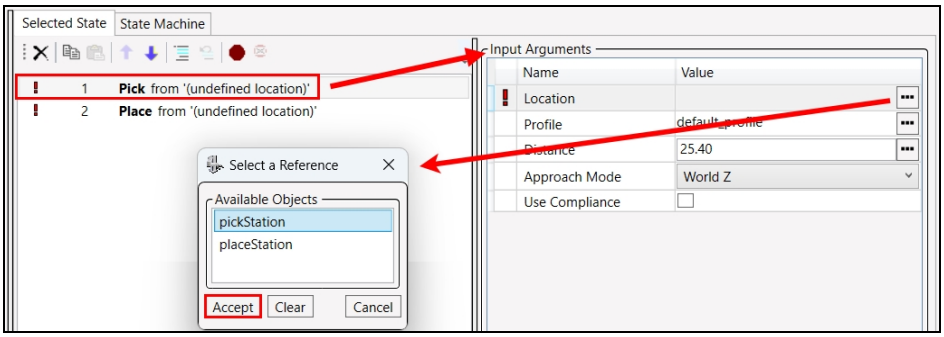
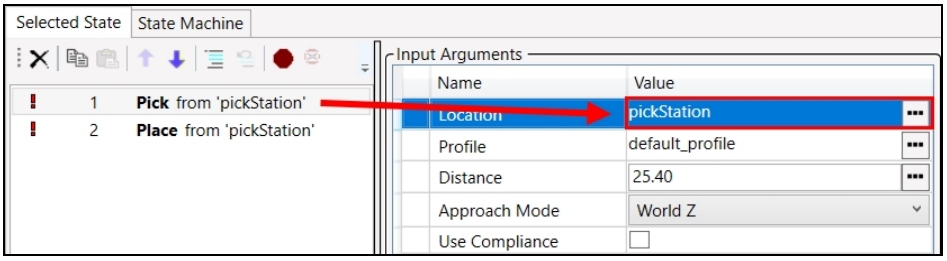
Step	Action
7.	Click on the **State Machine** tab. A graphic representation of the *Available States* will display the states' sequence of activity. In the example below, the black lines represent normal transitions, and the red lines indicate error transitions. If the activated **OnInitialize** state is OK, it will activate the **OnRun** state and run the program. If there is an error with **OnItialize** or **OnRun**, the **OnError** state will activate a corrective measure. 
8.	Click the *Selected State* tab and, in the *Available States* window, click the **OnRun** State. The *Selected State* window will be empty; you will build the program there. 

