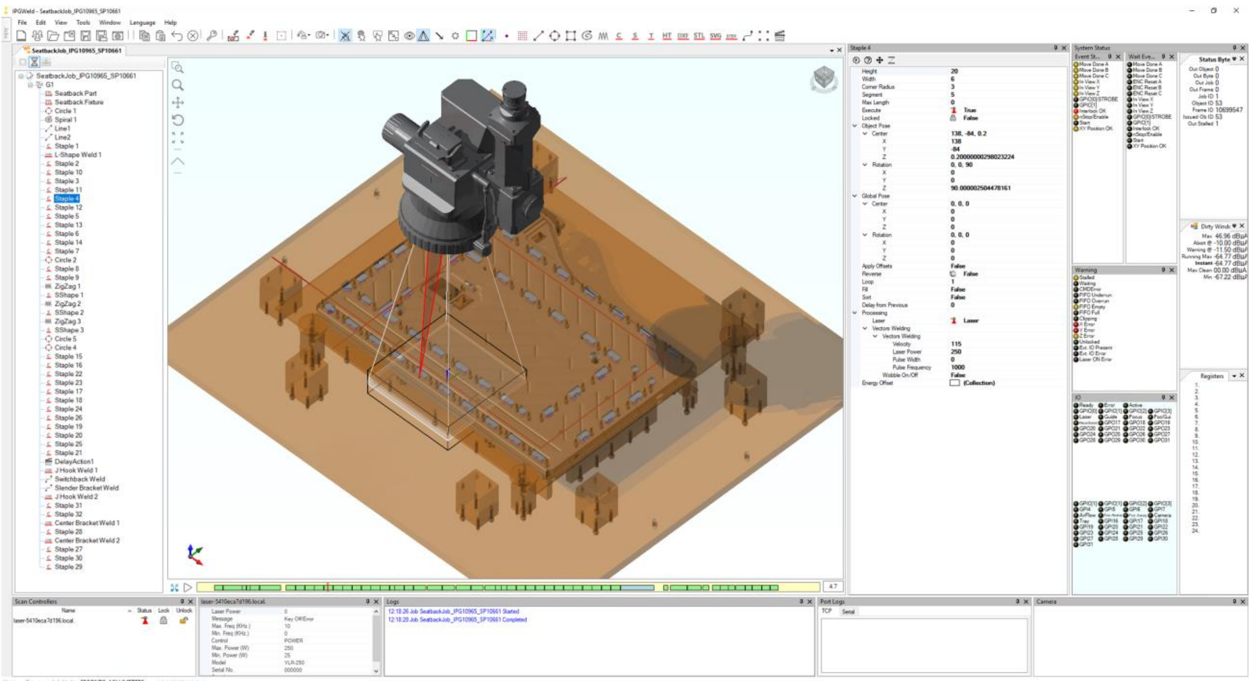
# IPGScan Software

## USER GUIDE



Part Number: DOCOXUGGUIXX0001

PUBLISHED: JULY 2022 | REVISION: B | DCO: 5126

Table of Contents

## Disclaimer Notice



This User Guide is provided "as is" and is subject to change and revision without notice. IPG believes that the information provided is accurate and reliable; however IPG makes no warranty or representation, express or implied, regarding this document, including without limitation any implied warranties of merchantability or fitness for a particular use, purpose or application, either alone or in combination with any other device, equipment, apparatus, materials or process. Users must take full responsibility for their application of any products.

Further, IPG does not assume responsibility for use of the information contained in this document or for any infringement of patents or other rights of third parties that may result from its use. IPG shall not be liable for errors in or omissions from this document or for any incidental, consequential, indirect or special damages, including without limitation, lost profits, lost production costs or similar damages, in connection with the furnishing, performance or use of this material.

IPG grants no license, directly or indirectly, under any patent or other intellectual property rights from use of the information provided herein.

IPG, IPG Photonics and the IPG Logo are registered trademarks of IPG Photonics Corporation. We have identified words that we consider as trademarks. Neither the presence nor absence of trademark identifications affects the legal status of any trademarks.

PATENT RIGHTS

This product may be patented in one or more jurisdictions. See the product for more information.

## Export, Import, and Customs Control Compliance

### (For US Products Only)

IPG is committed to complying with U.S. and foreign export, import and customs requirements. Export and re-export of lasers and other products manufactured by IPG are subject to U.S. and foreign laws and regulations, including the US Export Administration Regulations administered by the Department of Commerce, Bureau of Industry and Security. The applicable restrictions vary depending on the specific product involved, intended application, the product destination and the intended user. In some cases, an individual validated export license is required from the US Department of Commerce prior to resale or re-export of certain products.

You are ultimately responsible for exporting any IPG product in accordance with the Export Administration Regulations and the U.S. Customs and Border Protection Regulations. IPG recommends that you obtain your own legal advice when attempting to export. All export and custom classifications and information provided by IPG is subject to change without notice. IPG makes no representation as to the accuracy or reliability of the classification information provided. The stated classification only applies to equipment as it left the IPG factory. Any modifications or changes after leaving the IPG facility will be your responsibility to obtain further classifications. IPG is in no way responsible for any damages whether direct, consequential, incidental, or otherwise, suffered by you as a result of using or relying upon such classifications, groups, or symbols for any purpose whatsoever.

Information relating to U.S. export rules and regulations can be found at the U.S. Bureau of Industry and Security Website. Information related to U.S. Customs and Border Protection can be found at the U.S. Customs Website.

# 1 Overview

IPGScan and the Scan Controller seamlessly interface to IPG lasers. With the addition of the application software, IPG is able to offer a complete remote solution to Integrators and OEMs.

## 1.1 Computer Requirements

In order to operate any IPG Scanner with a Scan Controller, a computer must be connected and running IPGScan at all times.

Table 1 details minimum computer specifications that must be met for use with IPGScan software.

*Table 1 Computer Specifications*

| **Operating System** | Windows 10, Professional or Enterprise \| 64-bit |
|---|---|
| **CPU** | - 6 cores<br>- 3.60 GHz Operating Frequency<br>- CPUmark (23323 MT / 2256 ST) |
| **RAM** | - 32 GB DDR4-2400 ECC |
| **Hard Drive** | - 250 GB SSD |
| **Connections** | - Ethernet Ports<br>○ One is required for connecting to the Scan Controller<br>○ (Robotic OTF Processing Only) One for robot communications<br>○ (Optional) One for Remote API (TCP/IP) functionality with IPGScan<br>○ (Optional) One for connecting to laser software (i.e. LaserNet)<br>○ (Optional) One for an Ethernet camera |

While IPG does not supply a computer with the purchase of a scanner, the computer that is supplied with the purchase of an LDD system is capable of running IPGScan. Please refer to the appropriate LDD documentation or consult with an LDD Product Specialist for details concerning the LDD computers.

# 2 Software Installation

## 2.1 Software Installation Overview

IPG Scanning software consists of a number of different software suites. These software suites consist of the following:

1. IPGScan
    a. IPGScan is the user facing software for programming the scanning system. This software is what users will interface with 95% of the time when working with a scanner.
2. ScanPack
    a. While users will not interface with a ScanPack graphical user interface (GUI), this is an essential software component for the scanners to function properly. ScanPack takes the users program information from IPGScan and converts this information into something that is actionable by the scanner. Ultimately, users simply need to install ScanPack and then can forget that it exists.
3. The Scan Controller Utility
    a. The Scan Controller Utility is primarily used for setup of the scanner when a system is initially being setup. Some of the primary functions that users will perform using the Scan Controller Utility include, changing laser specification files, setting IP address settings, or backing up the scanner. Additional scanner setup functions exist but will rarely be utilized once a process is setup.
4. Bug Reporting Utility
    a. Unfortunately, no software is perfect. While IPG strives to develop robust and reliable software, testing does not always catch every bug prior to a software release. The Bug Reporting Utility provides a means for users to report these bugs to IPG for further improvement of the software.

## 2.2 Software Download

To download the appropriate scanning software, visit software.ipgphotonics.com. Once at the software website, navigate to the following location:

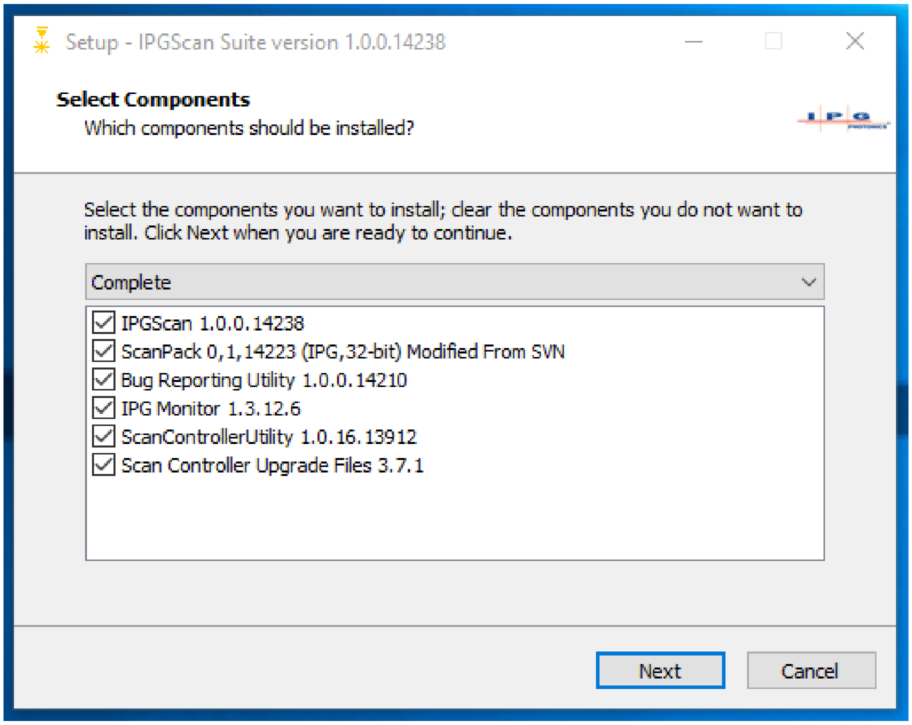
- ScannerSoftware → IPGScan

Within the IPGScan folder is the software suite installer that can be downloaded for the appropriate scanning software.

## 2.3 Software Suite Installer

Once the software is downloaded from the software website, users can run the software suite executable. Once run, users will be prompted to check which software they would like to have installed (see Figure 2-1). In general, it is best to just install all software packages.

*Figure 2-1 IPGScan Software Suite Installer*




If earlier versions of the software were installed prior to running the software suite installer, the installer will handle uninstalling the software prior to installing the new software. This means that users do not need to uninstall software from the Control Panel prior to installing the latest software.

**IMPORTANT**

**If users need to install an older version of software from what is currently installed on the computer then it is necessary to uninstall the current software in the Control Panel prior to installing the older software. Older version software suites will not uninstall new versions of software automatically.**

Once the appropriate software is selected, clicking "Next" will step users through the appropriate procedure to install all selected software packages.

### 2.3.1 IPGScan Installer

When installing IPGScan, users will be prompted with the choices to bypass user security, start when computer starts, and allow multiple instances (see Figure 2-2). These selections can be summarized in short:

- Bypass User Security
    a. Checking this makes it so users do not need to setup IPGScan security in order to use the software. It is often recommended that users bypass user security and set it up once they are finished with developing their process.
- Start When Computer Starts
    a. Checking this will cause IPGScan to startup when the computer starts.
- Allow Multiple Instances

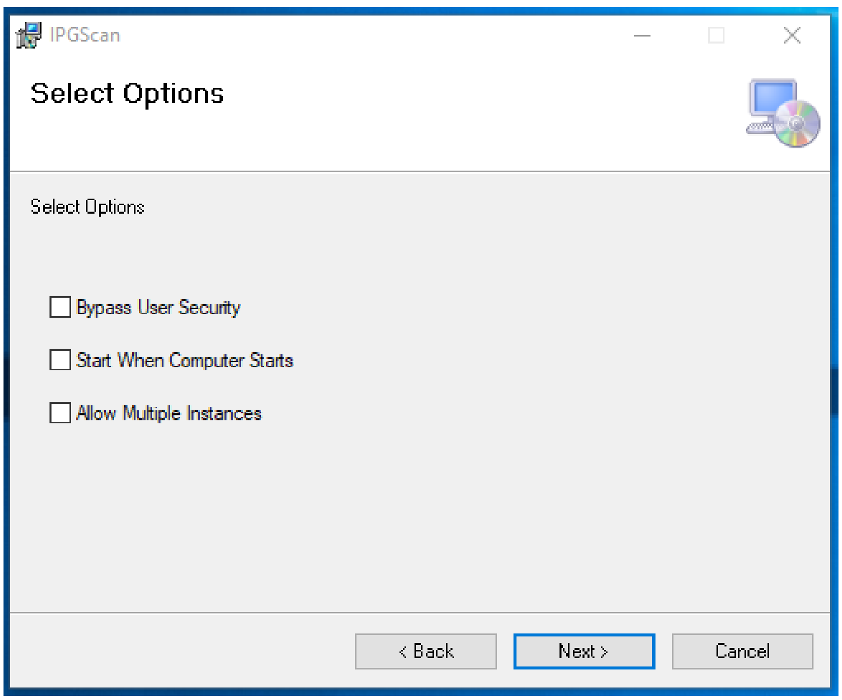
a. Checking multiple instances will allow users to open multiple IPGScan interfaces at one time. Given that only one scanner can be connected to a single instance of IPGScan at a given time, if users want to run multiple scanners from one computer at the same time, then opening multiple instances of IPGScan is required.

**The first instance of IPGScan to be opened is considered the primary instance. This will contain all of the normal IPGScan options. Any instances opened after the primary instance will have limited options available.**

**IMPORTANT**

**Opening multiple instances of IPGScan has the potential to have impacts on computer performance. Be sure to evaluate if running multiple instances of IPGScan will provide sufficient results on a given PC prior to implementing in production.**

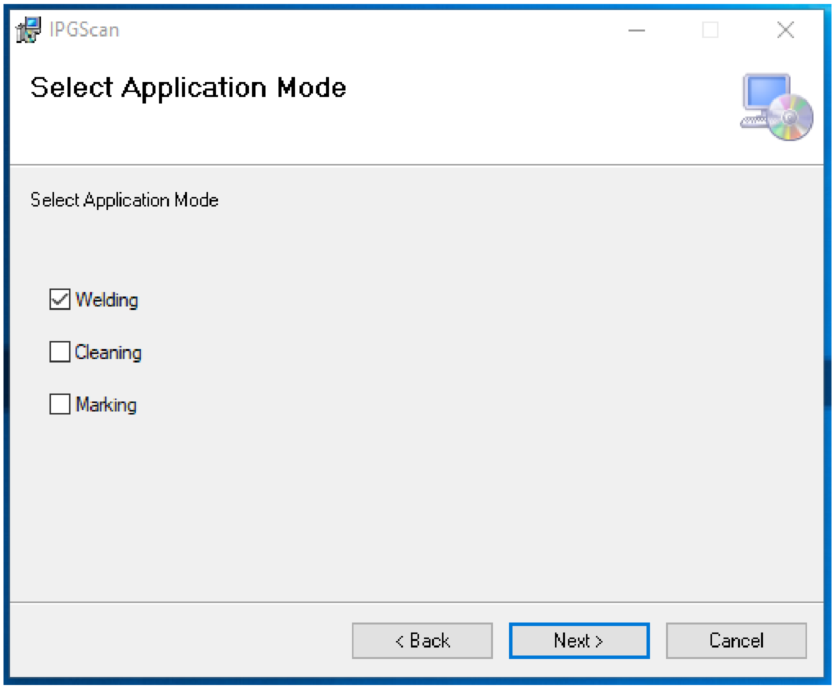
*Figure 2-2 IPGScan Installer Options*




Users will also be prompted to select which application modes are desired (see Figure 2-3). This selection should be based on what job types users will be expecting to work with for their given application. These can be summarized as follows:

- Welding
  - Most commonly used for welding applications that utilize YLS, YLR, and AMB lasers.
- Cleaning
  - Most commonly used for cleaning applications that utilize pulsed (YLPN lasers).
- Marking
  - Most commonly used for marking applications that utilize Integrated Markers.

*Figure 2-3 IPGScan Installer Application Modes*




All application modes can be selected. Users will have a desktop shortcut for each selected application mode. Finally, even if only one application mode is selected, users can still change the application type for a given IPGScan job even if the others are not selected during installation.

### 2.3.2 Additional Software Package Installers

For software packages such as ScanPack, The Scan Controller Utility, and the Bug Reporting Utility, the software suite installer will guide users through installation automatically.

# 3 System Security

## 3.1 Overview

When installing IPGScan, users have the option to "Bypass User Security." When this option is checked during installation, security is not utilized in IPGScan. If the "Bypass User Security" option is not checked when IPGScan is installed, security is utilized in IPGScan and users will be required to provide login credentials for feature permissions.

IPGScan offers the ability for Users and Groups to have different levels of access. The primary user has the ability to choose what privileges each group has, whether Operator, Technician, Supervisor, or Engineer. A Windows user with administrative rights is required to create and assign users to groups.

## 3.2 Security Settings in IPGScan

There are four security levels in IPGScan. Listed in order from highest to lowest below, each security level can perform operations specified at its level and levels below.

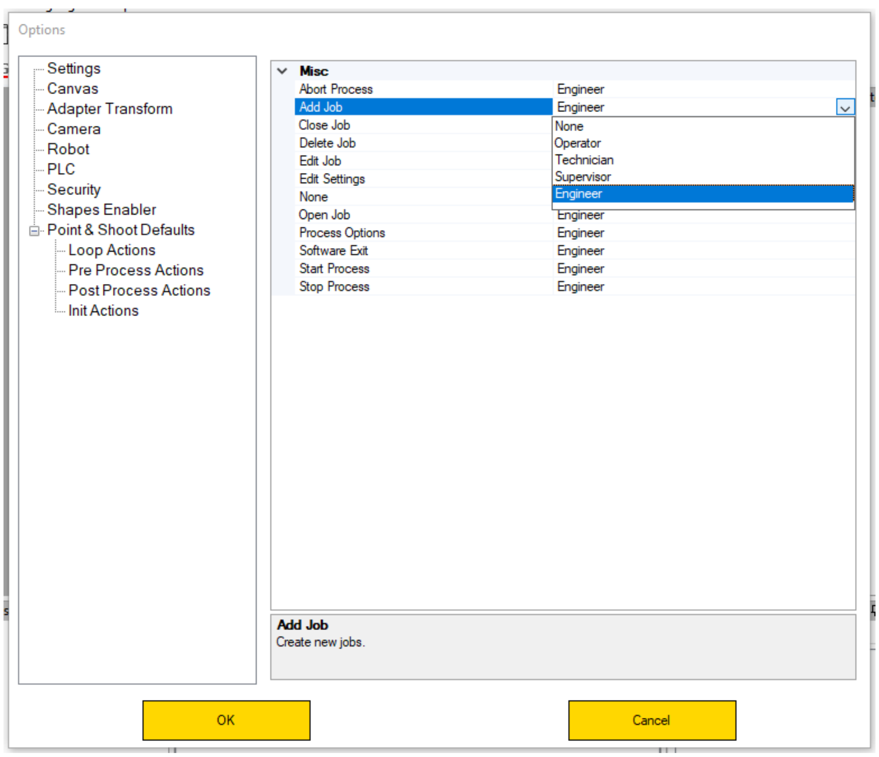
- Engineer
- Supervisor
- Technician
- Operator

For example, if an operation is set to the level of Supervisor, only users who are Supervisors or Engineers could do the operation.

By default, all settings are set to Engineer level. To adjust the security levels, use the following process.

1. Open IPGScan.
2. Click "View."
3. Click "Options."
4. Click "Security."
    a. In this window, users can define the permissions level that can perform given software functionality. See Figure 3-1.

*Figure 3-1 IPGScan Security Options*



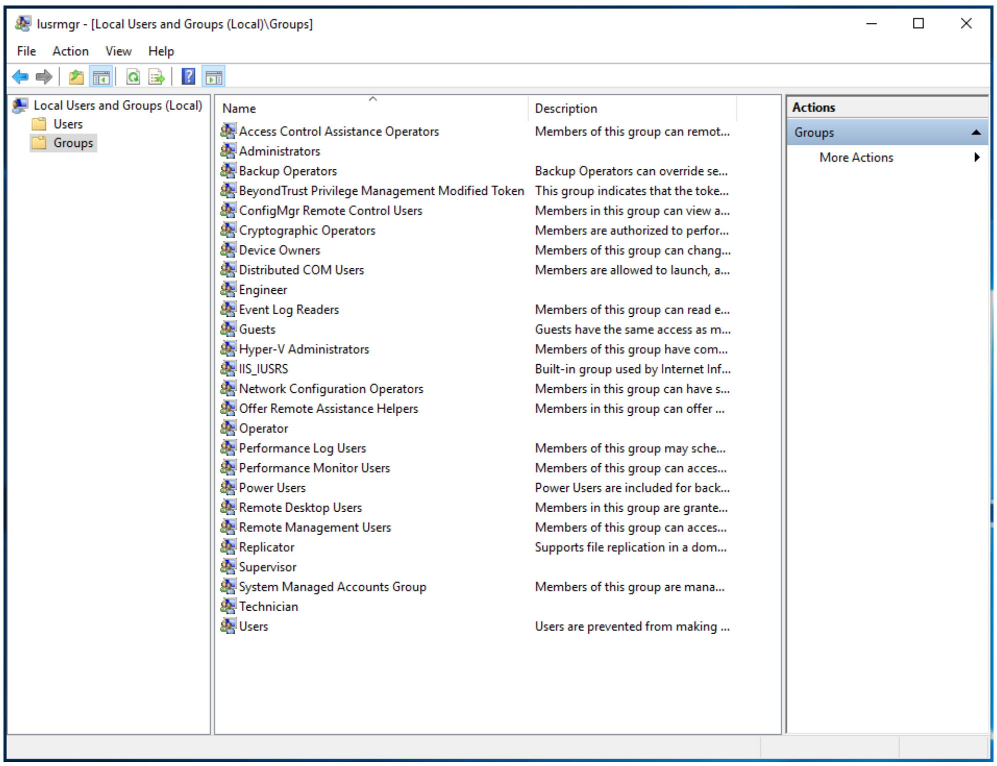
## 3.3 Assigning Users to Groups

If users do not "Bypass User Security" when installing IPGScan, it is necessary to assign "Users" to given login levels, also known as "Groups". Users must then sign into IPGScan prior to being able to utilize the software. This provides a means for users to setup permissions to be able to limit certain individuals to a given level of functionality in the software.