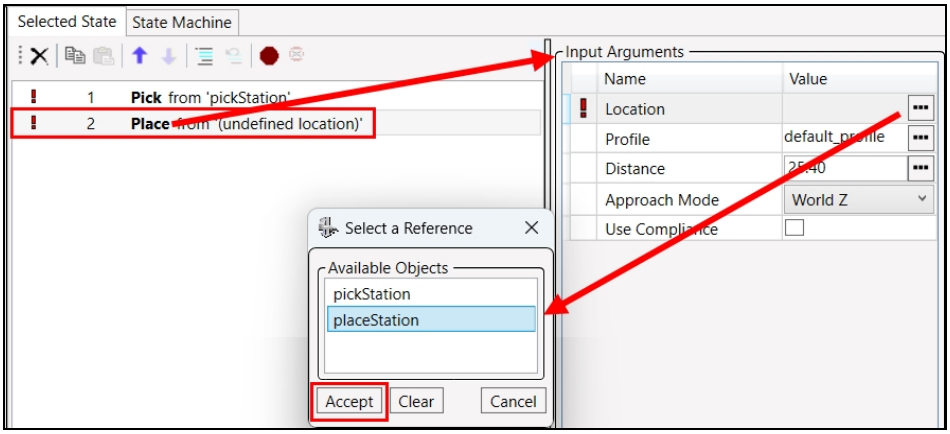
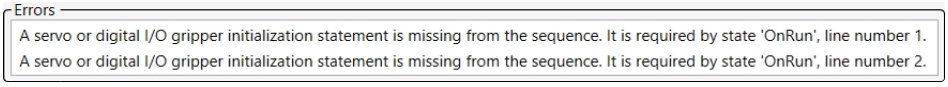
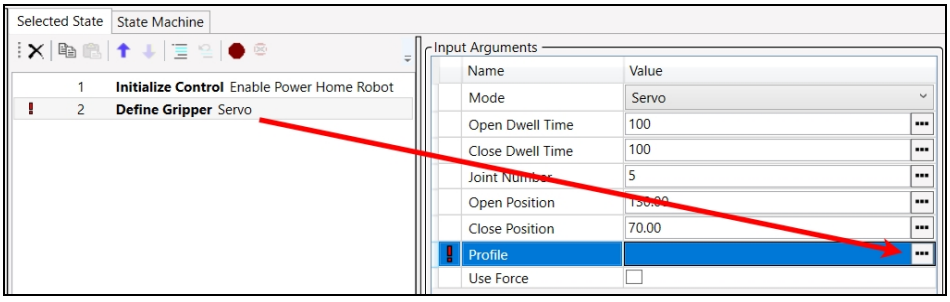
Step	Action
9.	Open the **Controller** drop-down menu and select **Statement Browser**. 
10.	The *Statement Browser* will display, shown here on the right side of the screen. It contains statements that can be dragged into the *Selected State* window to assemble a program. Statements in the *Statement Browser* are organized by category, shown in bold: Conditional, Data, General, Gripper, I/O, Motion, and Vision. 

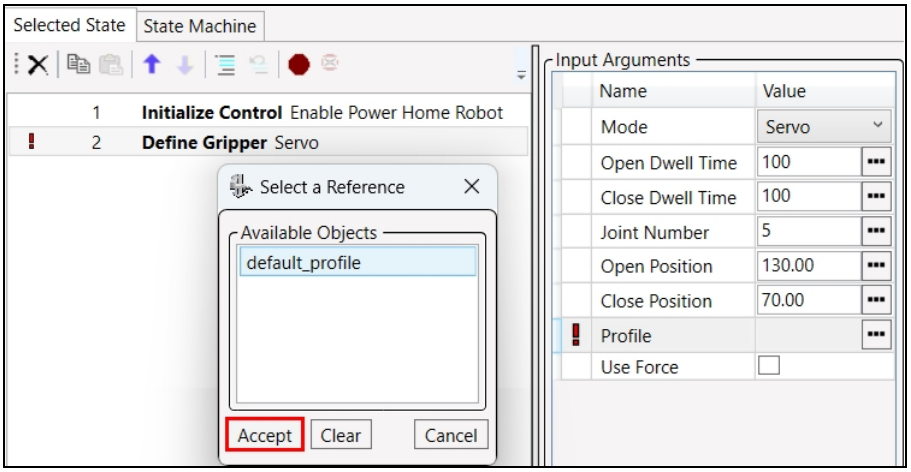
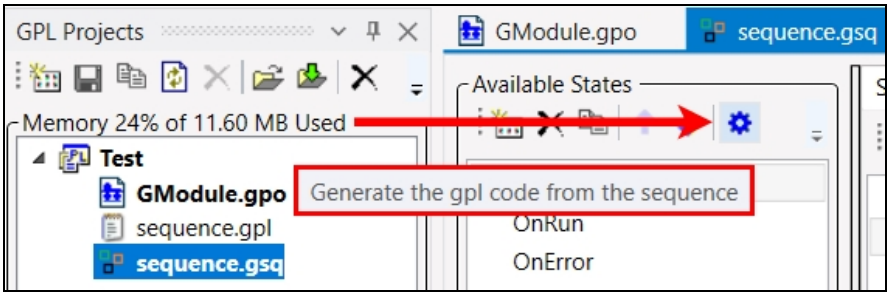
Step	Action
11.	Click on the **Pick** statement and drag **Pick** from the *Statement Browser* (Motion category) to the *Selected State* window. Do the same thing for **Place**. 
12.	Add input arguments to each statement. Double-click on **Pick** to display its *Input Arguments* window. 