Users can be assigned to Groups using the following procedure.

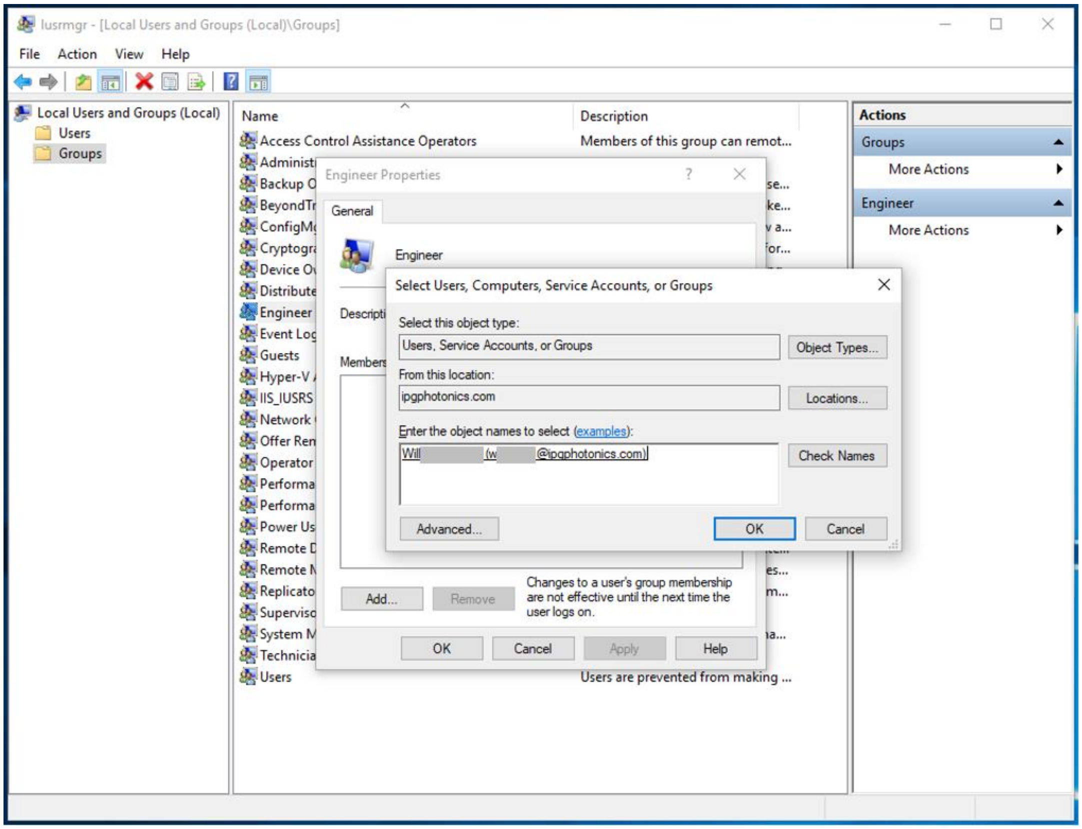
1. In the Run bar, type "lusrmgr.msc".
2. Run "lusrmgr.msc as an administrator.
3. With the Local Users and Groups window open, open the "Groups" folder (see Figure 3-2).

*Figure 3-2 Local Users and Groups Window*




4. Double click a desired security level (Engineer, Supervisor, Technician, or Operator) to add a user to.
5. In the pop-up window for the Group properties, click "Add."
6. In the "Select Users, Computers, Service Accounts, or Groups" window, users can now be added to the group (see Figure 3-3).
   a. If the computer is connected to a network domain, then user login domains can be used, otherwise, users will need to be created locally on the computer. Additional detail on creating users locally on the computer can be found in section 3.3.1.

*Figure 3-3 Adding a User to a Group*



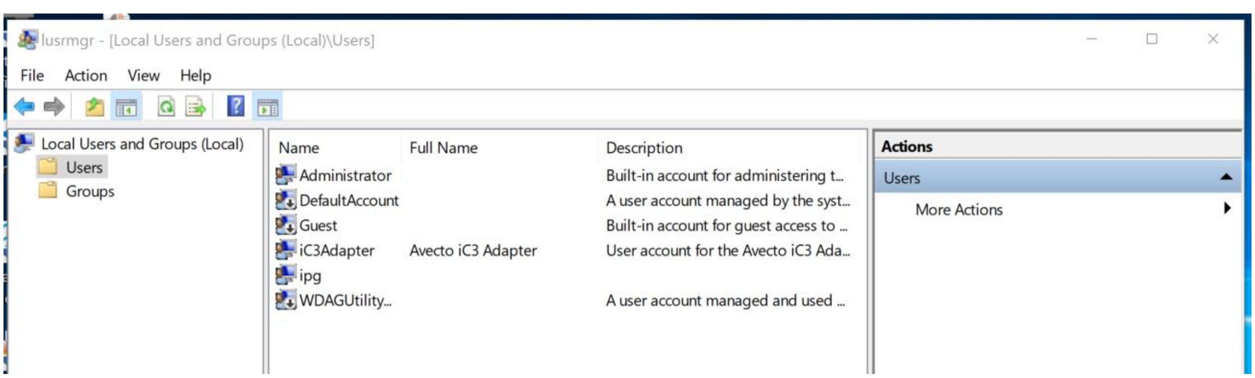

7. Click "Ok" and repeat this for as many users as desired.

### 3.3.1 Creating a Local User

Local users can be created by performing the following procedure:

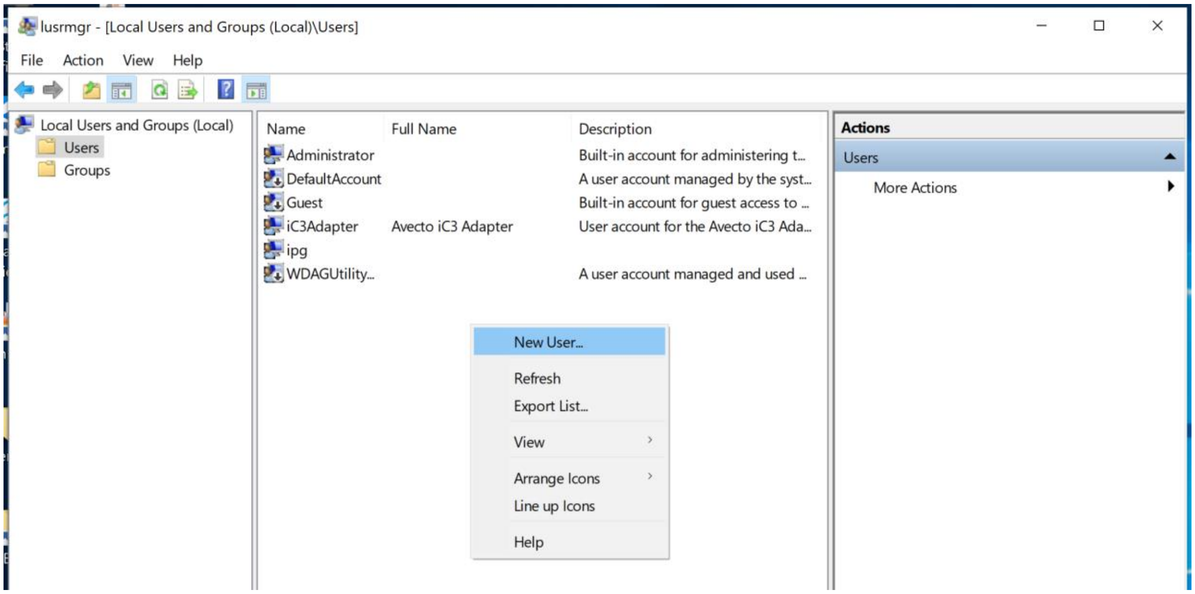
1. In the Run bar, type "lusrmgr.msc".
2. Run "lusrmgr.msc as an administrator.
3. With the Local Users and Groups window open, open the "Users" folder (see Figure 3-4).

*Figure 3-4 Local Users Folder*



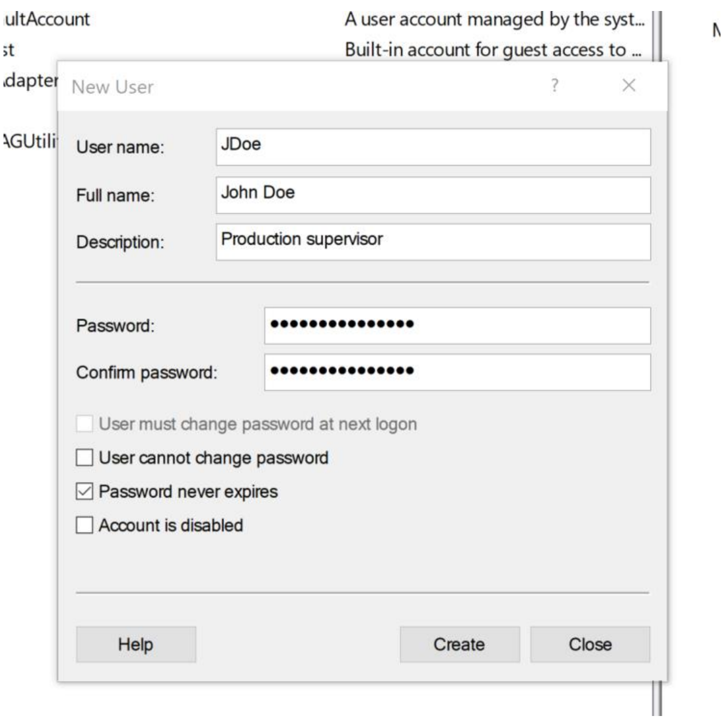

4. Right click and select "New User…" (See Figure 3-5).

*Figure 3-5 Opening the New User Window*




5. Enter the user credentials as desired. See Figure 3-6 as an example.
    a. Please note, password requirements are based on Network requirements or computer requirements. IPGScan does not dictate the password requirements for creating local users.

*Figure 3-6 Creating a Local User*




6. Click “Create” to finish creating the local user. This user can now be assigned to a Group.

# 4 IPGScan Basic Operation

IPGScan is a powerful and versatile software package that allows users to develop a scanning process with IPG scanners or integrated markers. The software has job types that are specific to Welding, Cleaning, or Marking applications. Each application is tailored to the lasers that are commonly utilized for those applications, thus, optimizing the parameters available to users for their given application. Finally, IPGScan is packed with a number of software tools allows users to create and modify standard process objects such as lines, circles, and staples while more advanced features include DXF import, dynamic text and barcode objects, and custom shapes. Ultimately, IPGScan serves as a powerful tool for users to develop a scanning process that is suited to their needs.

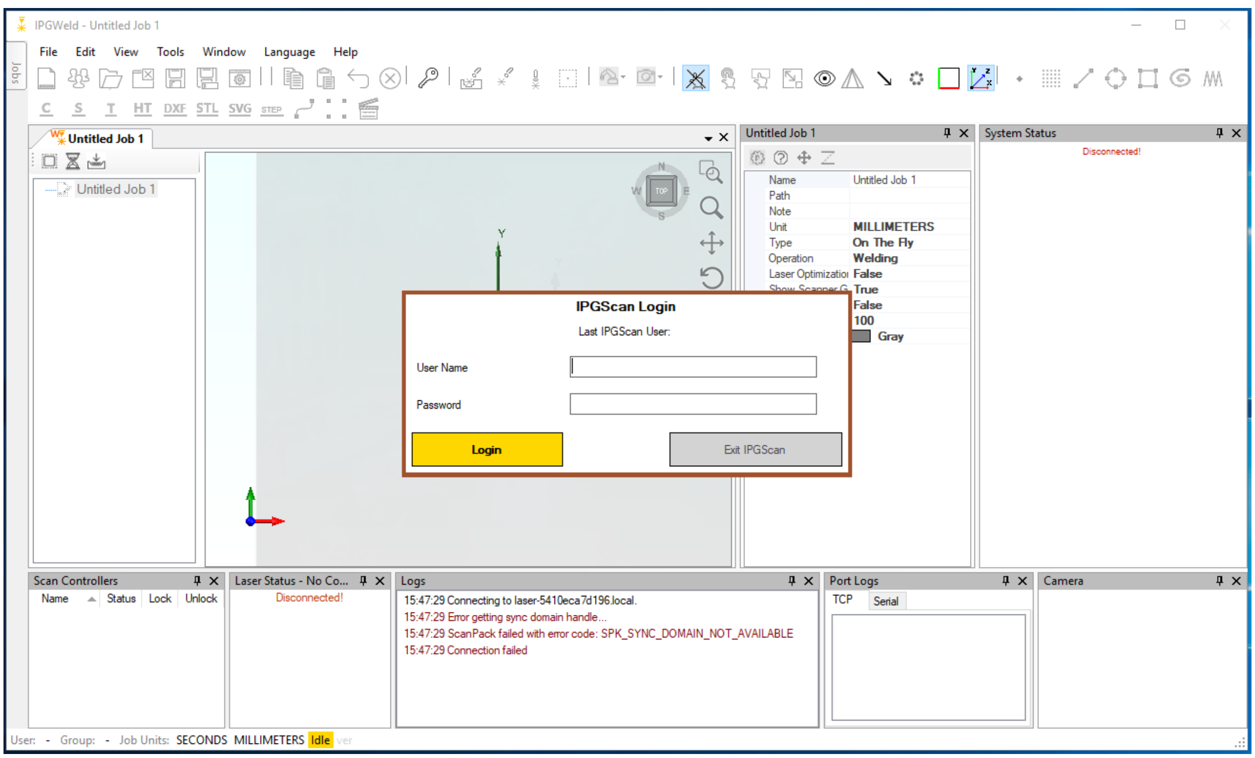
The following sections detail both setup and functionality of IPGScan.

## 4.1 Starting IPGScan

To start IPGScan:

1. Ensure “System Security”, in section 3 has been complete if “Bypass User Security” was not enabled when IPGScan was installed.
2. Go to Window’s Start Menu. Under All Programs → IPG Photonics folder.
3. Select IPGScan. The IPGScan Login window will appear, as shown in Figure 4-1.

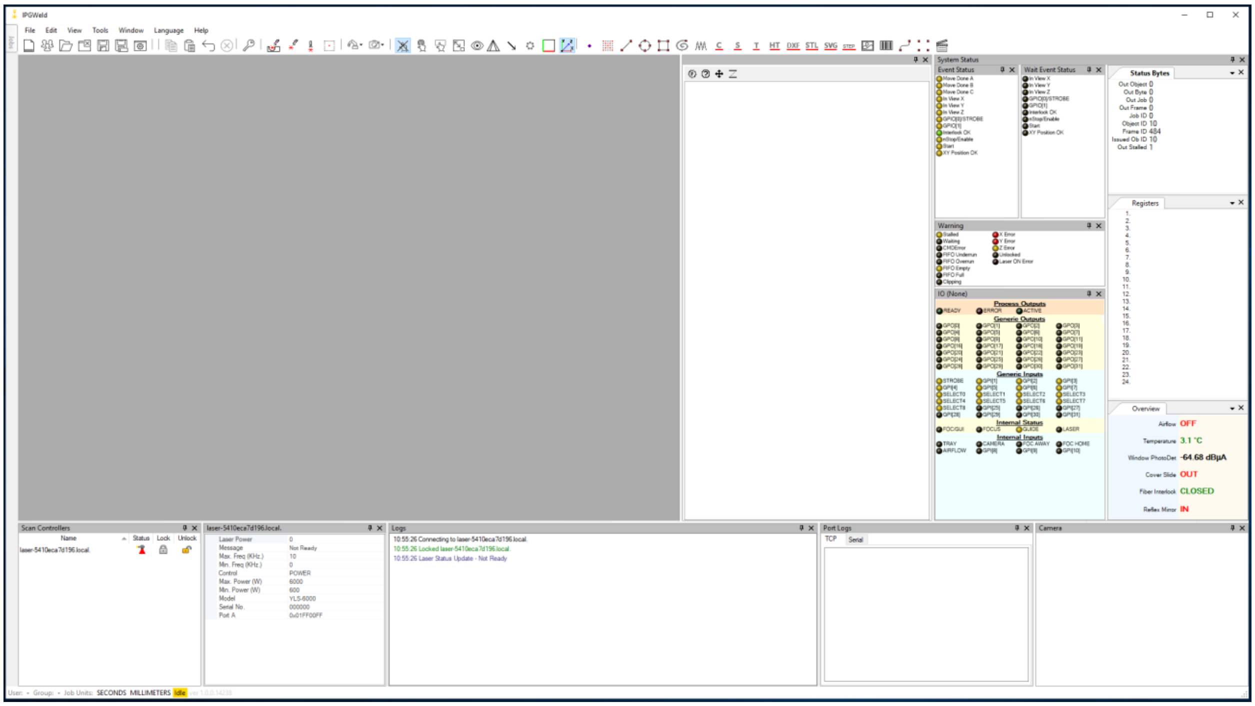
*Figure 4-1 IPGScan Login Window*




4. Enter your user name and password.
    a. If users were added from the network, enter the network domain name in the user name in the User Name box. *Example: XYZCompany-Domain\jsmith.*

   b. If users were added on the local drive of the computer, enter the local computer name in the user name in the User Name box. *Example: MyPC\jsmith*.
   c. The last user logged in will not be prompted to login again.
5. Click “Login”. The IPGScan Workspace appears, as shown in Figure 4-2.

*Figure 4-2 IPGScan Workspace*



## 4.2 IPGScan Layout

Users will find that IPGScan contains a number of windows, each of which provides setup functionality or system status information. Figure 4-3 shows the IPGScan software main window layout while Table 4-1 provides a brief description of each item.

*Figure 4-3 IPGScan Software Layout Window*

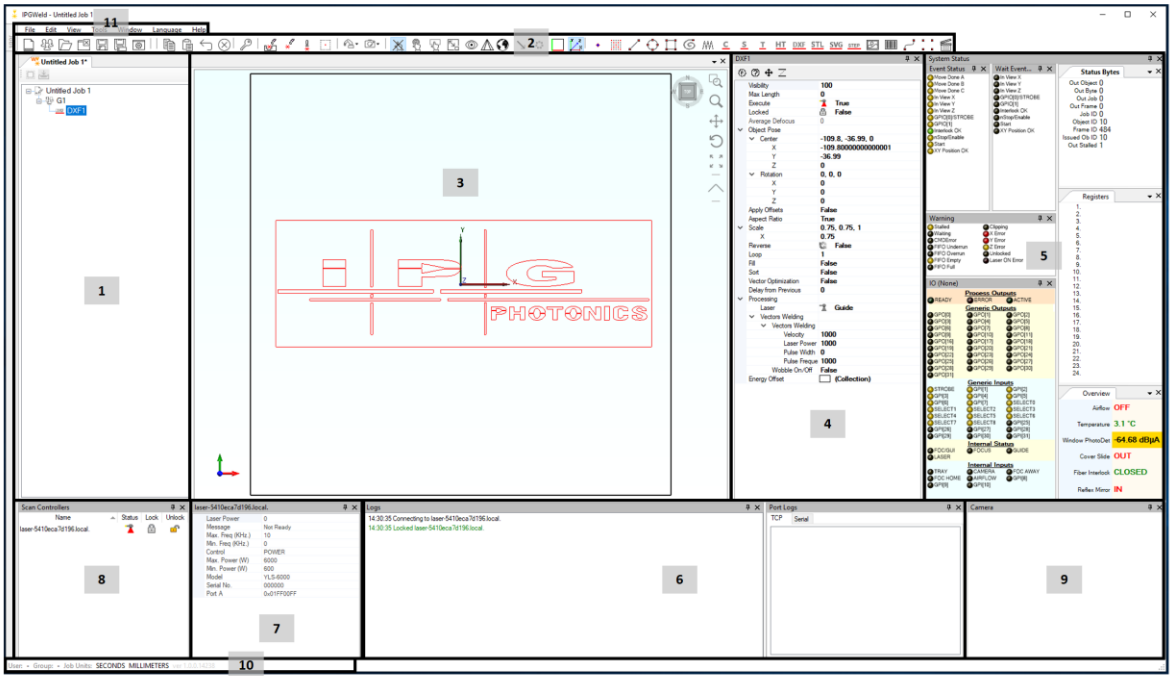



*Table 4-1 IPGScan Software Layout Items*

| Number | Description |
|---|---|
| 1 | Job Tree |
| 2 | Tool Bar |
| 3 | Canvas Field of View/Viewport |
| 4 | Data/Parameter Window |
| 5 | System Status |
| 6 | Logs and Port Logs |
| 7 | Laser System |
| 8 | Scan Controllers |
| 9 | Camera Window |
| 10 | Program Information |
| 11 | File Menu |

### 4.2.1 Resetting the IPGScan Layout

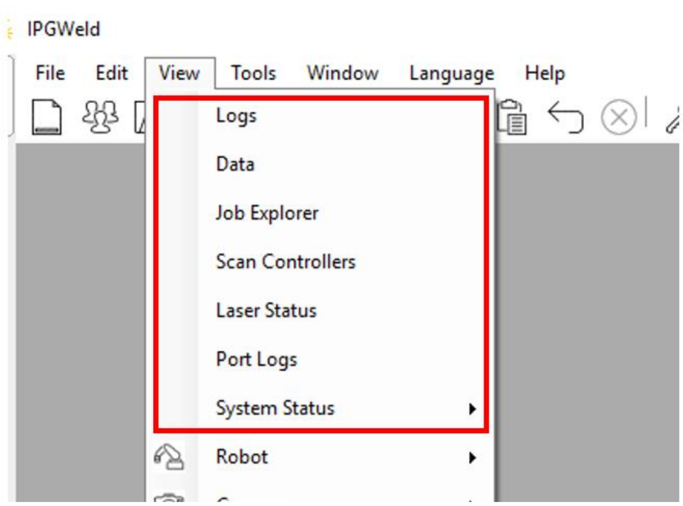
Users have the ability to modify the IPGScan layout by dragging and dropping individual windows. Windows can also be closed if users feel they do not serve a purpose. For instance, IPGScan has the ability to display an Ethernet camera video within the software. If users prefer to utilize HDMI cameras, this window then serves no purpose and users may want to close it.

Should users wish to reset the IPGScan layout back to the default configuration, the following procedure can be followed.

1. Click “View” in the Tool Bar.
2. Click “Reset Layout.”
3. Close IPGScan and relaunch the software.

Individual windows can also be reopened without resetting the IPGScan layout by selecting the appropriate window from the “View” menu. See Figure 4-4.

*Figure 4-4 Opening a Closed Window in IPGScan*



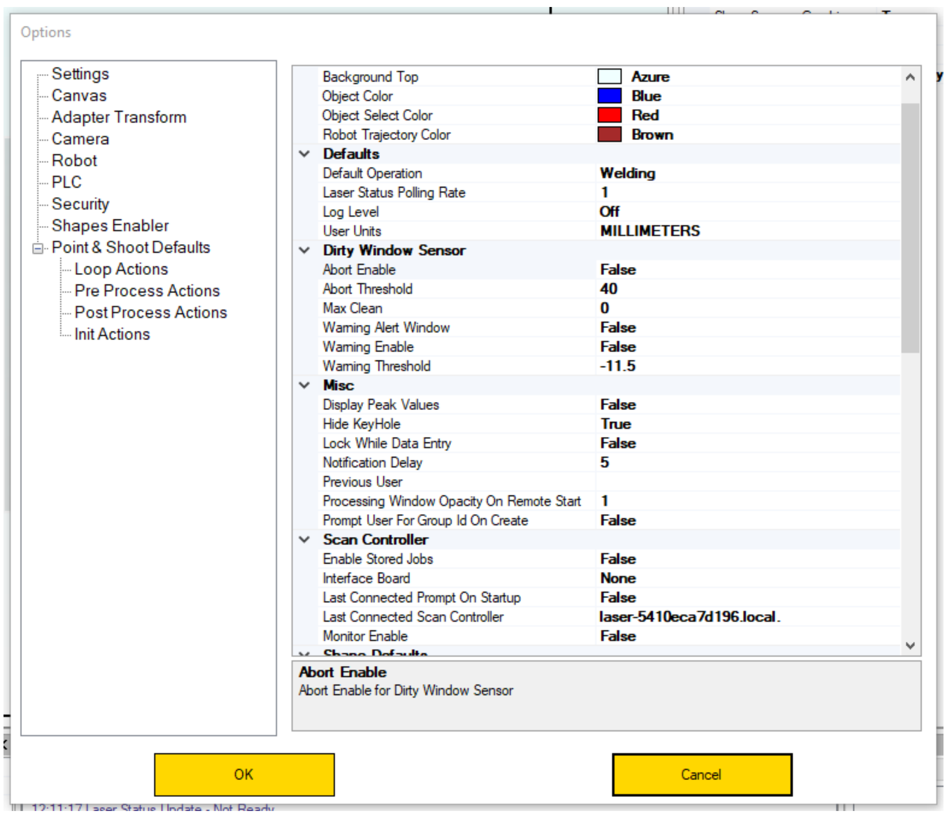
## 4.3 Options Menu

IPGScan contains numerous options that allow users to configure IPGScan as desired as well as setup a scanning process. The following sections outline the available options as well as the functions of each option.

In order to access the IPGScan Options:

1. Click on "View."
2. Click "Options" (Alt+O) to open the menu.
    a. The Options menu will appear as a pop-up window. See Figure 4-5.

*Figure 4-5 IPGScan Options Menu*




### 4.3.1 Settings

Within the Settings menu, users can further customize the IPGScan layout and setup specific software functionality. The following list includes available options:

- Colors – Allows users to change the color scheme of IPGScan.
- Defaults
    - Default Operation – Allows users to change the default job Operation type that is used when a new job is created.
    - Laser Status Polling Rate – Sets the rate of time for which the Laser Status will be updated in IPGScan.
    - User Units – Allows users to set the units of measure in IPGScan.
    - Log Level – Please refer to section 4.3.1.1.

- Dirty Window Sensor – Allows users to configure the Dirty Window Sensor for protective window contamination monitoring. Please refer to section 5 for additional details.
- Misc
    - Display Peak Values – When set True, it provides an output of peak power, velocity, and frequency in the Logs Window whenever users update processing parameters in a Marking job.
    - Enable UI Animations – Enables/disables some of the animations within the Processing Window. For large jobs or jobs with small Processing Objects, having this feature set to False may improve processing performance.
    - Hide KeyHole – Enable/disable the display of the keyhole parameter for Process Objects. Keyhole allows users to specify a delay period at the start of process objects prior to the start of galvo motion.
    - Lock While Data Entry – When enabled, the job tree will be locked when users enter new parameters for a Process Object. Users must hit ENTER or ESC to accept or cancel the new parameters prior to selecting a new Process Object in the Job Tree.
    - Previous User – Lists the previous user name that was logged into IPGScan.
    - Processing Window Opacity On Remote Start – Changes the opacity of the Processing Window when started via Remote Start.
    - Prompt User For Group ID On Create – When enabled, users will be prompted to enter a Group ID value when a new Group is created in the Job Tree.
- Scan Controller
    - Enable AMB – Enables/disables the display of Ring and Core power control for Process Objects for AMB Laser use. Please note that an appropriate AMB laser specification file is still required.
    - Interface Board – Allows users to select the External Interface Board that is being utilized in the process. This will tailor the display of the IPGScan IO for the specific board.
    - Last Connected Prompt On Startup – When true, users will be prompted to connect to the previously connected Scan Controller when IPGScan is launched.
    - Last Connected Scan Controller – The name of the Scan Controller that was last connected to IPGScan.
    - Monitor Enable – Enables signal monitoring functionality in IPGScan. This must be set to true for use of Laser On Monitor Functionality.
- Shape Defaults – Allows users to set default parameters for newly created Process Objects within an IPGScan job.
- TCP/IP
    - Actions Port – Where users can specify a desired port number for Action Controls of Load Register – Ethernet type.
    - Case Preserving TCPIP Communication – Enable/disable case preservation for TCP/IP communication.
    - Command Port – Where users can specify a desired port number for Remote API use.
    - Encoding – Encoding selection for TCP/IP communication.
    - Remote Mode Alert – Enables/disables a pop-up Window that locks IPGScan during Remote API use.

  - TCP Ready Message – Enables/disables a ready message when a TCP/IP connection is established using the Remote API. The message is, "IPGScan TCP Command Interface is ready."
- Viewport – Settings that allow the user to customize the display of a grid layout within the IPGScan Viewport window. Additional tools such as a ruler and origin point can be enabled/disabled.

#### *4.3.1.1 Log Level*

The "Log Level" setting determines the type of messages which IPGScan will write to the log. A higher log level corresponds to fewer and less frequent log messages. Logs are written to "C:\IPGP\IPGScan\Logs." In ascending verbosity, the log levels are:

- Off (least logging)
- Critical
- Error
- Warning
- Information
- Verbose
- Activity Tracing (most logging)

Because logging adds additional CPU load, it is recommended to keep the "Log Level" set to "Off" when it is determined that the system is operating as desired. This is done so that scanner processing will not be affected. If errors begin to occur, turn "Log Level" to a desired level and resume processing; use the log to help fix the error. Once the error is rectified, turn "Log Level" back to "Off."

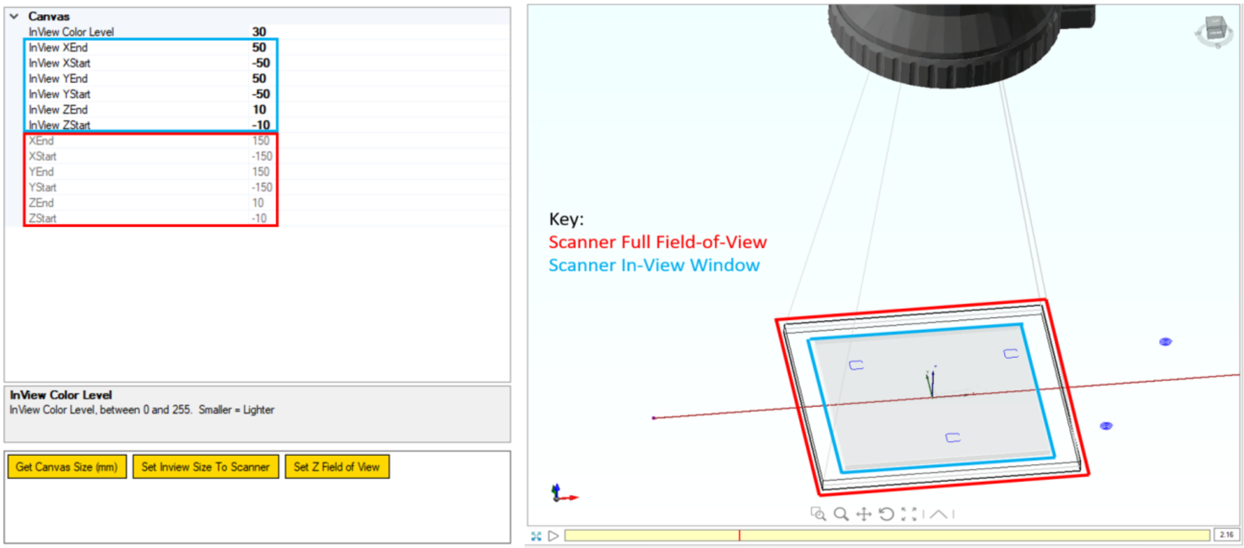
The "Log Level" setting is accessible in the Options under "Settings" and then "Defaults."

### 4.3.2 Canvas

The canvas is the area for which the scanner can process. This may also be referred to as the scanner Field of View (FOV). For the majority of applications that utilize a robot (for both Point and Shoot or On-The-Fly processes) or stationary scanner, the canvas size is typically defined by the focus lens of the scanner. For instance, a 2D High Power Scanner that utilizes a 400mm focal length lens has a 200mm x 200mm Field of View. In instances where a scanner is mounted on a gantry, the canvas size is the total area for which processing can occur. This area may be greater than the scanners allowable process area if it was used in a stationary application. Setup for gantry applications requires the use of the Stage Configuration Utility (see section "Appendix - Stage Configuration Utility").

Within the Canvas settings, users can view the scanner's full field-of-view, modify InView settings, and set Z Field of View dimensions. See Figure 4-6.

*Figure 4-6 Canvas Settings*




The following details each available parameter.

- InView Color Level
    - For Robotic OTF applications only.
    - Changes the opacity of the scanner InView window display in the canvas.
- InView Dimension Parameters
    - The purpose of the InView Window is to reduce the allowable area for which processing can occur in the Full Field-of-View. This may be desired in applications where approach of angle of the beam is important.
    - For Robotic OTF applications only.
    - Users can enter the desired Start and End dimensions for X, Y, and Z dimensions for the InView Window here. The InView Window should be smaller than the scanners Full Field-of-View size. Once users enter a desired InView Window size, click the “Set InView Size To Scanner” button.
- Scanner Full Field-of-View Parameters
    - The scanners Full Field-of-View parameters, which are defined by the scanners calibration files. These are not modifiable by the user.
- Set Z Field of View
    - This allows users to modify the Z dimensions of the scanners Field-of-View. This may be desired in instances where users want to process out of focus. Additional information can be found in section “Opening a Scanners Z Tolerance (Defocus).”

#### *4.3.2.1 Opening a Scanners Z Tolerance (Defocus)*

When using a 2D scanner, if users wish to defocus the beam, the scan head either needs to be positioned closer to the work surface or further from the work surface. This ultimately changes the relative position of the focus plane in relation to the work surface. Furthermore, if IPGScan is unaware of this offset and if no scaling factor is applied to Process Objects within the IPGScan job, laser output will likely be dimensionally incorrect according to the specified parameters within the job. For this reason, adjusting the scanners FOV Z tolerance allows users to position Process Objects within IPGScan so they more accurately represent the Process Object location on the real world work surface in relation to the scan head. Ultimately, this results in the correct dimensional output of Process Objects without the user having to specify any scaling factors within the IPGScan job.

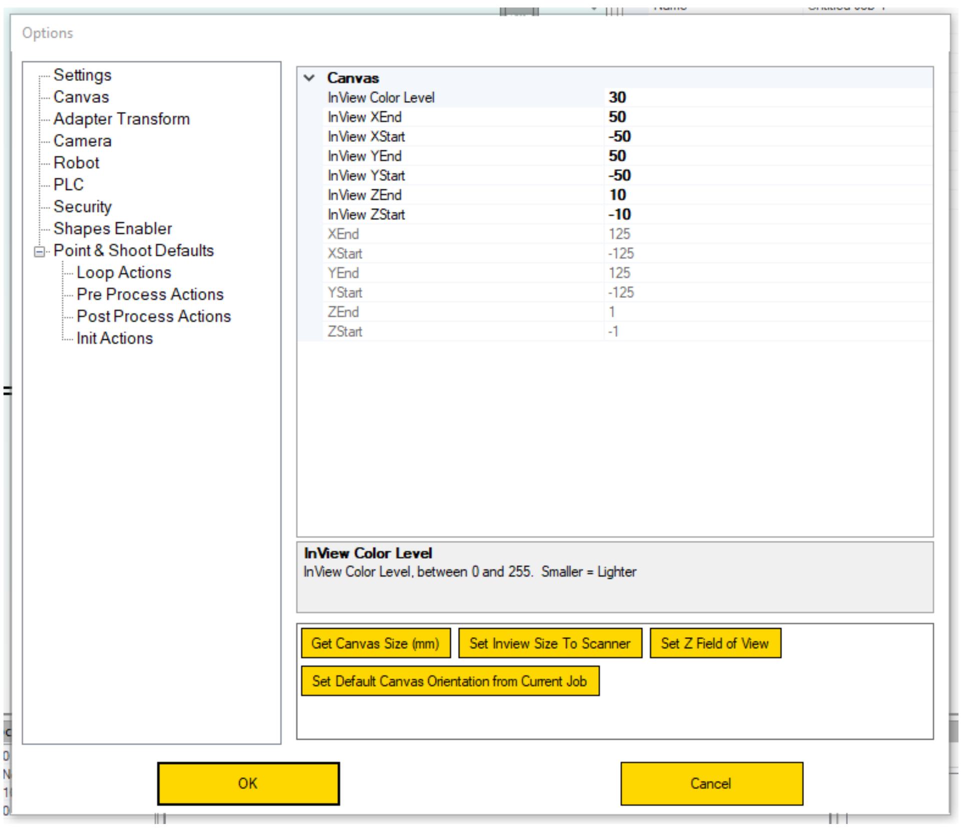
The following sections outline how users can adjust the Z dimension of the scanners FOV as well as provide examples for Point and Shoot and On-The-Fly applications.

#### 4.3.2.1.1 Set Z Field of View Procedure

The following steps detail how users can adjust a scanners FOV Z tolerance.

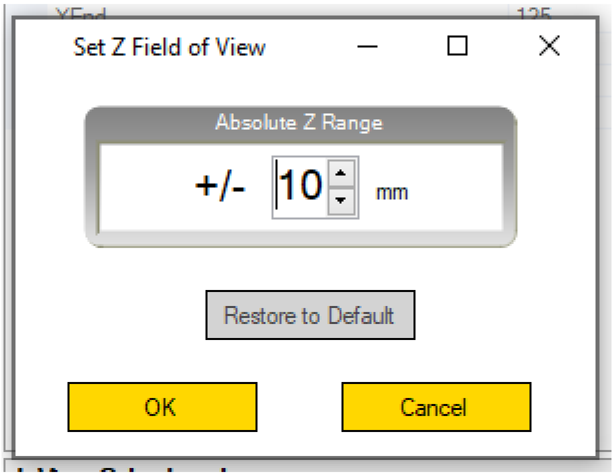
1. Open IPGScan and connect to a desired scanner.
2. Click “View”.
3. Click “Options”.
4. Click “Canvas”.
    a. Here users are presented with the scanners current Canvas (FOV) size as well as InView (Robotic OTF and Gantry Application Specific parameters) size. See Figure 4-7.
    b. For this example, notice how the default “ZEnd” and “ZStart” values are 1 and -1.

*Figure 4-7 IPGScan Option Window - Canvas Parameters*



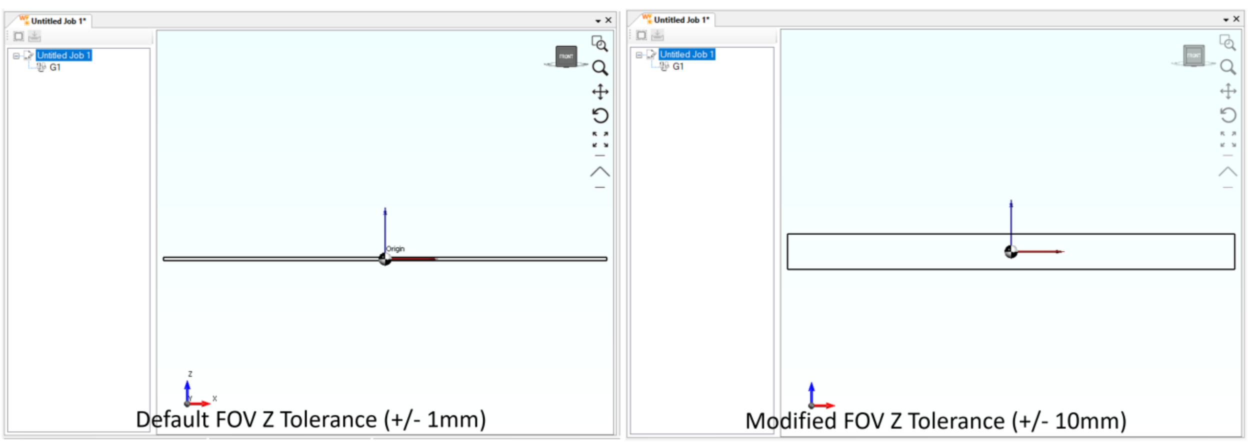

5. Click “Set Z Field of View”.
6. Adjust the “Absolute Z Range” to the desired +/- value.
    a. In this example, the Absolute Z Range is set to +/- 10mm from the nominal focus position (Z FOV dimension of 20mm total). See Figure 4-8.

*Figure 4-8 Setting the Absolute Z Range*




7. Click “Ok”.
8. Click “Ok” to acknowledge the change and then close and reopen IPGScan.
    a. Upon reopening, users should notice that the FOV Z tolerance has been modified (from a Front or Side View). See Figure 4-9.

*Figure 4-9 Change in FOV Z Tolerance*




To restore FOV Z tolerances back to the default settings, simply navigate to the “Set Z Field of View” window (as seen in Figure 4-8) and click “Restore to Default.” Acknowledge the changes, close IPGScan, and reopen IPGScan for the changes to take effect.

#### 4.3.2.1.2 Robotic On-The-Fly Example

In the case of Robotic OTF jobs, defocusing/opening up the scanner FOV may be desired for process results (i.e. weld quality) or to make it easier to pass Dryruns and setup the process.

Figure 4-10 provides an example of a robotic OTF job where no intentional defocus was implemented. In this example, the scanner has a default FOV Z tolerance of +/- 1mm. After performing a Dryrun of the process, users can select each Process Object and view the average defocus for that object. In this example, Weld 4 has an average defocus of .400mm.

*Figure 4-10 Robotic OTF Job Without Defocus*

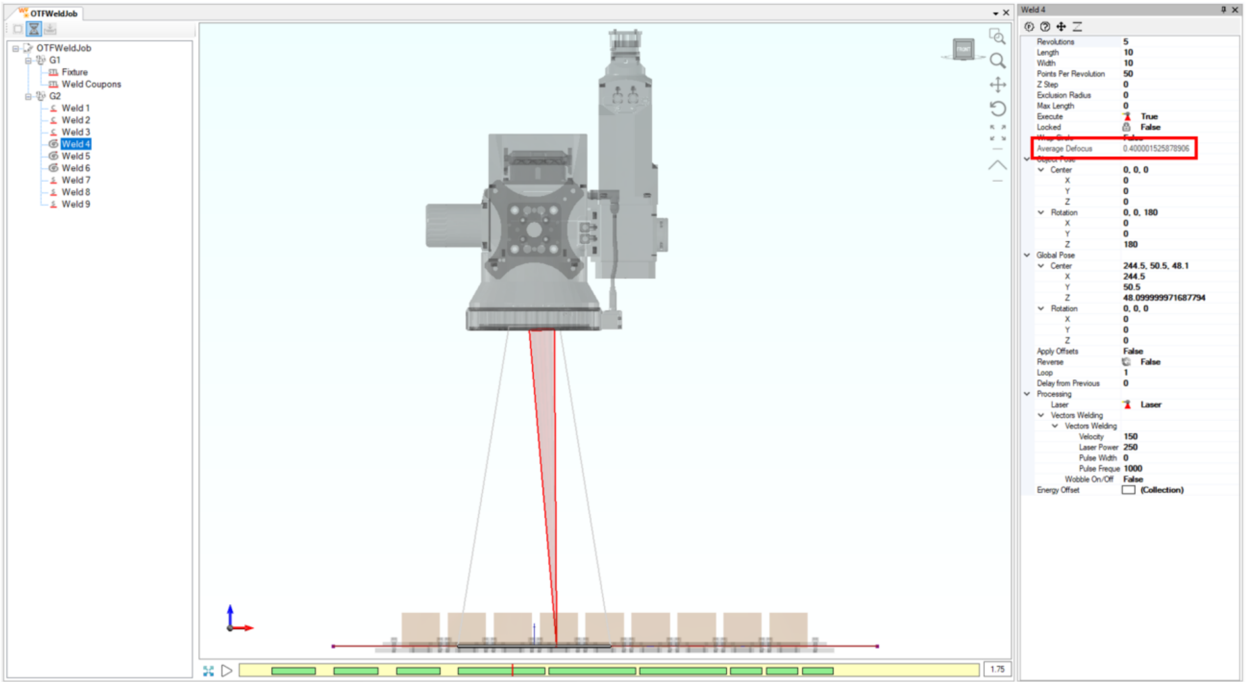
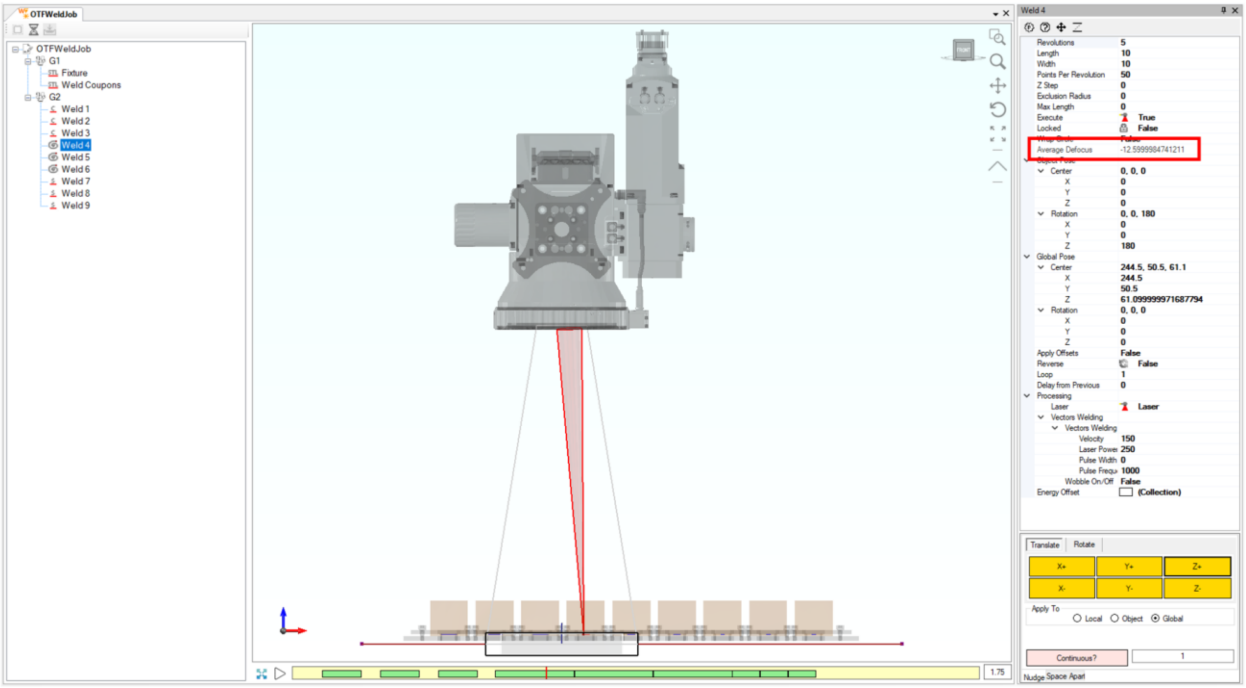



Figure 4-11 provides an example of a robotic OTF job where an intentional defocus was implemented. In this example, the scanner was configured with an “Absolute Z Range” of +/- 15mm. The In-View Z tolerance was also increased to +/- 14mm. With the Z tolerance of the scanner’s FOV increased, the user can now program the robot to position the scanner closer to the work surface or further from the work surface in order to apply an intentional defocus.

| | |
|---|---|
| **IMPORTANT** | **For Robotic OTF applications, the robot trajectory must be recaptured and loaded into the IPGScan job anytime a robot program change is made. Once a new trajectory is loaded into the job, users should Dryrun the job to ensure it completes as expected.** |

With the intentional offset applied, users can once again perform a Dryrun and select each individual object to view the average defocus. In this example, Weld 4 has an average defocus of -12.599mm. This means that the scanners nominal focus position is 12.599mm below the Process Objects program location (which should be at the work surface of the material in the real world). An average defocus value that is positive indicates that nominal focus position would be above the Process Objects program location.