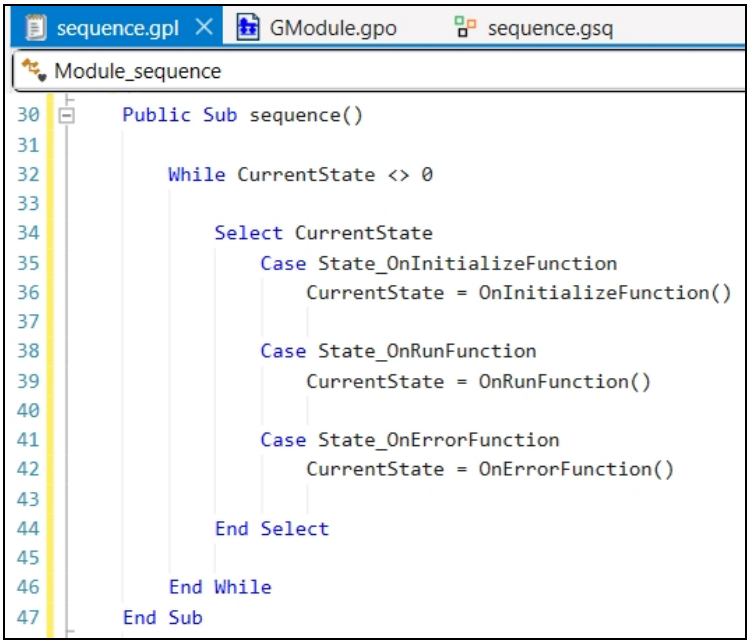
Step	Action
13.	A red exclamation point indicates an error. The *Error* window at the bottom of the screen explains errors. One of the error messages displayed here is, "The input argument 'Location' is not defined in state 'OnRun' line number 1." That means a Location must be defined. 
14.	Define a location. In the Test project, double-click on the .gpo file. When the GPO editor windows display, select **Locations**, and in the *Values* window click on *Create New Variable*. 

Step	Action
15.	Name the new variable, *pickStation* and click **Accept**. 
16.	Click on click on *Create New Variable* again, name the new variable *placeStation*, and click **Accept**. 
17.	Variable data, such as the Location parameters, can be modified here. For information about how to teach the robot positions, see Editing a .gpo File file. 

Step	Action
18.	Return to the **sequence.gsq** > *Selected State*window. • Select the **Pick** statement, which will display the *Input Arguments* window. • In the *Input Arguments* window, click on the three dots to the right of *Location* to display the *Select a Reference* window. It will contain the newly created *pickStation* and *placeStation* Location variables. • In the *Select a Reference* window, select **pickStation**. • Click **Accept**. 
19.	The value will display in the *Input Arguments* > *Location* > *Value* field. 

Step	Action
20.	• Select the **Place** statement, which will display the *Input Arguments*window. • In the *Input Arguments* window, click on the three dots to the right of *Location* to display the *Select a Reference* window. • In the *Select a Reference* window, select **placeStation**. • Click **Accept**. 
21.	The Error messages indicate that a servo or digital I/O gripper initialization statement is missing. 
22.	• Since it is an initialization issue, in the *Available States* window, select **OnInitialize**. • In the *Statement Browser*, expand the *Gripper* category, and drag **Define Gripper** to the *Selected State* window. • Double-click on **Define Gripper** to display the *Input Arguments*. Click on the three dots to the right of *Profile*. 

Step	Action
23.	The *Select a Reference* window displays, and it contains a *default_profile*. Select it, and click **Accept**. 
24.	On the Available States window, click on the gear icon to generate the GPL code. 

Step	Action
25.	The code will display. 
26.	Save the project, and click **Run**.