*Figure 4-11 Robotic OTF Job With Defocus*

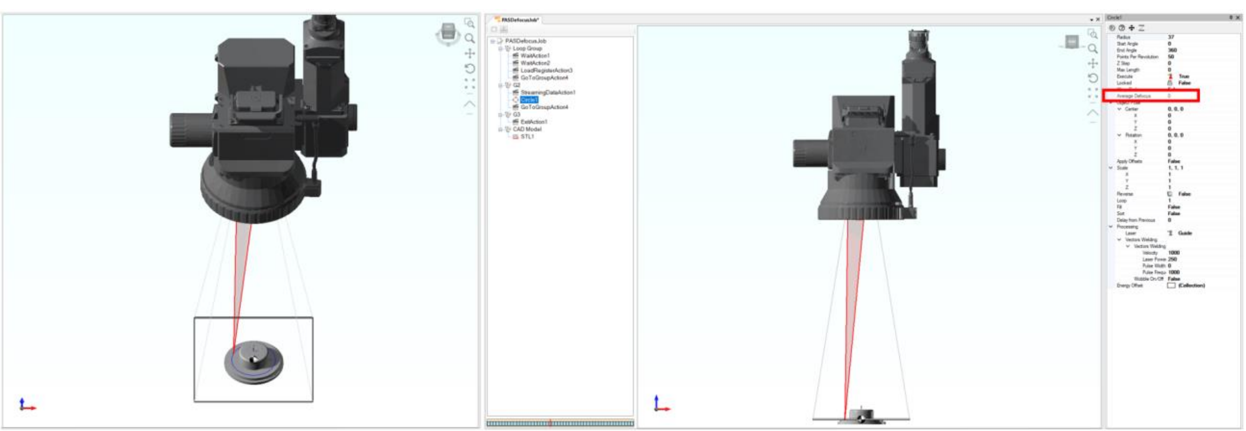



### 4.3.2.1.3 Default and Point and Shoot Job Example

Defocus can also be utilized with Default and Point and Shoot type jobs. For instance, instead of having to apply a scale or purposely offset a Process Objects size in IPGScan in order to compensate for an intentional defocus, users can utilize the defocus functionality to increase Z tolerance of the scanner. With the increased Z tolerance, users can then preserve Process Object sizing parameters, but simply offset the objects so the distance between the scan head and work surface correlate in IPGScan to how they do in the real world.

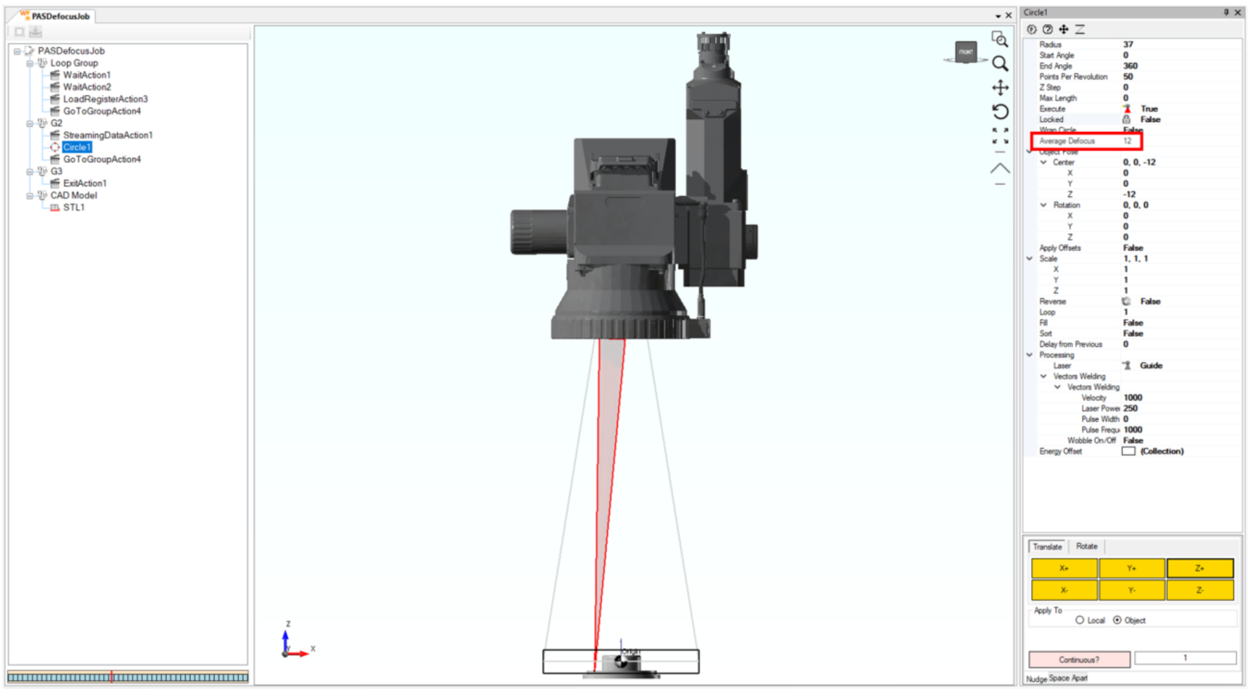
Figure 4-12 outlines and example of a Pin and Bushing part with a single weld. In this example, the scanner is configured with default Z tolerances of +/- 1mm. For this reason, the weld is positioned at the scanners nominal focus position and an average defocus of 0mm can be seen when Circle1 is selected.

*Figure 4-12 Point and Shoot Example Without Defocus*



If users wish to apply to defocus in this instance, the Z tolerance can then be increased as seen in Figure 4-13. In this scenario, the scanners "Absolute Z Range" was increased to +/- 15mm. With the increased Z tolerance, users can then position the Process Object (Circle1) so that the distance of the Process Object in relation to the scan head in IPGScan matches that of the scan head and the weld surface in the real world. When the IPGScan setup matches that of the real world setup, no scaling or intentional offsetting of Process Object parameters is required in order to compensate for the scanner to process out of focus. As seen below, Circle1 has an average defocus of 12mm, which indicates that the nominal focus position of the scanner is 12mm above the Process Object position (work surface position).

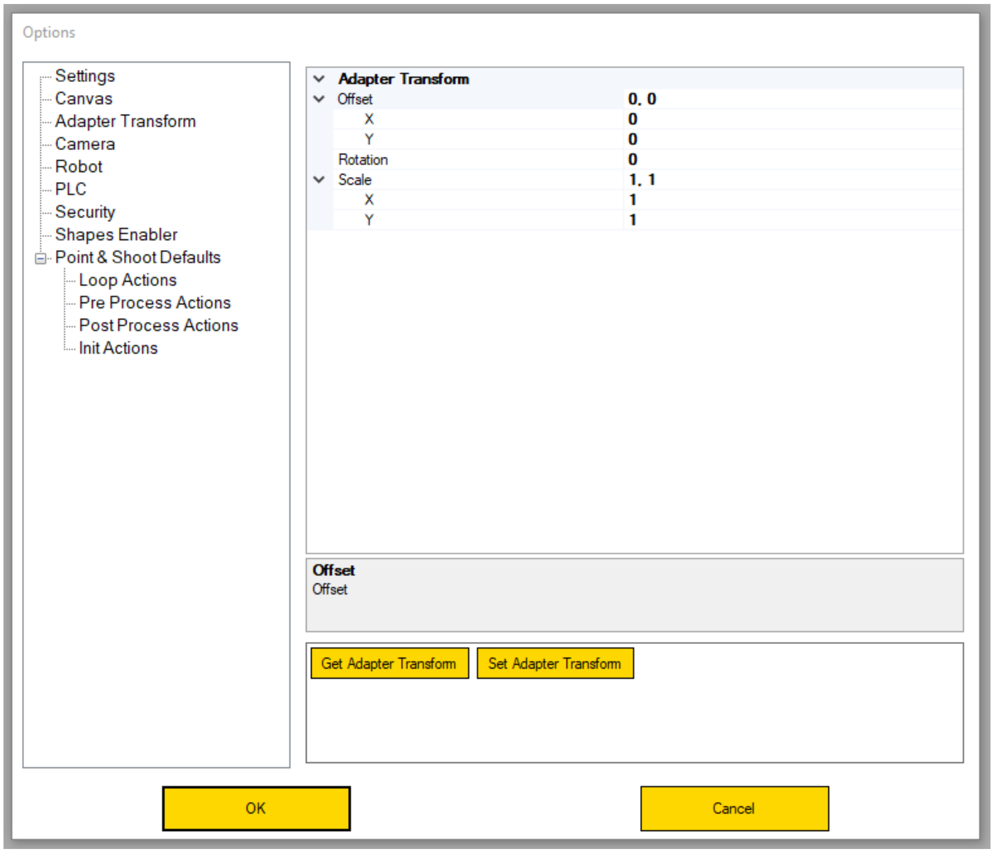
*Figure 4-13 Point and Shoot Example With Defocus*




## 4.3.3 Adapter Transform

Users have the ability to adjust the scanners field of view relative to the scan head itself using offset, rotation, and scale parameters. Clicking the "Get Adapter Transform" button will update the Adapter Transform values in the settings with any transforms that may currently be applied to the scanner. If users wish to apply a transform, simply update the values with the desired transform and click "Set Adapter Transform." Figure 4-14 displays the available Adapter Transform setting.

*Figure 4-14 Adapter Transform Settings*




### 4.3.4 Camera

IPGScan has the ability to interface with an Ethernet camera. These settings are only needed for camera integration and setup. See Section 10.1 for Sentech Ethernet Camerasetup.

### 4.3.5 Robot

The robot settings are only required for Robotic On-The-Fly. When using Point and Shoot, the user does not need to change any of these settings. See Section 12 for instructions on how to set up Robotic On-The-Fly.

### 4.3.6 Security

The security settings are where privileges can be set for given user levels. See Section 3 for details concerning security setup.

### 4.3.7 Shapes Enabler

This allows the user to select which features they would like to have displayed in the Tool Bar and Tool Menu. Table 4-2 outlines the available Objects and their corresponding default enable setting.

*Table 4-2 Shapes Enabler Settings List*

| Object | Default Enable Setting |
|---|---|
| Action Control | True |
| Barcode | False |

| | |
|---|---|
| C Shape | True |
| Circle | True |
| DXF | True |
| Hershey Text | True |
| Image | False |
| Line | True |
| Multi Shape | True |
| Point | True |
| Point Array | True |
| Points | True |
| Rectangle | True |
| S Shape | True |
| Spiral | True |
| STEP | True |
| STL | True |
| SVG | True |
| Text | True |
| Zigzag | True |
| | True |

### 4.3.8 Point & Shoot Defaults

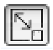
Users can adjust the default Action Controls that are generated with newly created Point & Shoot job types. Please refer to Section 11.3 for additional detail.

## 4.4 Toolbars

### 4.4.1 Mouse Resize

The mouse resize feature allows users to grab objects in the canvas and resize them by dragging the mouse.

#### *4.4.1.1 Using Mouse Resize*

1. Click the Resize button in the tool bar. 
    a. The cursor/mouse pointer should now change into a small square.
2. Click and hold on the desired object in the canvas.
3. Drag the mouse to resize the object. See Figure 4-15 Mouse Resize below.

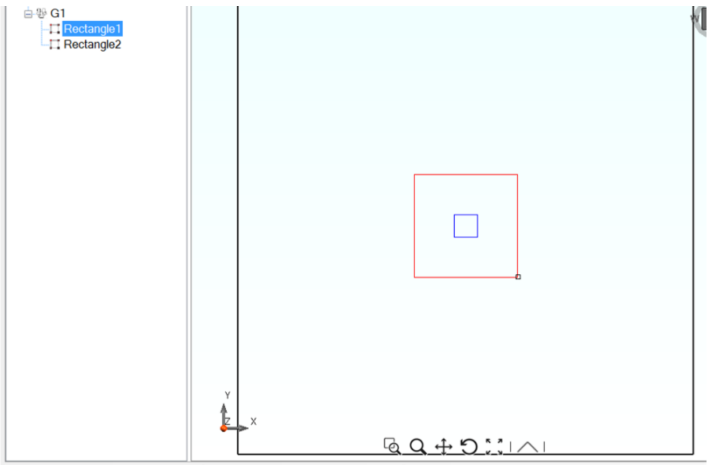



*Figure 4-15 - Mouse Resize Feature*

4. Once finished, click No Select  in the tool bar to revert the pointer back to its default state.

### 4.4.2 Park At

In the toolbar, the Park At button, , will set the target position of the Scan Head in the Scan Head's field of view. The Park At button is only accessible when either the guide beam or the focus guide beam is enabled.

Figure 4-16 shows the Park At window. The values in the X, Y, and Z boxes are the target position of the Scan Head. Changing the increment radio button will change the magnitude of the adjustment when either the buttons to the side of the boxes or the buttons under the increment radio buttons are pressed. When the "Park" button is pressed, the Scan Head is parked at the location specified in the X, Y, and Z boxes, given that it is a valid location. Pressing the "Origin" button will park the Scan Head at (0, 0, 0). Pressing the "Reset" button will reset the text boxes back to the value (0, 0, 0).

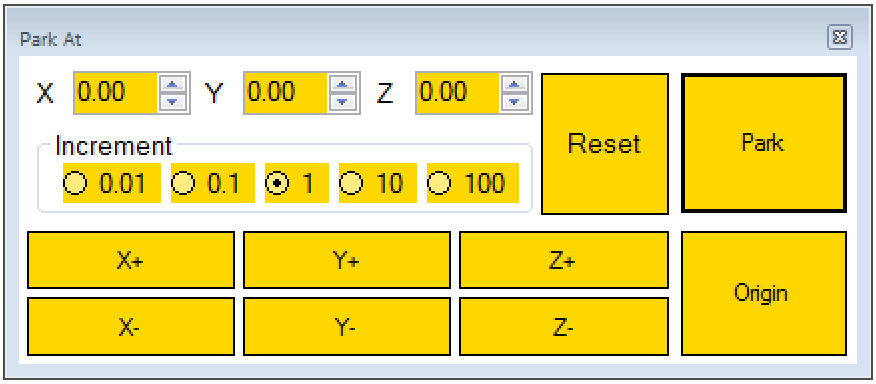



*Figure 4-16 - Park At Window*

### 4.4.3 Stage Motion

The stage motion dialog allows the user to control the position of a stage when controlled by an attached IPG Scan Controller. The stage motion dialog can be accessed under the Tools menu.

| | |
|---|---|
| **IMPORTANT** | **Users must have the "Interface Board" selection set to "Motion IO" for the Stage Motion tool to be accessible in the Tools menu when using recently releases of IPGScan.** |

Figure 4-17 shows the Stage Motion Dialog. The text boxes in the middle correspond to the destination position of the stage motion. Pressing "go" will move the stage to the position represented by all 3 textboxes. A textbox without a valid number will be treated as a 0. Pressing the home button will home the axes with a selected checkbox next to each letter. In Figure 1, no axes will be homed when the home button is pressed.

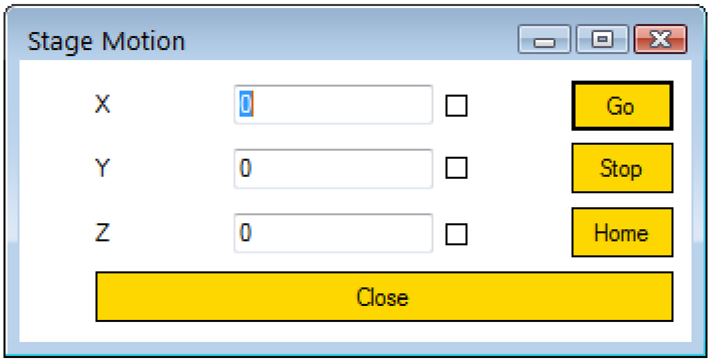



*Figure 4-17 - Stage Motion Dialog*

### 4.4.4 Rotary Configuration

The rotary configuration of the scan controller can be configured from IPGScan by selecting Tools → Rotary Configuration → Enable. This will open a dialog box where the user can enter the desired Radius and Width of their rotary configuration. Figure 4-18 shows the rotary configuration dialog. This configuration is applied for this instance of IPGScan and does not edit any of the configuration files. See section 13 and section 20 for more information about the rotary.

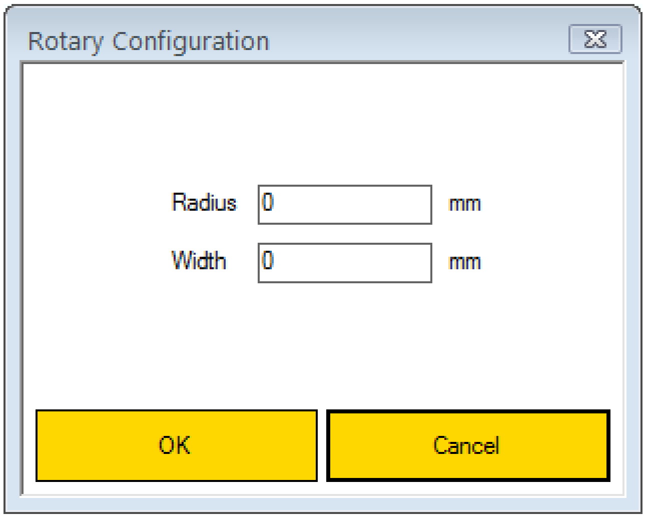



*Figure 4-18 - Rotary Configuration*

### 4.4.5 Initialize Scanner

The currently connected scan controller can be initialized by going through the dropdown menu Tools→Scanner→Initialize. The scan controller is initialized during the first lock of the scan controller during that instance of IPGScan. Initialization can also be done by restarting IPGScan. The scan controller only needs to be re-initialized if the configuration files are changed.

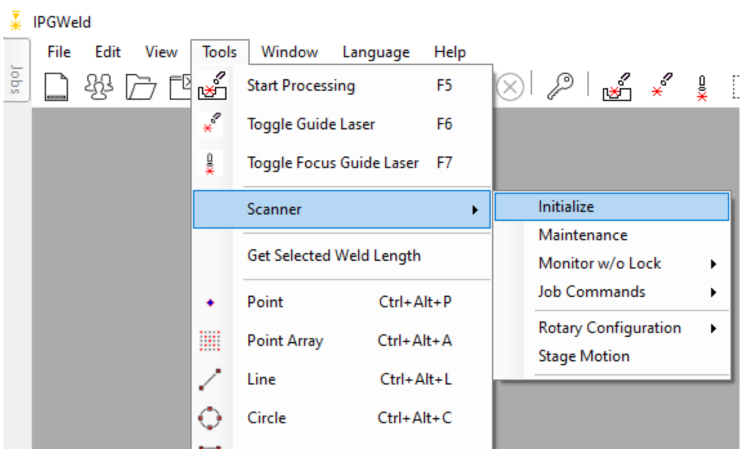



*Figure 4-19 - Initialize Scanner*

### 4.4.6 CSV and XML Export

IPGScan can export the data of a job to a CSV file or an XML file. This provides the user with a way to quickly output parameters for record keeping without having to individually check each object in the job.

#### *4.4.6.1 Exporting a Job*

1. Open IPGScan
2. Open the desired job
3. Click “File”
4. Click “Export to CSV” or “Export to XML”

a. Figure 4-20 details steps 3 and 4

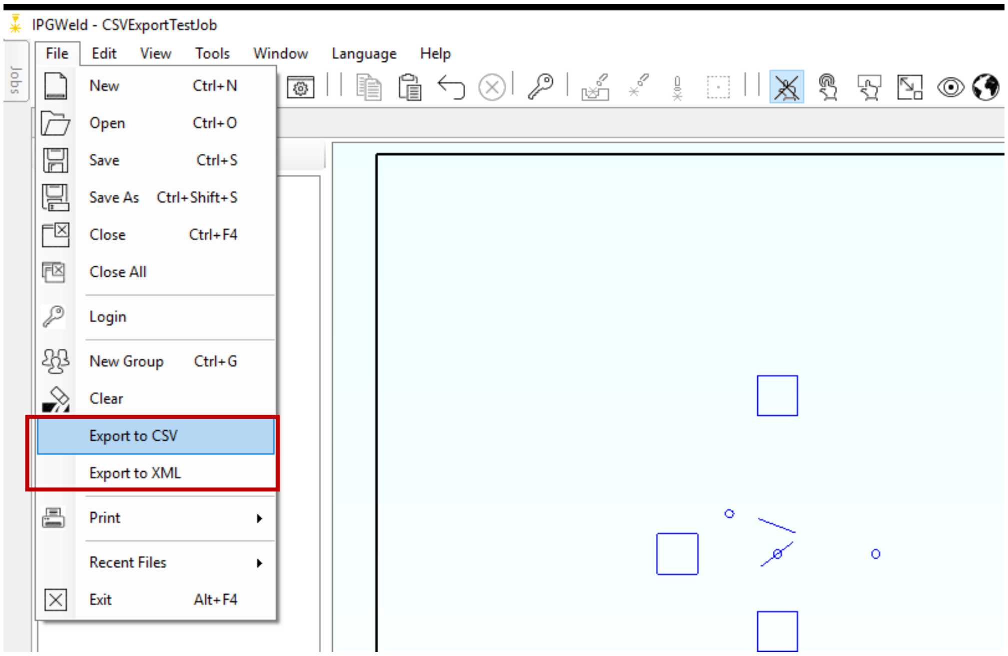
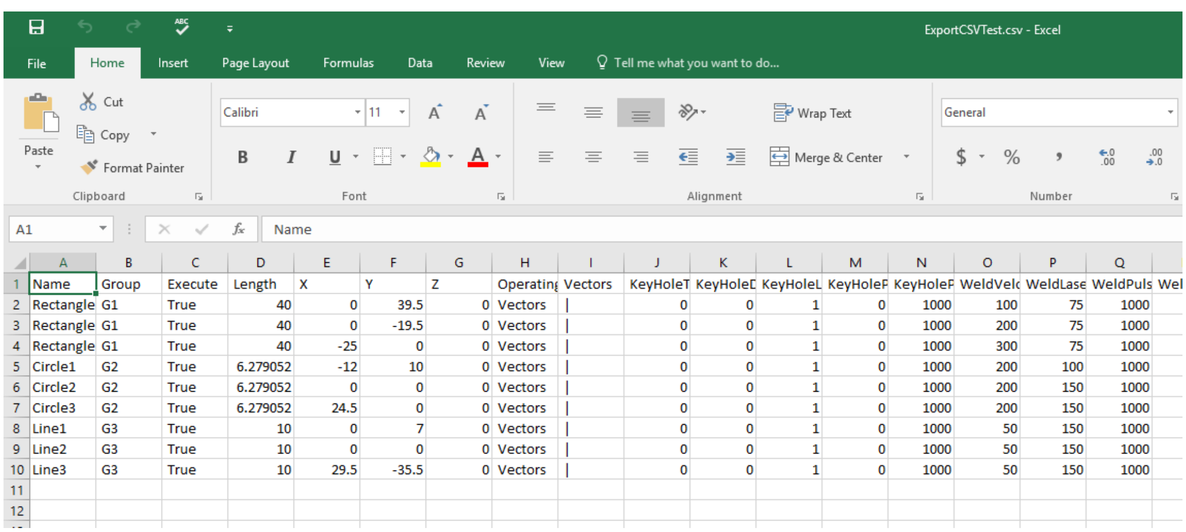



*Figure 4-20 - Export to CSV Feature*

5. Navigate to the desired save location
6. Name the file
7. Click "Save"
8. The user can now navigate to the saved file and open it for viewing
   a. Figure 4-21 and Figure 4-22 detail a portion of the example output for each file type.
   b. Please keep in mind that parameter variables are output for all IPGScan job types (IPGWeld/IPGClean/IPGMark). Users should refer to the parameters specific to the job Operation Type that they are using.




| Name | Group | Execute | Length | X | Y | Z | Operatin | Vectors | KeyHoleT | KeyHoleD | KeyHoleL | KeyHoleP | KeyHoleP | WeldVelc | WeldLase | WeldPuls | Wel |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rectangle | G1 | True | 40 | 0 | 39.5 | 0 | Vectors | \| | 0 | 0 | 1 | 0 | 1000 | 100 | 75 | 1000 | |
| Rectangle | G1 | True | 40 | 0 | -19.5 | 0 | Vectors | \| | 0 | 0 | 1 | 0 | 1000 | 200 | 75 | 1000 | |
| Rectangle | G1 | True | 40 | -25 | 0 | 0 | Vectors | \| | 0 | 0 | 1 | 0 | 1000 | 300 | 75 | 1000 | |
| Circle1 | G2 | True | 6.279052 | -12 | 10 | 0 | Vectors | \| | 0 | 0 | 1 | 0 | 1000 | 200 | 100 | 1000 | |
| Circle2 | G2 | True | 6.279052 | 0 | 0 | 0 | Vectors | \| | 0 | 0 | 1 | 0 | 1000 | 200 | 150 | 1000 | |
| Circle3 | G2 | True | 6.279052 | 24.5 | 0 | 0 | Vectors | \| | 0 | 0 | 1 | 0 | 1000 | 200 | 150 | 1000 | |
| Line1 | G3 | True | 10 | 0 | 7 | 0 | Vectors | \| | 0 | 0 | 1 | 0 | 1000 | 50 | 150 | 1000 | |
| Line2 | G3 | True | 10 | 0 | 0 | 0 | Vectors | \| | 0 | 0 | 1 | 0 | 1000 | 50 | 150 | 1000 | |
| Line3 | G3 | True | 10 | 29.5 | -35.5 | 0 | Vectors | \| | 0 | 0 | 1 | 0 | 1000 | 50 | 150 | 1000 | |

*Figure 4-21 - CSV File Example*

```xml
<Shapes>
  <WeldShapesBase xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="RectangleShape">
    <ScanPackStartTime>-1</ScanPackStartTime>
    <ScanPackEndTime>-1</ScanPackEndTime>
    <Failed>false</Failed>
    <Points>
      <Point3D>
        <X>-5</X>
        <Y>-5</Y>
        <Z>0</Z>
      </Point3D>
      <Point3D>
        <X>-5</X>
        <Y>5</Y>
        <Z>0</Z>
      </Point3D>
      <Point3D>
        <X>-5</X>
        <Y>5</Y>
        <Z>0</Z>
      </Point3D>
      <Point3D>
        <X>5</X>
        <Y>5</Y>
        <Z>0</Z>
      </Point3D>
      <Point3D>
        <X>5</X>
        <Y>5</Y>
        <Z>0</Z>
      </Point3D>
      <Point3D>
        <X>5</X>
        <Y>-5</Y>
        <Z>0</Z>
      </Point3D>
      <Point3D>
        <X>5</X>
        <Y>-5</Y>
        <Z>0</Z>
      </Point3D>
      <Point3D>
        <X>-5</X>
        <Y>-5</Y>
        <Z>0</Z>
      </Point3D>
    </Points>
    <ImageSize>
      <Width>0</Width>
      <Height>0</Height>
    </ImageSize>
    <MaxLength>0</MaxLength>
    <Weld>true</Weld>
    <Locked>false</Locked>
    <WeldLength>40</WeldLength>
```


*Figure 4-22 - XML File Example (Opened using Google Chrome)*

### 4.4.7 DXF Export

Any Multi Shape object can be exported as a DXF file. To do this, right-click on a Multi Shape and select “Export to DXF.” See Figure 4-23.

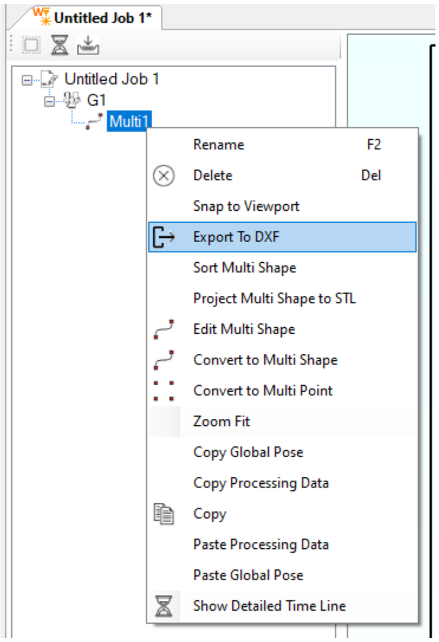
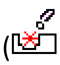



*Figure 4-23 - Export Multi Shape to DXF*

## 4.5 Parameter Tools

### 4.5.1 Processing Window

The following section describes the features and use of the Processing Window. The Processing Window can be opened by pressing the Processing Window button (), by pressing the "F5" button, or by selecting "Start Processing" under the Tools menu.

When opening the Processing Window, only the selected processing objects and action controls from the Job Tree will be brought into the Processing Window. If the job is selected, then all groups in the job will be brought into the Processing Window. For all groups selected, all of their objects will be brought into the Processing Window. Figure 4-24 shows the Processing Window when the job is selected before opening the Processing Window. Figure 4-25 shows the Processing Window only.

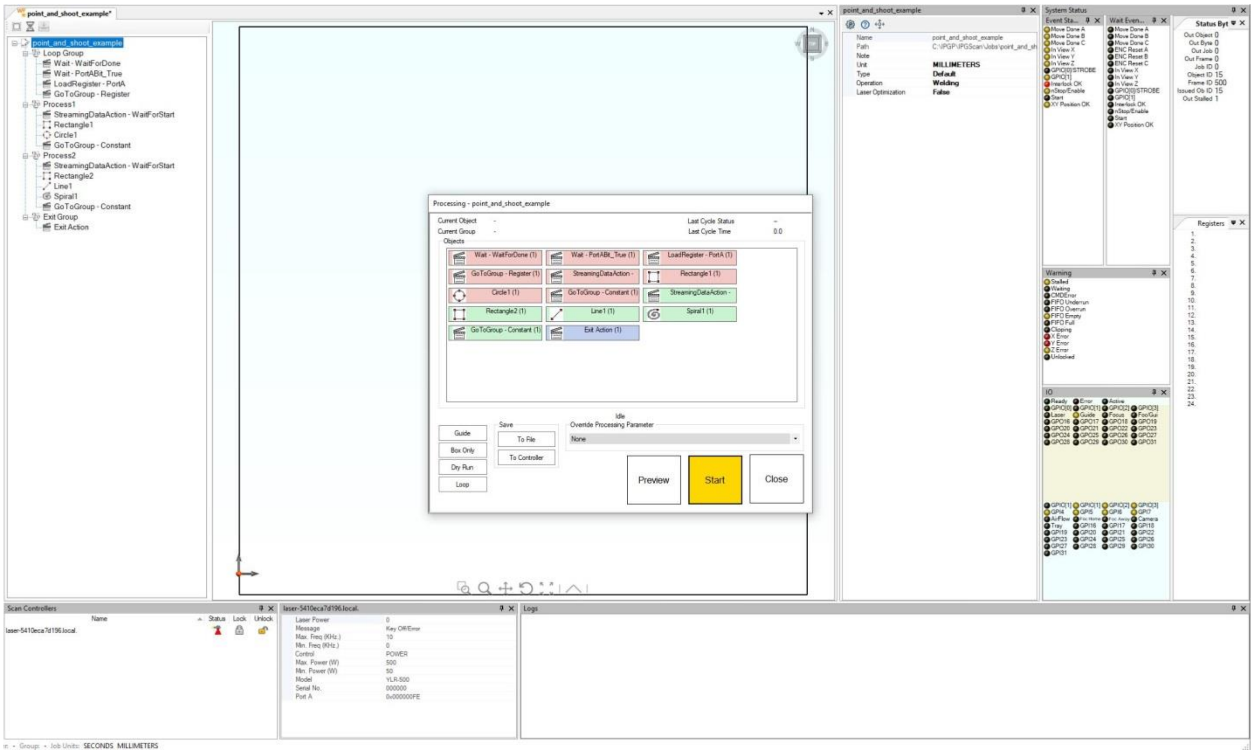



*Figure 4-24 - Processing Window in IPGScan*

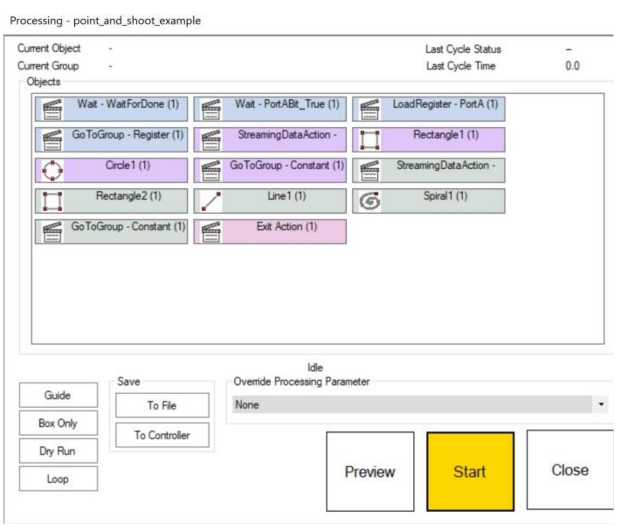



*Figure 4-25 - Processing Window*

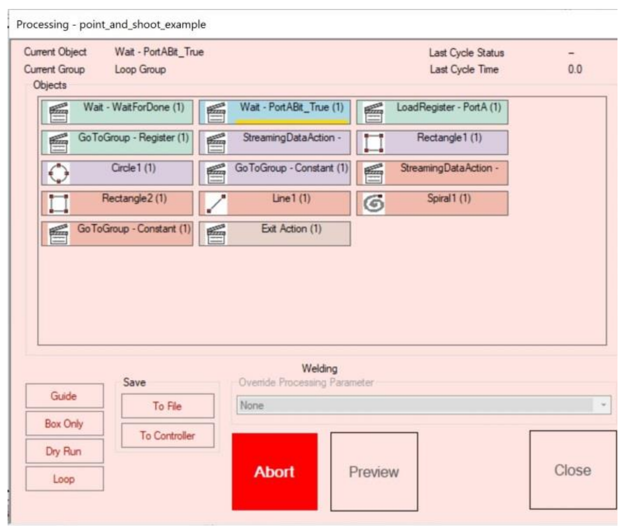



*Figure 4-26 - Processing Window When Processing is Active*

Within the Processing Window, the objects that will be run are listed in order of execution (not including any “GoToGroup” actions) within the “Objects” box. Table 4-3 describes the buttons found on the Processing Window.

*Table 4-3 - Processing Window Buttons*

| Button | Description |
|---|---|
| Guide | The processing objects will be output with the guide beam regardless of the laser configuration of the object. |

| | |
|---|---|
| Box Only | Outputs a box around the output a position of each processing object. |
| Dryrun | The IPGScan job will be processed by ScanPack but the laser will not be output and the galvos of the scan head will not move. This can be used with Robotic On-The-Fly jobs to evaluate the feasibility of the job. |
| Loop | The IPGScan job will be restarted after the last object is finished until the user stops execution. |
| Fast Mode | Outputs the processing objects with the “Fast Mode” processing parameters overwriting the programmed processing settings. Only available when “Guide” is selected. |
| Close | Closes the Processing Window if processing is not active. |
| Start | Start processing. |
| Preview | Guide and Loop will automatically be selected and then processing will be started. |
| Abort | Stops execution, stops firing the laser, and flushes the buffer. See Figure 4-26. |

#### *4.5.1.1 Job Information*

At the top of the Processing Window are four pieces of information about the job: Current Object, Current Group, Last Cycle Status, and Last Cycle Time. Table 4-4 describes each status. Figure 4-27 shows the job information in the top purple rectangle.

*Table 4-4 - Job Information*

| **Information** | **Description** |
|---|---|
| Current Object | The name of the object currently being output by the scanner. |
| Current Group | The name of the group which is currently being processed. |
| Last Cycle Status | The status of the previously run cycle; either “OK” or “Failed.” |
| Last Cycle Time | The time elapsed during the previously run cycle. |

#### *4.5.1.2 Override Processing Parameters*

The drop down menu under “Override Processing Parameter” will show all of the Processing Parameters set up in IPGScan (see Section 4.11.5). Selecting “None” will run the processing objects with their programmed configuration. Selecting any of the processing parameters will override the programming of the processing objects and will use the configuration specified in the selected processing parameter. Figure 4-27 shows this drop down in the lower blue rectangle.

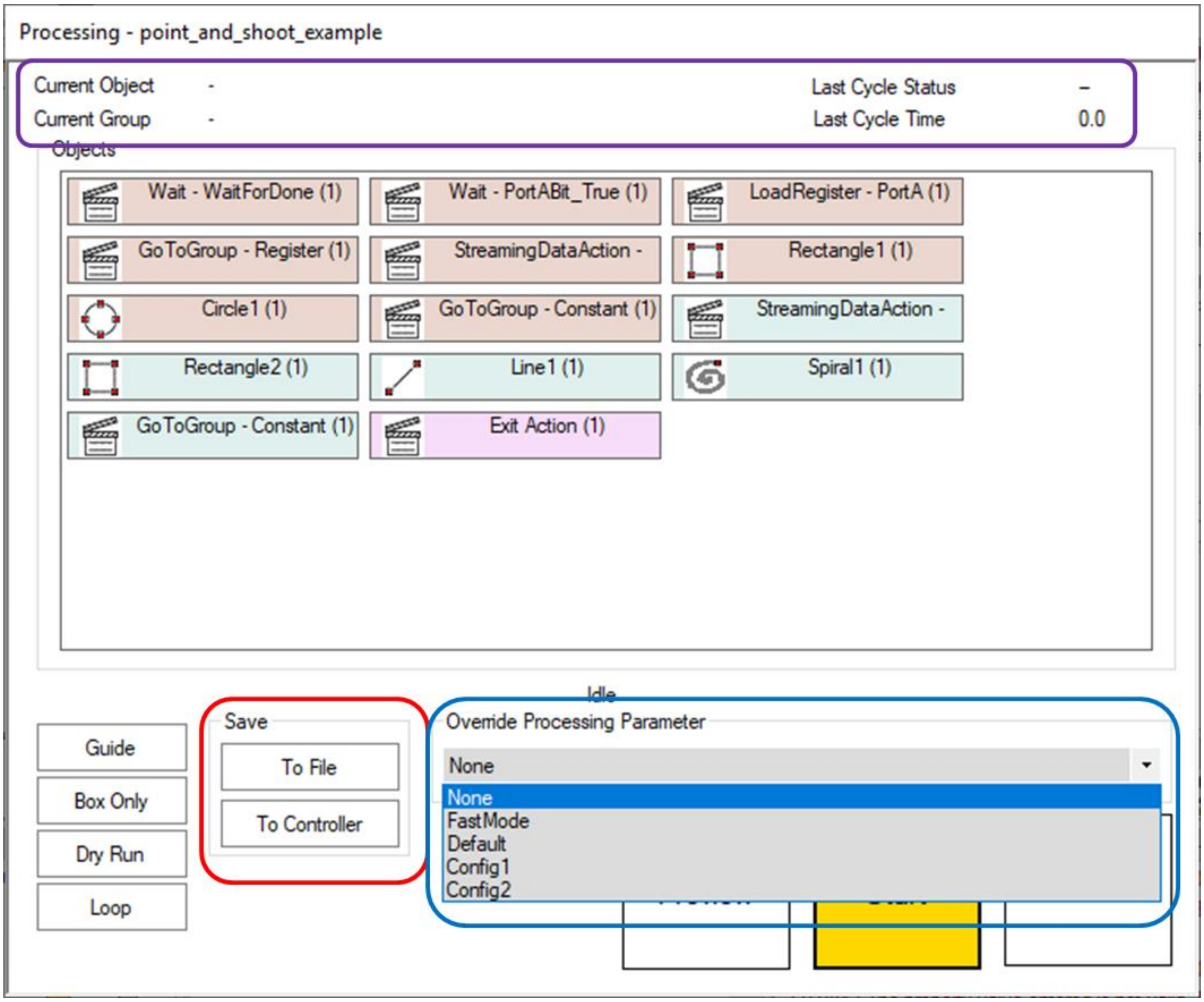



*Figure 4-27 - Processing Window Sections*

### 4.5.2 Select All

All weld shapes can be selected by clicking on the job or any of the groups in the job tree. Then "Select All Shapes" can be chosen (see Figure 4-28). This will select all of the weld shapes in the collection and leave out any ActionControls. If the job is selected, the "Select All Shapes" icon ( ) can be selected instead.

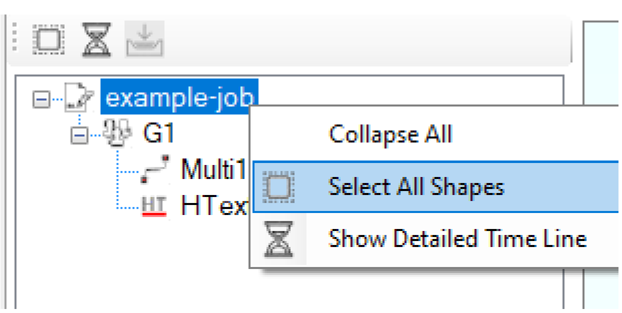



*Figure 4-28 - Select All Shapes*

### 4.5.3 Display Process Parameters Only

At the top of the Parameter Window is the "Display Processing Parameters Only" toggle. This will limit the Parameter Window to only Processing Parameters for processing objects (compared to ActionControls and Reference objects).

### 4.5.4 Copy/Paste Processing Data

For objects, users can copy the processing parameters of an object and paste the processing data onto another object.

1. In the Job Tree, right click on an object.
2. In the drop down menu users can select:
    a. Copy Processing Data: copies an object's processing data to a clipboard.
    b. Paste Processing Data: sets an object's processing data to the previously copied processing data.

## 4.6 System Status

The System Status windows are a collection of windows which show different statuses of the current system. They are available under the View dropdown menu. For the statuses which are represented by LEDs, a lit LED shows a true state (eg. "In View X" or "Interlock OK" in Figure 4-29). An unlit LED shows a false state (eg. "GPIO[1]" or "Start" in Figure 4-29).

### 4.6.1 Event Status

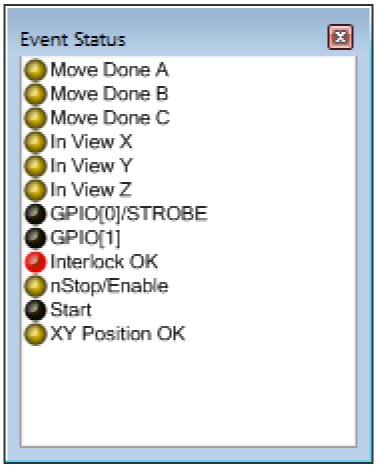



*Figure 4-29 - Event Status*

Figure 4-29 shows the event status window. This window shows when each event is or is not currently true/active.

1. Move Done A/B/C
    a. True when the stage is done moving axis A/B/C.
2. In View X/Y/Z
    a. True when the next vector is within the In View window for the Scan Head.
3. GPIO[0/1]

   a. True when each signal GPIO[0/1] is currently active. This state will adapt with the configuration of each signal for active low versus high and input versus output.
4. Interlock OK
   a. True when the fiber interlock is satisfied.
5. nStop/Enable
   a. True when the nStop (not stop)/Enable signal is true. This will allow the scanner system to run.
6. Start
   a. True when the Start signal is active.
7. XY Position OK
   a. True when the X and Y galvos are in position.

### 4.6.2 Wait Event Status

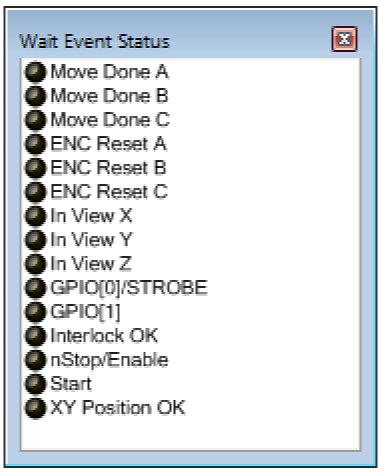



*Figure 4-30 - Wait Event Status*

Figure 4-30 shows the Wait Event Status window. Active signals in this window represent events which are blocking the execution of the buffer.

1. Move Done A/B/C
   a. The scanner is waiting for motion to finish on stage axis A/B/C.
2. ENC Reset A/B/C
   a. The scanner is waiting for Encoder A/B/C to reset.
3. In View X/Y/Z
   a. The scanner is waiting for the next output vector to become In View in the X/Y/Z direction(s).
4. GPIO[0/1]
   a. The scanner is waiting for the GPIO[0/1] signal to be active.
5. Interlock OK
   a. The scanner is waiting for the interlock to be satisfied.
6. nStop/Enable
   a. The scanner is waiting for the nStop (not Stop)/Enable signal to be active.

7. Start
    a. The scanner is waiting for the Start signal to be active.

### 4.6.3 Warning Status

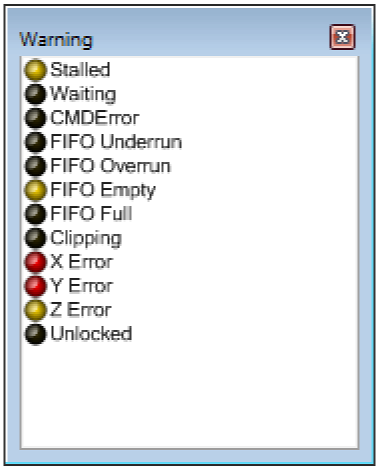



*Figure 4-31 - Warning Status*

Figure 4-31 shows the Warning status window. The information in this window displays different warnings about the scanner system.

1. Stalled
    a. True when there is no data being processed from the buffer.
2. Waiting
    a. Waiting for an event to be satisfied (see Section Wait Event Status)
3. CMDError
    a. True when the command from the computer does not match the command received on the scan controller.
4. FIFO Underrun
    a. Active when a FIFO underrun occurs. This occurs when the scan controller reaches the end of the FIFO (buffer) without seeing a properly terminated command. This usually indicates a communication issue.
5. FIFO Overrun
    a. Active when a FIFO overrun occurs. This occurs when the computer driving the scan controller attempts to add data to the buffer when it is already full. This is indicative of a communication issue.
6. FIFO Empty
    a. Active when the FIFO is empty. This warning is only a cause for concern when the FIFO is not expected to be empty.
7. FIFO Full
    a. Active when the FIFO is full.
8. Clipping

   a. True when the galvos are being commanded to an unreachable position. This usually only occurs during STAGE_TRACKING when the scanner system is waiting for the next vector to enter the In View window.

9. X/Y/Z Error
   a. Active when there is an error in the X/Y/Z galvo of the Scan Head. A Z error is expected in 2D heads.

### 4.6.4 IO

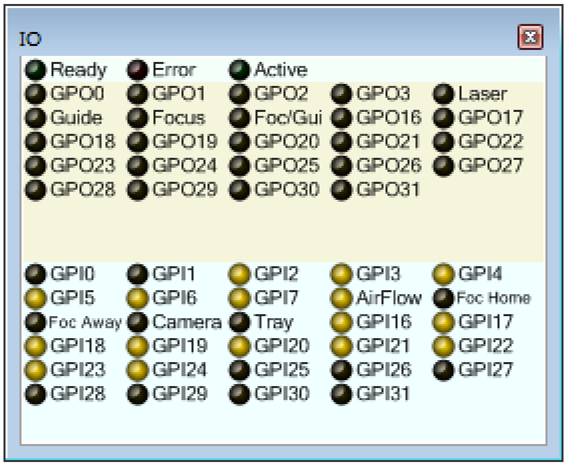



*Figure 4-32 - IO Status*

Figure 4-32 shows the IO status window. The top white section shows Port F; the middle yellow section shows Port C (outputs); the bottom white section shows Port A (inputs). An active LED light corresponds to an active signal. An inactive LED light corresponds to an inactive signal.

### 4.6.5 Status Bytes

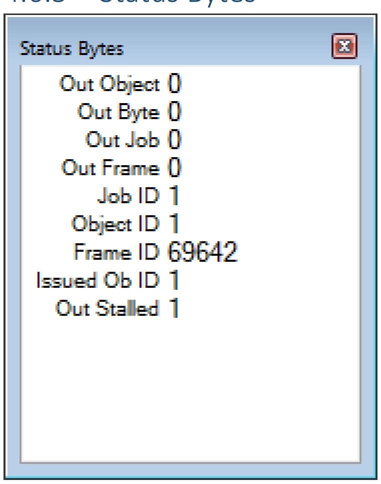



*Figure 4-33 - Status Bytes*

Figure 4-33 shows the status bytes window. This window shows information about the scanner system and the FIFO.

1. Out Object

   a. The current number of objects in the FIFO. The scan controller will not process commands from the FIFO until the FIFO is either at least half full or there is at least one object in the FIFO.
2. Out Byte
   a. The current number of bytes in the FIFO.
3. Out Job
   a. The current number of jobs in the FIFO.
4. Out Frame
   a. The current number of frames in the FIFO.
5. Job ID
   a. In IPGScan, this represents the Group ID of the most recently completed group.
6. Object ID
   a. In IPGScan, this represents a count of the number of objects started by the scan controller in the current group.
7. Frame ID
   a. An incrementing count of the number of frames executed by the scan controller.
8. Issued Object ID
   a. In IPGScan, this represents a count of the number of objects sent to the FIFO.

### 4.6.6 Registers

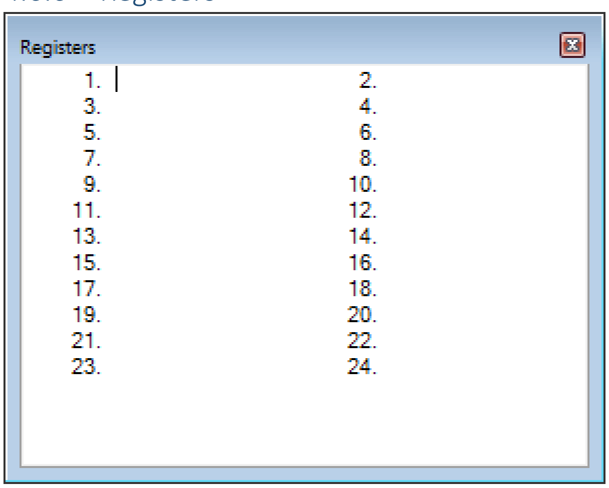



*Figure 4-34 - Registers*

Figure 4-34 shows the registers window. This window shows the current value of all registers in IPGScan.

### 4.6.7 Dirty Window Sensor

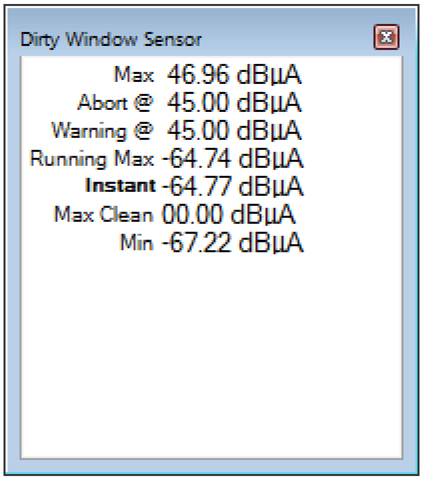



*Figure 4-35 - Dirty Window Sensor*

Figure 4-35 shows the Dirty Window Sensor window which describes data recorded by the dirty window sensor.

1. Max
    a. The maximum possible value.
2. Abort @
    a. If the dirty window sensor reads a value at or above this value, the currently executing job will abort.
3. Warning @
    a. If the dirty window sensor reads a value at or above this value, a warning will display on the screen. This will not interrupt execution.
4. Running Max
    a. The maximum value read by the dirty window sensor during this tracked segment. The tracked segment can be reset by right clicking on the words “Running Max.”
5. Instant
    a. The current value read by the dirty window sensor.
6. Max Clean
    a. The Max Clean value from the options.
7. Min
    a. The minimum possible value.

### 4.6.8 Overview

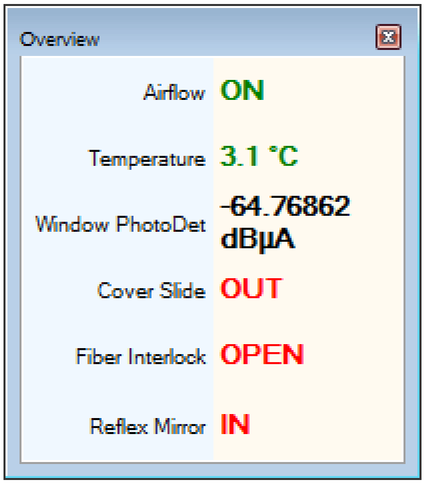



*Figure 4-36 - Overview*

Figure 4-36 shows the overview window. This window shows general information about the scanner system.

1. Airflow
   a. The status of the airknife for the system [ON/OFF]
2. Temperature
   a. The temperature inside of the scan head
3. Window PhotoDetector
   a. The current value of the dirty window sensor
4. Cover Slide
   a. The position of the cover slide in the collimator [IN/OUT]
5. Fiber Interlock
   a. The status of the fiber interlock in the scan head [OPEN/CLOSED]
6. Reflex Mirror
   a. The position of the reflex mirror in the scan head [IN/OUT]

### 4.6.9 Stage Positions

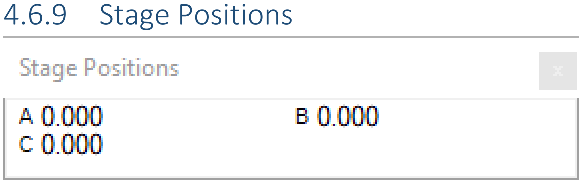



*Figure 4-37 - Stage Positions*

Figure 4-37 shows the Stage Positions window. This window will keep updated with the current position of the stage system. This does not work with robots for robot tracking.

## 4.7 Port Logs

Port Logs allow users to view the TCP or Serial commands between IPGScan and an external device. Figure 4-38 shows a screenshot of the Port Logs. If this window is not visible, click on "Port Logs" under the Tools dropdown menu.

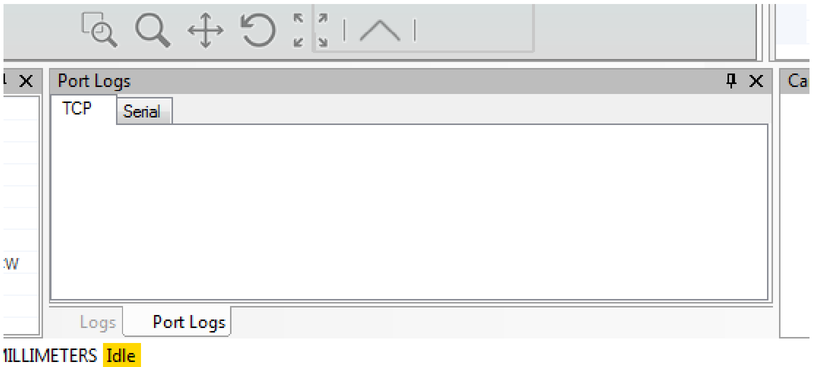



*Figure 4-38 - Port Logs*

To clear this window, right click on the Port Logs window and click "Clear" (see Figure 4-39).

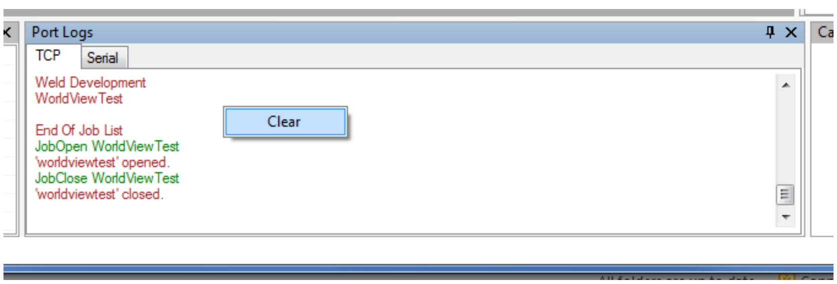



*Figure 4-39 - Clear Port Logs*

## 4.8 Concepts

### 4.8.1 Execution Buffer

An IPGScan job is mostly run sequentially in an execution buffer in ScanPack. All IPGScan processing objects and some ActionControls are sent from IPGScan to the ScanPack buffer. Some other IPGScan Action Controls are executed immediately at the IPGScan level.

#### *4.8.1.1 Detailed Information*

All of the commands executed by the Scan Controller are organized in a buffer. The buffer is maintained in a FIFO structure. The Scan Controller waits on the head of the buffer to receive the next command to execute.

Commands sent to the buffer are made up of the following entities (Table 4-5). Table 4-6 shows the ID counters which are tracked as the buffer progresses.

*Table 4-5 - ScanPack Buffer Entities*

| Entity | Description |
|---|---|
| Frame | A frame contains the Scan Controller instructions for one time cycle, around 10 µs. A larger instruction, like outputting vectors, could be made up of multiple frames. |

| | |
|---|---|
| Object | A complete set of actions for the Scan Controller. This can include instructions which purposely do not finish an object. |
| End of Frame (EOF) | A flag which allows the current frame to be written to the buffer. |
| End of Object (EOO) | Allows frames in the buffer to execute. The scan controller will not begin to dequeuer frames from the buffer for execution until an EOO is present or the buffer is at least half full. |
| End of Job (EOJ) | A flag which tells the Scan Controller that the current object finishes the job. This flag is sent by LaserLib (installed with ScanPack) when the laser is turned off. |

*Table 4-6 - ScanPack Buffer ID Counters*

| ID | Description |
|---|---|
| FrameID | Counts the number of Frames. The FrameID starts at 0 and is incremented by EOF entities. FrameID is only reset with a new instance of ScanPack. This can be done by restarting IPGScan. |
| ObjectID | In IPGScan, this counts the number of objects output by the scanner. This value is reset to 0 after going to a new group. |
| JobID | In IPGScan, this is the GroupID of group most recently completed. |

### 4.8.2 Groups and Group IDs

Groups are a critical part of creating programs in IPGScan. Not only are they used to organize programs into sub-sections, but they also are used to carry out a number of functions. For example, groups allow users to skip over or skip to different sets of process objects depending on their scenario.

Use the Action Controls GoToGroup (section 4.13.9) and Load Register (Section 4.13.10) to select different groups during execution. The following provides information regarding Groups and their use.

Groups are created using the “Create Group” button  in the toolbar. Figure 4-40 shows a job containing four empty Groups.

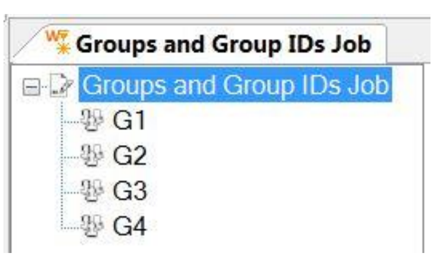



*Figure 4-40 Groups in the Job Tree*

Each Group has a Group ID. The Group ID is used as an identifier for its corresponding Group. In order to set the Group ID, click on the Group in the Job Tree and then change the value “Group ID” in the

Properties Window. Figure 4-41 shows Group G3 being selected in the Job Tree and its corresponding Group ID is 3. Group ID values are limited to integers. The Group ID has no relation to the name of the group. For example, a Group name “Group 20” could have a Group ID of 32.

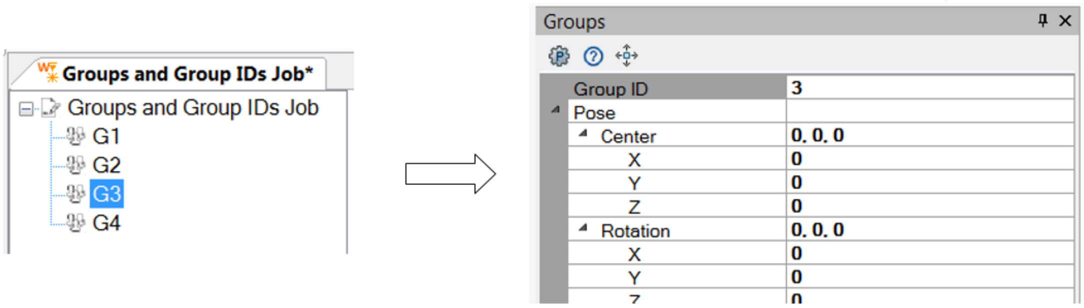



*Figure 4-41 Setting Group ID*

Group ID’s are the basis behind how Point and Shoot Processing works (Section 11). In a Robot/PLC program, users reference the desired IPGScan Group through the use of the Group ID. The IPGScan Controller might receive the desired group ID via Digital Bits (Port A), Ethernet communication, or any other viable option outlined in the Load Register Action Control (Section 4.13.10).

### 4.8.3 Coordinate Systems

The following figures relate the coordinate system of IPGScan and the coordinate system of the Scan Head. The figures below diagram the two coordinate systems. In these figures the positive X axis is represented in green, the positive Y axis is represented in blue, and the positive Z axis is represented in red. In the IPGScan canvas diagram, the positive Z axis is going out of the page. In a default job (represented below), objects closer to the bottom of the IPGScan canvas will be output closer to the IPG logo on the Scan Head. For an On-The-Fly job, the IPGScan axes match the user calibrated coordinate system.

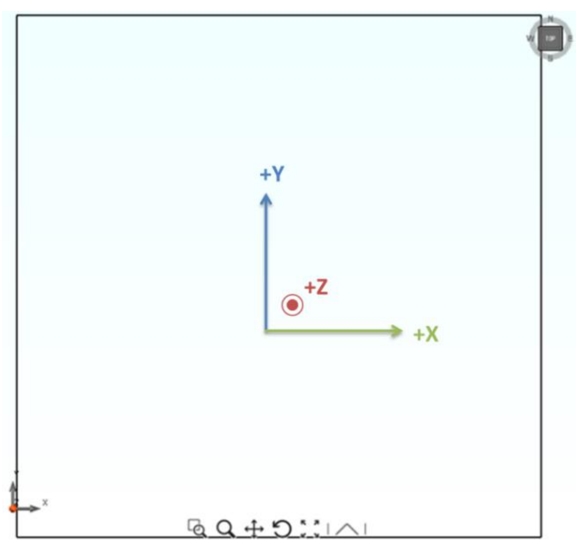



*Figure 4-42 - IPGScan Default Coordinate System*

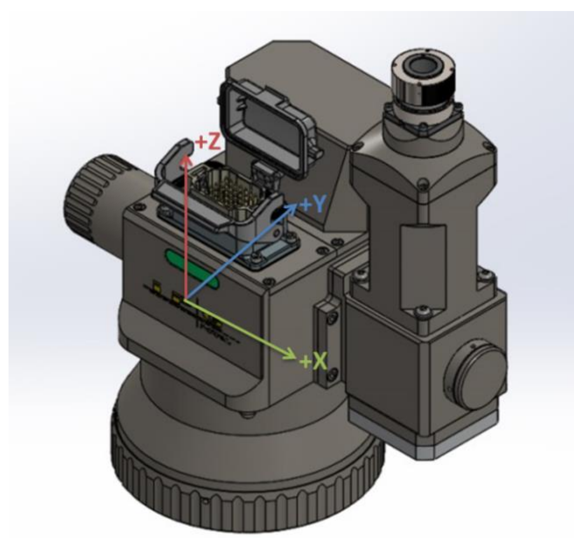



*Figure 4-43 - Scan Head Default Coordinate System*

**Note:** The symbol represents the direction out of the page and towards the reader.

**Note:** In Figure 4-42 and Figure 4-43: the positive X direction is green, the positive Y direction is blue, and the positive Z direction is red.

1. Select View→Options→Security. Figure 4-44 will be shown.

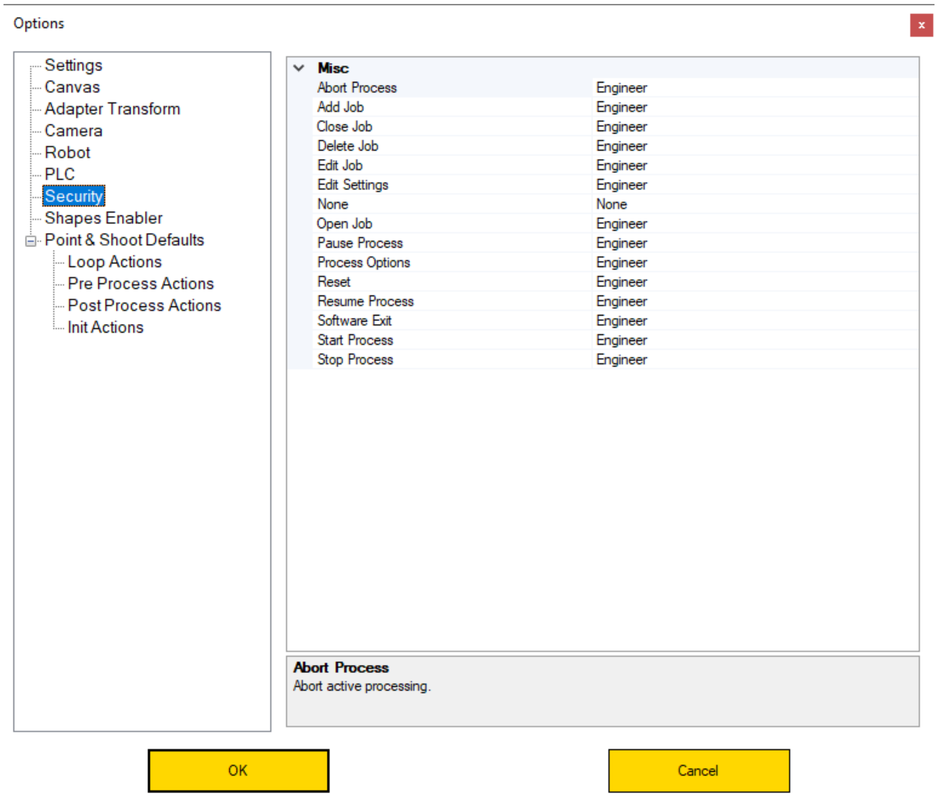



*Figure 4-44 - Security Settings*

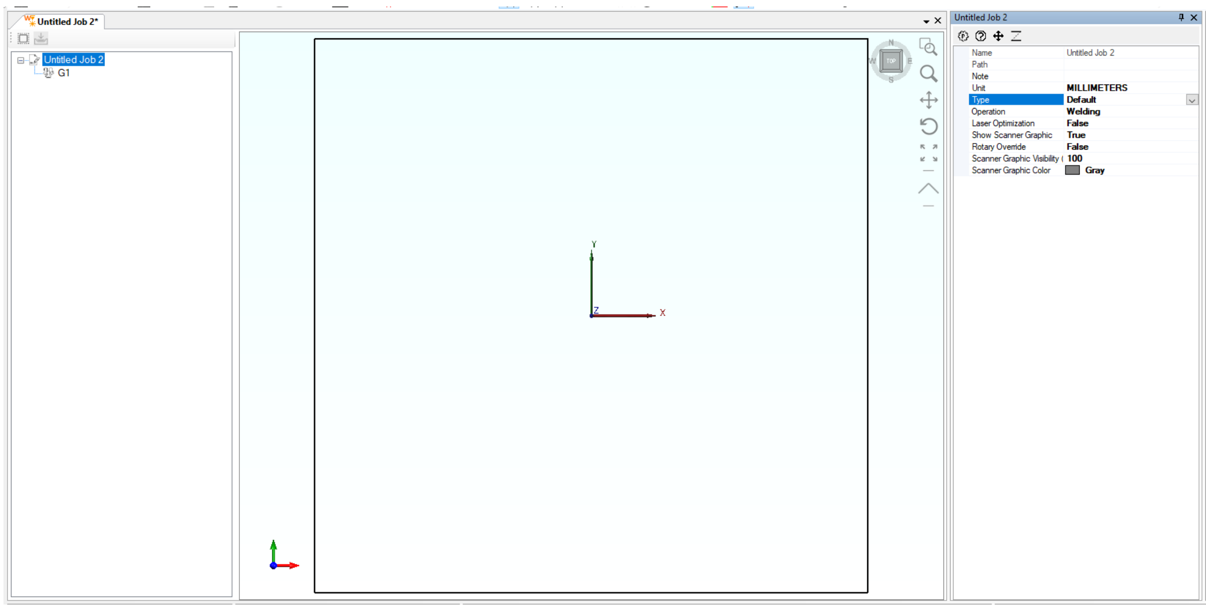
2. Click on the listbox next to a function to select the security level for that function.

## 4.9 Creating a Job

The following section details how users can create a new Job in IPGScan, create Groups within the Job, and add Process Objects or Action Controls to Groups.

1. Select File → New or click the New icon in the Tool Bar to create a new Job file. An unsaved and untitled Job will as appear as shown in Figure 4-45.

*Figure 4-45 New Job Window*

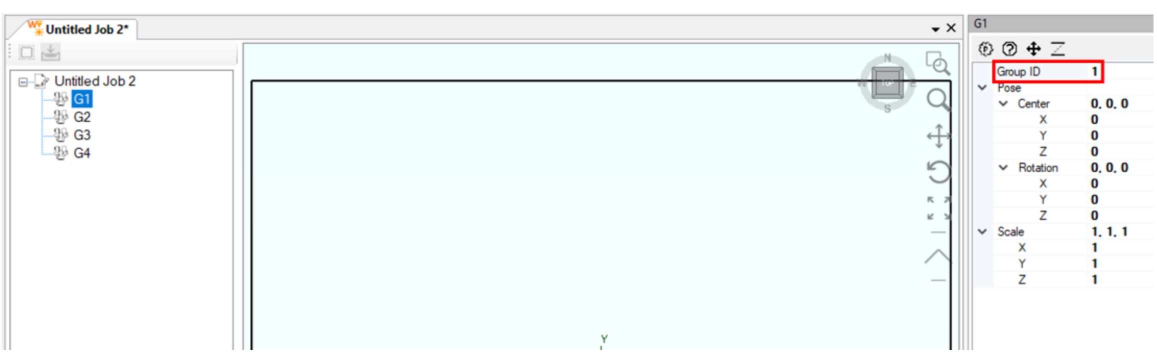



The Job Tree window displays the new Job. When the Job is selected, the type of Job is listed in the Properties window. Job types (Default / On The Fly / Point & Shoot) can be changed in this menu.

Once a Job is created, users can then add Groups to the Job. Groups allow users to break the Job into sections that can be called/executed at desired times in a process using automation equipment such as a PLC or Robot Controller. By selecting a Group in the Job Tree, users can locate the Group ID for the Group, as shown in Figure 4-46. The Group ID is what allows users to utilize the SELECT Bits on an External Interface to select which Group to execute and when.

2. To create a Group in IPGScan, click File → New Group or click the Create Group icon in the Tool Bar.

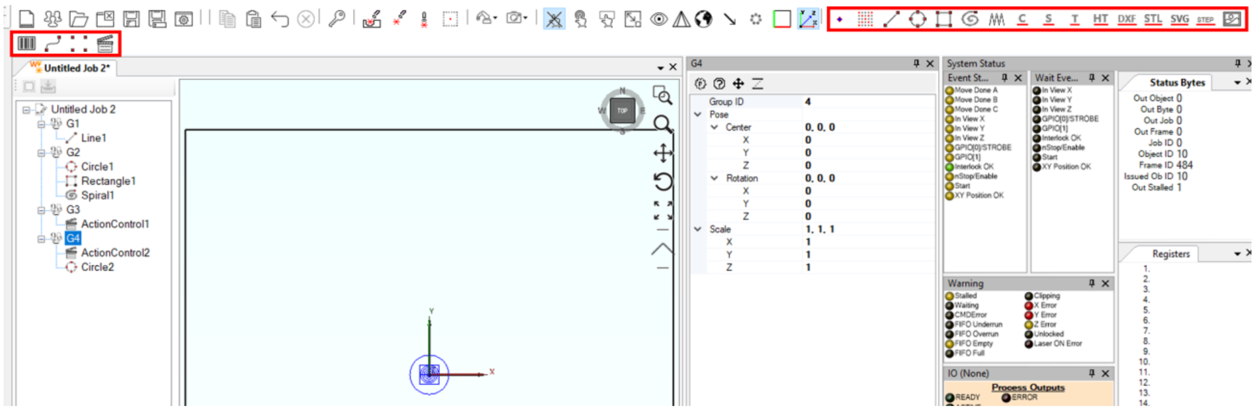
*Figure 4-46 Multiple Groups within a Single Job*



Within Groups, Process Objects and Action Controls can be added. Process Objects are the patterns that the user wishes to have the scanner perform while firing the laser while Action Controls provide additional control functionality within the IPGScan Job.

3. To add a Process Object or Action Control to a particular Group, simply select the desired Group in the Job Tree and click on the desired Process Object icon or Action Control icon in the Tool Bar. See Figure 4-47.

*Figure 4-47 Adding Process Objects and Action Controls to Groups*




If users wish to rename Groups, Process Objects, or Action Controls, simply right click on the desired item and click Rename. Selecting the desired item in the Job Tree and clicking F2 on a keyboard will also allow users to rename items.

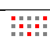
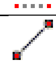
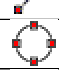
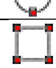
Finally, once a Job has been created, users can save the Job clicking File → Save or by clicking the Save icon.

With a basic understanding of how to create an IPGScan Job and Job structure, users can begin to develop a process.

## 4.10 IPGScan Objects

The following table (Table 4-7) outlines the processing objects which can be added to an IPGScan job. Objects can be added by clicking on its icon in the toolbar, in the “Tools” dropdown menu, or by using the corresponding keyboard shortcut. Some objects are processing objects which will cause the laser to fire when run; other objects are reference objects. Reference objects can be viewed in IPGScan and are meant to assist the user in placing processing objects and visualizing entire systems.

*Table 4-7 - IPGScan Objects*

| Object | Icon | Keyboard Shortcut | Object Type |
|---|---|---|---|
| Point |  | CTRL+ALT+P | Processing |
| Point Array |  | CTRL+ALT+A | Processing |
| Line |  | CTRL+ALT+L | Processing |
| Circle |  | CTRL+ALT+C | Processing |
| Rectangle |  | CTRL+ALT+R | Processing |

| | | | |
|---|---|---|---|
| Spiral | | CTRL+ALT+S | Processing |
| Zigzag | | CTRL+ALT+Z | Processing |
| C Shape | C | CTRL+ALT+E | Processing |
| S Shape | S | CTRL+ALT+X | Processing |
| Text | T | CTRL+ALT+T | Processing |
| Hershey Text | HT | CTRL+ALT+H | Processing |
| DXF | DXF | CTRL+ALT+D | Processing |
| STL | STL | CTRL+ALT+Y | Reference |
| SVG | SVG | CTRL+ALT+V | Processing |
| STEP | STEP | CTRL+ALT+G | Reference |
| Multi Shape | | CTRL+ALT+Y | Processing |
| Points | | CTRL+ALT+O | Processing |

### 4.10.1 Multiple Creation

Multiple IPGScan objects can be created at the same time. To create multiple instances of the same object, "CTRL + click" on the shape icon in the toolbar. The "Multi Create" window (below Figure 4-48) will appear to help the user create an array of the desired object. The user is asked to specify the number of desired rows and columns, the distance separating the columns and rows, and the coordinate at which to start creating instances.

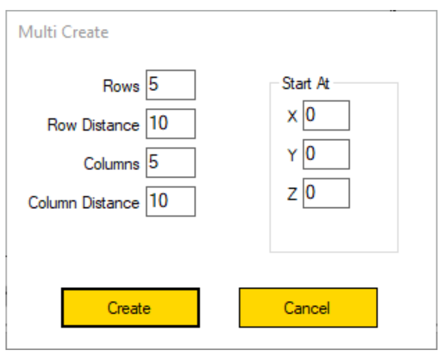



*Figure 4-48 - Multi Create Window*

### 4.10.2 Configuration Parameters

Each IPGScan Object has several configuration parameters which control different properties of the object. Section 4.10.2.1 describes the parameters which are common to most IPGScan objects. Section 4.10.2.1.3 describes the parameters which are specific for a subset of IPGScan objects.

#### *4.10.2.1 Common Configuration Parameters*

- Max Length
  - Max Length defines the maximum segment length. Segments longer than this value will be split into multiple segments. This can be used with the features like Energy Offset.
  - A value of 0 will leave segments at their specified length.
- Locked
  - When true, all properties of the locked object will be hidden an un-editable.

- Object Pose
  - Center – (X, Y, Z) coordinates, in millimeters, of the center of the object
  - Rotation – rotation angle, in degrees, of the object about each axis
- Global Pose
  - An additional pose used with Robot On-The-Fly jobs
- Apply Offsets
  - See Section 4.10.2.1.1
- Reverse
  - When true, the direction and execution order of the vectors or points of the object will be reversed
- Loop
  - Defines the number of times to execute the object
- Fill
  - See section 4.10.2.1.2.

#### 4.10.2.1.1 Apply Offsets

When Apply Offsets is enabled, users have the ability to change object parameters (position, rotation, and process parameters) using register values during the execution of a job.

To use Offsets:

1. Select the desired feature(s) in the Job Tree.
2. Set Apply Offsets to True in the parameter window.
   a. This causes Offsets to appear within the parameter window.
3. Click on the ellipsis in the Offsets box. The OffsetSelector Collection Editor will appear (Figure 4-49).

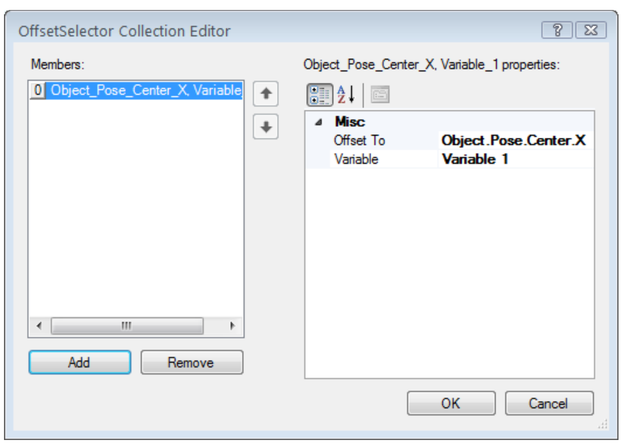



*Figure 4-49 - OffsetSelector Collection Editor*

4. In the OffsetSelector Collection Editor, click the Add button in order to add an offset.
5. Under the “Properties” section, set the “Offset To” parameter to the desired feature to be offset.
   a. The following parameters can have offsets applied
      i. Object.Pose.Center.X
      ii. Object.Pose.Center.Y

iii. Object.Pose.Center.Z
iv. Object.Pose.Rotation.X
v. Object.Pose.Rotation.Y
vi. Object.Pose.Rotation.Z
vii. Scale X
viii. Scale Y
ix. Scale Z
x. Velocity
xi. Power
xii. Pulse Width
xiii. Pulse Frequency
xiv. Keyhole Time
xv. Energy
xvi. Pitch
xvii. Relative Speed
xviii. Frequency

6. Once a parameter has been selected for offset, set the Variable that will be referenced for the offset data.
    a. For loading data into registers, please refer to section 4.13.10.
7. Repeat steps 4 and 5 until all desired offsets are applied to the feature(s).
8. Once all desired offsets are applied, close and save the OffsetSelector Collection Editor box by clicking on Ok.

#### 4.10.2.1.2 Fill

Some objects can be filled with vectors. If Fill is being used, the following configuration parameters are used.

- Beam Diameter – The beam diameter of the optical setup. This is applied as a buffer around the shape outline.
- Fill Angle – the angle of the fill vectors in degrees
- Fill Pitch – the distance between the fill vectors
- Shape Outline – when true, the outline of the object will be output by the scanner as well as the fill vectors
- Fill Direction
    - Unidirectional – all fill vectors will be in the same direction
    - Bidirectional – each fill vector will alternate the execution direction
- Fill Type
    - EdgeToEdge – The fill vectors will start and end on opposite edges of the object (Figure 4-50)
    - IsoGrid – Every other fill vector will be offset along the direction of the vector (Figure 4-51)
    - Orthogonal – The start and ending points of the fill vectors will be in line if possible (Figure 4-52)

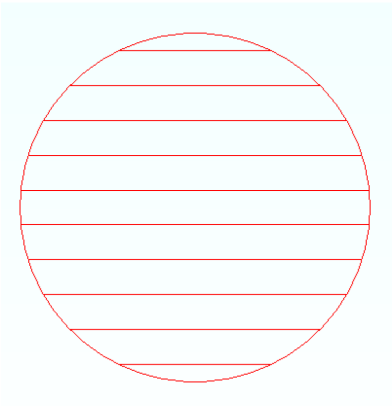


*Figure 4-50 - Edge To Edge Fill*

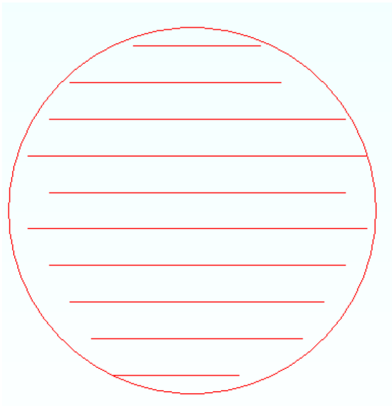


*Figure 4-51 - IsoGrid Fill*

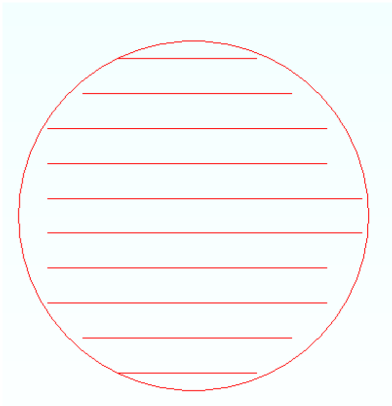


*Figure 4-52 - Orthogonal Fill*

#### 4.10.2.1.3 Sort

There are two algorithms for sorting vectors. Table 4-8 describes these algorithms.

*Table 4-8 - Sort Algorithms*

| Algorithm Name | Description |
|---|---|
| Polyline | |
| StartProximity | Starting with the first vector, this algorithm finds the vector with the closest starting point to the current vector's ending point. It makes that vector the next vector and continues the algorithm until there are no more vectors left to sort. |

### *4.10.2.2 Shape Specific Configuration Parameters*

#### 4.10.2.2.1 Point Array

- Height – the height of the point array in distance units
- Width – the width of the point array in distance units
- Distance – the space between points in distance units
- Direction
  - Unidirection – each row of the point array will be ordered in the same direction
  - Bidirectional – each row of the point array will alternate its order so the execution order will snake around the point array
- Projection – the layout of the point array. See the fill types in Section 4.10.2.1.2.

#### 4.10.2.2.2 Line

- Length – the length of the line in distance units

#### 4.10.2.2.3 Circle

- Radius – radius of the circle
- Start Angle - starting angle (in degrees) of the circle. For example, if the starting angle is 0, it means the circle starts at 3 o'clock; and if the starting angle is 270, it means the circle starts at 6 o'clock.
- End Angle – ending angle (in degrees) of the circle. For example, if the ending angle is 90, it means the circle ends at 12 o'clock; and if the ending angle is 270, it means the circle ends at 6 o'clock.
- "Start Angle" and "End Angle" together define the portion of a circle; It can be a full circle, multiple revolutions, or an arc. Together they also define the direction of the circle. For example, with a "Start Angle" of 180 degrees and an "End Angle" of 90 degrees, the circle would

start at 9 o'clock and continues clockwise until it ends at 12 o'clock (see Figure 4-53).

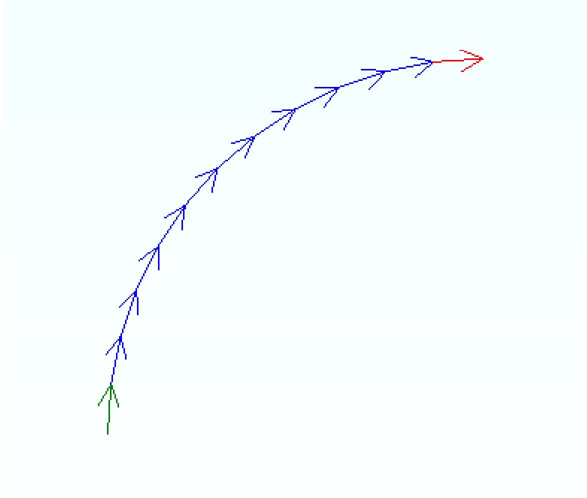


*Figure 4-53 - Partial Circle Example*

- Points Per Revolution - defines the number of points for each revolution of the circle (See Figure 4-54, Figure 4-55, Figure 4-56).

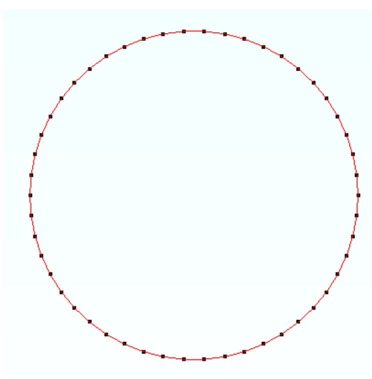


*Figure 4-54 - Circle with 50 Points per Revolution*

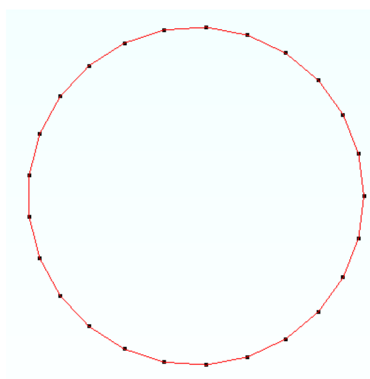


*Figure 4-55 - Circle with 25 Points per Revolution*

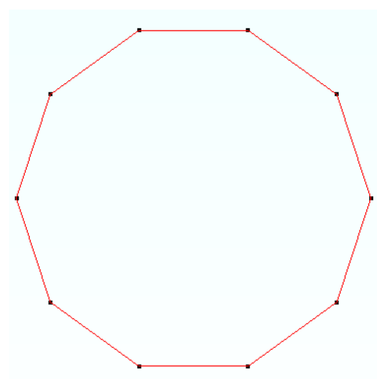


*Figure 4-56 - Circle with 10 Points per Revolution*

- Z Step - defines the increase in Z coordinate per revolution. For example, a circle with a start angle of 0 degrees, an end angle of 720 degrees, and a Z step of 1mm looks like the pattern in Figure 4-57.

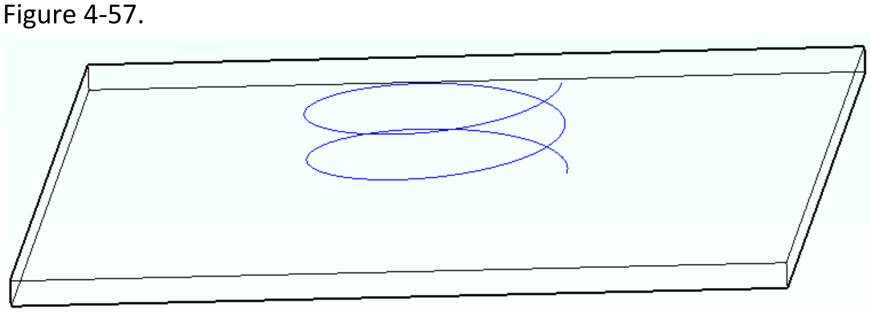


*Figure 4-57 - Circle with Z Step Example*

- Scale – Scale - changes the object's size by the specified factor in each axis.

#### 4.10.2.2.4 Rectangle

- Height – the height of the rectangle
- Width – the width of the rectangle
- Corner Radius – the radius of the corners of the rectangle. The example in Figure 4-58 shows a rectangle of 15mm by 10mm with a corner radius of 2mm.

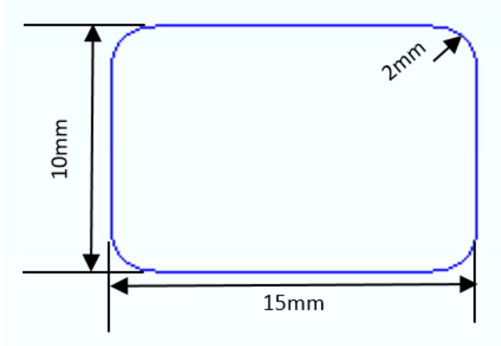



*Figure 4-58 - Corner Radius Rectangle Example*

- Segment – the number of segments in the rectangle. Segments less than 4 will leave an open rectangle; Segments greater than 4 will repeat the edges of the rectangle.

#### 4.10.2.2.5 Spiral

- Length – length of the spiral
- Width – width of the spiral
- Points Per Revolution – number of points for each revolution
- Z Step - defines the increase in Z coordinate (in mm) per revolution. The example below shows a spiral with a z-step of 2 (mm).

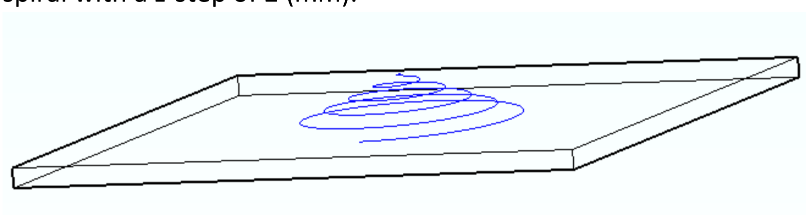


*Figure 4-59 - Spiral Z Step Example*

- Exclusion Radius – the radius at which the spiral ends. The example below shows a spiral with width & length of 25mm and an exclusion radius of 5mm.

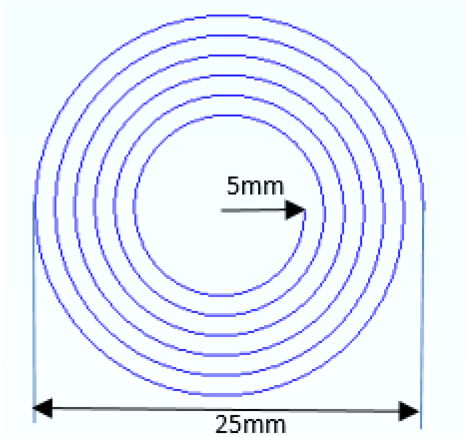



*Figure 4-60 - Spiral Exclusion Radius*

### 4.10.2.2.6 Zigzag

- Height – height (in mm) of the zigzag shape
- Width – width (in mm) of the zigzag shape
- Segment – number of segments in the zigzag shape. The example below shows a zigzag with height of 4mm, width of 10mm and segment of 10.

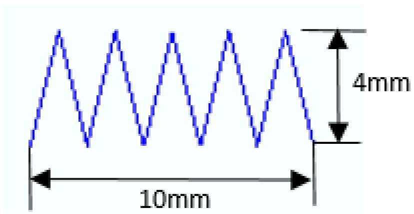



*Figure 4-61 - Zigzag Segments*

### 4.10.2.2.7 C Shape and S Shape

- Height – height of the shape
- Width – width of the shape
- Corner Radius – radius of the corners
- Segment – additional length starting at the beginning and end of each shape
- Scale - changes the object's size by the specified factor in each axis.
- Aspect Ratio – if true, the same scale factor will be used for all axes.

### 4.10.2.2.8 Text

- Text – the string to display
- Font – the font to use to display the string
- Font Style – stylization of the string. Options include: Regular, Bold, Italic, Underline, Strikeout.
- Height – the height of the text in distance units
- Width – the width of the text in distance units. As height is adjusted, this value will not change.

- Width % - set the width as a percentage of the original width. As height is adjusted, the resulting width of the text will scale.
- Dynamic text - if the text content is to be dynamically loaded, choose one of the variables in the list (see Figure 4-62) to store the dynamic text content. You can refer to section 4.13.10 on how to load a value to a variable. For static text content, keep 'None' checked and define the text content in 'Text' field.


*Figure 4-62 - Dynamic Text Variable List*

- Wrap Circle – defines if the text is to be wrapped along a circle
- Wrap Circle Radius – defines the radius of the circle that the text to be wrapped along if 'Wrap Circle' is set to true. The example below shows the text with wrap circle set to 'True' and 'Wrap Circle Radius' of 10mm.

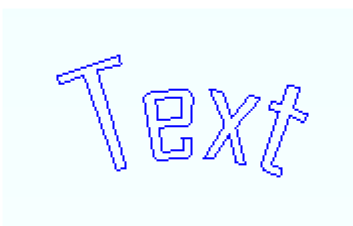



*Figure 4-63 - Text Wrap Circle Example*

- Scale - changes the object's size by the specified factor in each axis.
- Aspect Ratio – if true, the same scale factor will be used for all axes.

#### 4.10.2.2.9 Hershey Text

- Text - defines the text that will be displayed
- Font Size – defines font size of the Hershey Text
- Dynamic Text – if the text content is to be dynamically loaded, choose one of the variables in the list (see Figure 4-62) to store the dynamic text content. You can refer to 'Action Control' -> 'Load Register' action on how to load a value to a variable. For static text content, keep 'None' checked and define the text content in 'Text' field.
- Wrap Circle – defines if the H text is to be wrapped along a circle
- Wrap Circle Radius – defines the radius (in mm) of the circle that the text to be wrapped along if 'Wrap Circle' is set to true. The example below shows the text with wrap circle set to 'True' and 'Wrap Circle Radius' of 10 (mm).


*Figure 4-64 - Hershey Text Circle Wrap Example*

- Scale - changes the object’s size by the specified factor in each axis.
- Aspect Ratio – if true, the same scale factor will be used for all axes.

##### 4.10.2.2.10DXF

- Scale - changes the object’s size by the specified factor in each axis.
- Aspect Ratio – if true, the same scale factor will be used for all axes.

##### 4.10.2.2.11STL

- Transparency – adjust the opacity of the displayed model
- Color – adjust the color of the displayed model
- Scale - changes the object’s size by the specified factor in each axis.
- Aspect Ratio – if true, the same scale factor will be used for all axes.

##### 4.10.2.2.12SVG

- Scale - changes the object’s size by the specified factor in each axis.
- Aspect Ratio – if true, the same scale factor will be used for all axes.

##### 4.10.2.2.13STEP

- Transparency – adjust the opacity of the displayed model
- Color – adjust the color of the displayed model
- Scale - changes the object’s size by the specified factor in each axis.
- Aspect Ratio – if true, the same scale factor will be used for all axes.

##### 4.10.2.2.14Multi Shape

- Scale - changes the object’s size by the specified factor in each axis.
- Aspect Ratio – if true, the same scale factor will be used for all axes.

##### 4.10.2.2.15Points

- Scale - changes the object’s size by the specified factor in each axis.
- Aspect Ratio – if true, the same scale factor will be used for all axes.

### 4.10.3 Extended Object Descriptions

#### *4.10.3.1 DXF*

DXFs are imported as a series of vectors from the original DXF. The File Explorer will then open so that a DXF file can be selected. After selecting a DXF file, the import preferences are shown (Figure 4-65).

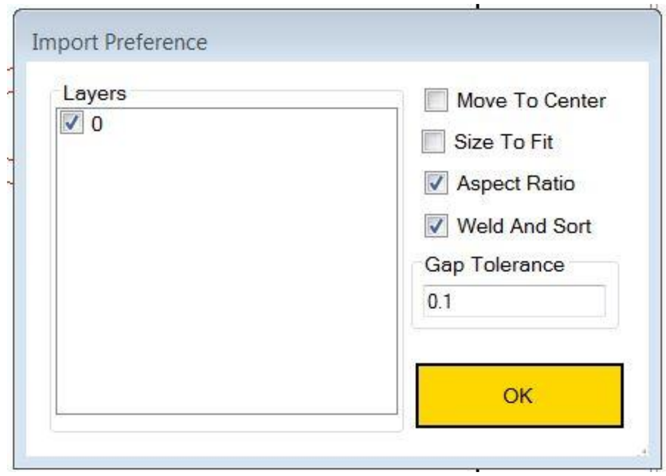



*Figure 4-65 - DXF Import Preferences*

There are six sections of the Import Preferences:

- On the left of the Import Preferences window, there is a list of all layers present in the DXF. All selected layers will be imported into IPGScan.
- The “Move To Center” checkbox will set the imported object’s X and Y Center so that the imported object will appear at the IPGScan origin.
- The “Size To Fit” checkbox will adjust the imported object’s Scale value so the imported object will be the maximum size for the canvas size.
- If the “Aspect Ratio” checkbox is checked, the Aspect Ratio of the original DXF will be maintained while calculating “Size To Fit.”
- “Weld and Sort” joins potentially “broken” vectors and makes laser output more consistent throughout the DXF object. Any vectors that are disconnected by less than the specified “Gap Tolerance” will be joined so that the laser does not turn off and then on again. This feature also changes the direction of all vectors to be continuous. This feature is generally intended for welding objects in order to prevent the laser from starting and stopping throughout the weld object.

The order of the vectors in a DXF object is the same as the order of vectors in the original DXF file. To adjust this order after importing, convert the DXF object into a Multi Shape object. You are not able to convert objects back to a DXF type.

### *4.10.3.2 STL and STEP*

3D CAD models can be imported into IPGScan from STL or STEP files. These models can be used in IPGScan as reference for process objects.

#### 4.10.3.2.1 Transparency

STL and STEP objects will be opaque when imported. The Transparency property will change the transparency of the displayed object. For the Transparency property, a value of 0 is 0% transparent; a value of 255 is 100% transparent (See Figure 4-66, Figure 4-67, and Figure 4-68).

Figure 4-66 - STL Transparency 0


Figure 4-67 - STL Transparency 127


Figure 4-68 - STL Transparency 255

#### 4.10.3.2.2 Color

The Color property changes the display color of the STL object or STEP object within IPGScan. Each imported STL object has a different color to help distinguish different objects. The color can be changed by entering a different RGB value or selecting from a list. An example job containing two STL objects with different colors is shown below.

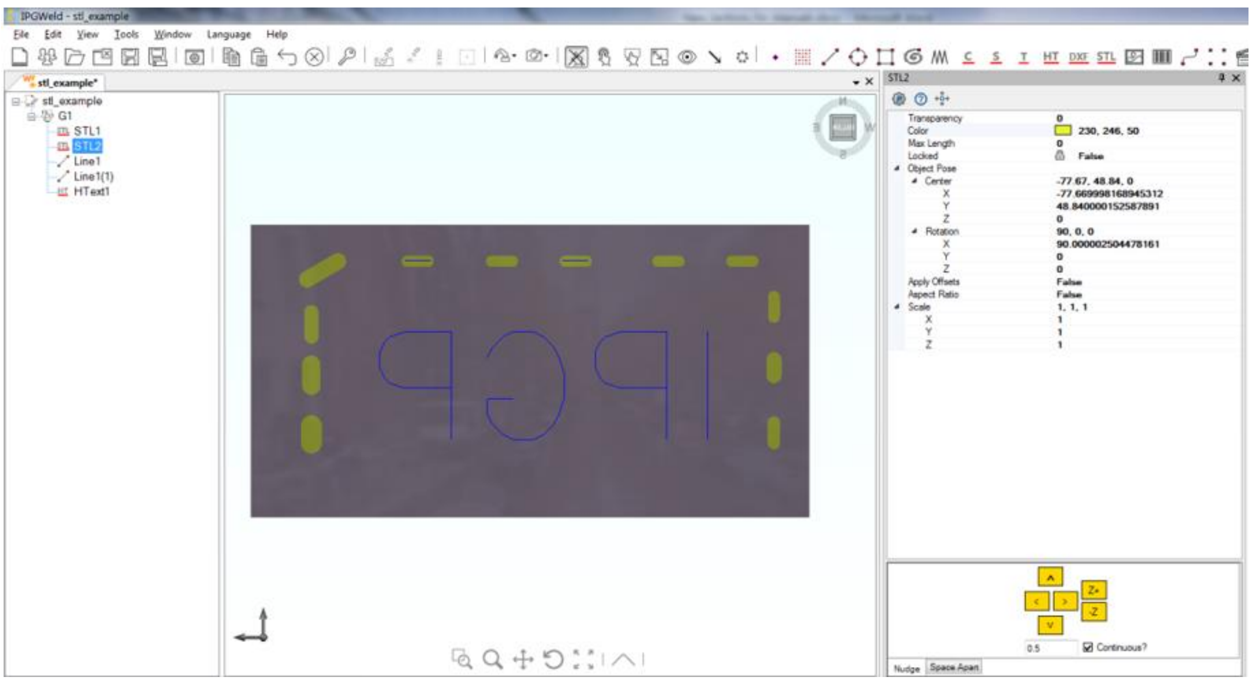



*Figure 4-69 - Two STL Objects with Different Colors*

### *4.10.3.3 Multi Shape*

The Multi Shape feature gives the user the ability to create freeform shapes and edit the vectors of existing shapes, including DXFs.

#### 4.10.3.3.1 Creating Freeform Shapes

To create a freeform shape with Multi Shape, add a Multi Shape object to a job from either the Tools menu or the toolbar . The Multi Shape window opens as shown in Figure 4-70.

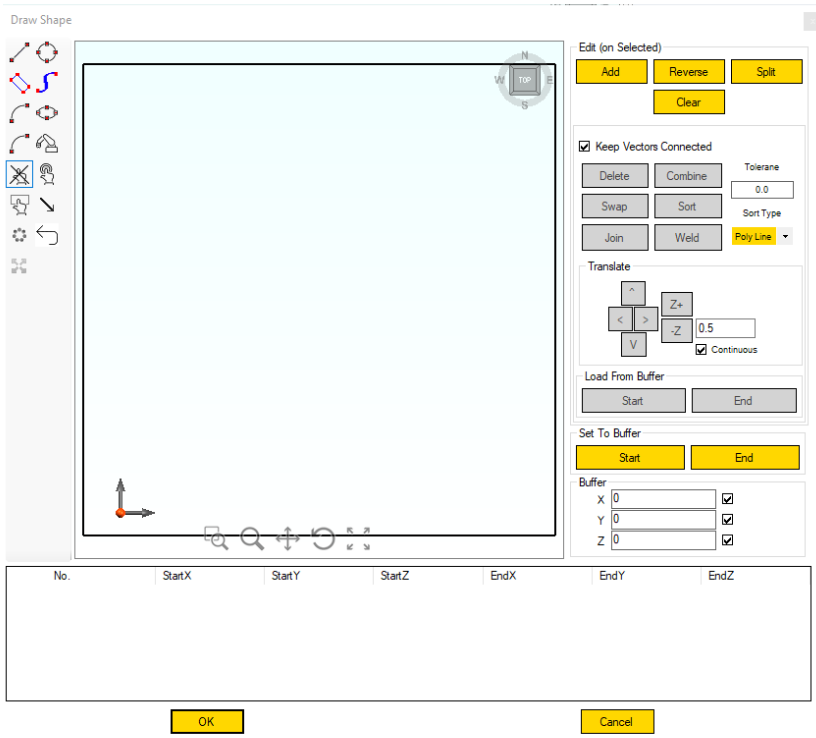
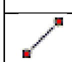
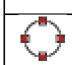
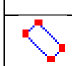
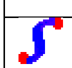
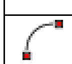
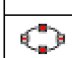
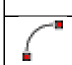
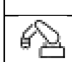
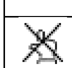
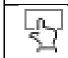



*Figure 4-70 - New Multi Shape Window*

The icons in the left toolbar are used to create, view, or move shapes. Table 4-9 describes each tool.

*Table 4-9 - Multi Shape Tools*

| Icon | Shape | Description |
|---|---|---|
|  | Line | Create a single vector based on a starting point and an ending point. |
|  | Circle | Create a circle based on a center point and a radius. |
|  | Polyline | Create a series of vectors based on a starting point and connecting points. |
|  | Spline | Create a curved line based on a starting point and connecting points. |
|  | Arc | Create a circular arc based on a starting point, a radius, and a sweep angle. |
|  | Ellipse | Create an ellipse based on a center point, the endpoint of the semi-major axis, and the endpoint of the semi-minor axis. |
|  | Elliptical Arc | Create an elliptical arc based on a center point, the endpoint of the semi-major axis, the endpoint of the semi-minor axis, and a sweep angle. |
|  | Load From Robot Trajectory | When the job has a robot trajectory imported, the trajectory is loaded into the Multi Shape editor as a series of vectors. |
|  | No Select | Turn off selection by mouse. |

| | | |
|---|---|---|
|  | Select By Pick | Select vectors with the mouse. |
|  | Select By Rectangle | Select vectors with a reference rectangle drawn by dragging the mouse. |
|  | Show Curve Direction | Turn on arrows at the end of the vectors to show the execution direction. |
|  | Show Vertices | Turn on dots at the vertices of all vectors. |
|  | Undo | Undo the last action. |
|  | Translate | Move a vector by dragging the mouse. Requires the “Keep Vectors Connected” checkbox to be unchecked. |

**Note**: Use the left mouse button to draw each shape and the right mouse button to stop.

##### 4.10.3.3.2 Converting an Object to a Multi Shape

Any processing object or objects can be converted into a Multi Shape object for additional control. To convert a processing object:

1. Select desired object(s) in the job tree.
    a. When multiple objects are selected and converted they are combined into a single Multi Shape.
2. In the right-click menu, select “Convert to Multi Shape.”

Converted objects will no longer have their previous processing parameters. For example, a circle converted into a Multi Shape, will no longer have a “Radius” property. Any changes to a converted shape must be done from within the Multi Shape window.

**Note:** Multi Shape objects cannot be converted back into standard processing objects.

##### 4.10.3.3.3 Editing Vectors in a Multi Shape

The Multi Shape window can also be used to edit the vectors of any Multi Shape object. There are additional tools within the Multi Shape window for control of vectors besides those in the primary IPGScan window. These controls are on the right side of the Multi Shape window and in the table at the bottom of the Multi Shape window.

###### 4.10.3.3.3.1 Unconnected Editing

There are several editing options which could be used to change the current Multi Shape to an unconnected shape. These options are locked with the “Keep Vectors Connected” checkbox. When checked the “Keep Vectors Connected” will keep all vectors in the multi-shape connected. All end points of a vector will connect with another vector. When disabled, the end points of vectors do not have to remain connected.

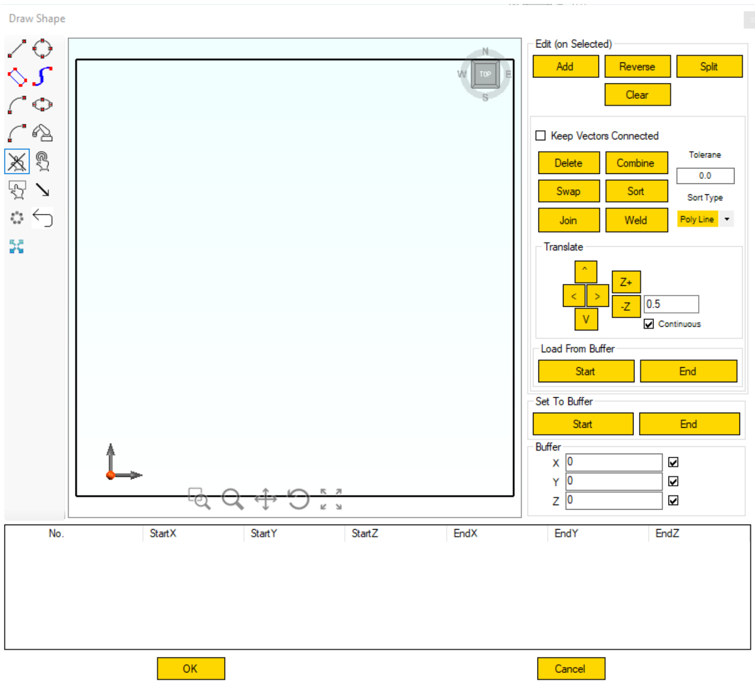



*Figure 4-71 - Multi Shape Window with Keep Vectors Connected Unselected*

### 4.10.3.3.3.2 Vector Data Table

At the bottom of the Multi Shape window is a table containing the data for all vectors in the current Multi Shape. The vectors are listed in their execution order. Figure 4-72 shows the Multi Shape window with the vector data table for a square.

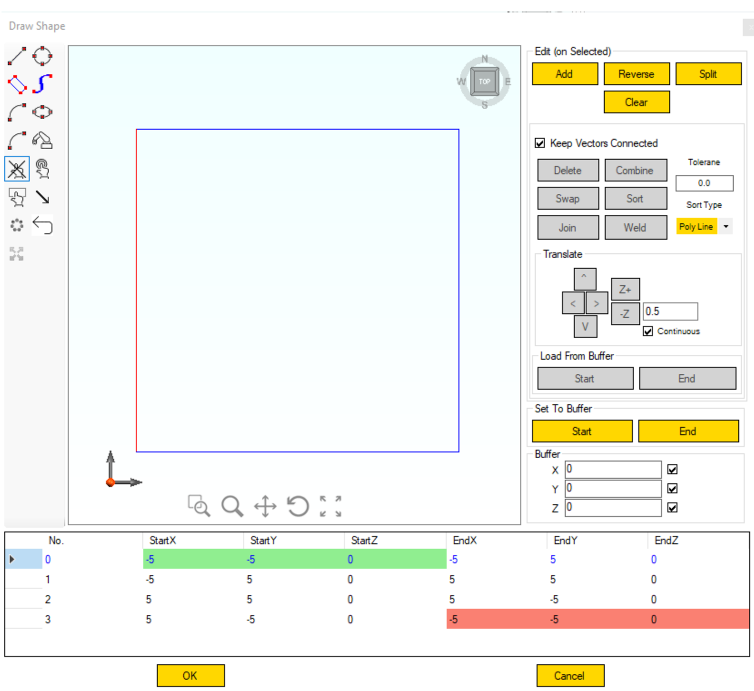



*Figure 4-72 - Multi Shape Vector Data Table*

The columns StartX, StartY, and StartZ represent the X, Y, and Z position of the starting point of each vector, respectively. The columns EndX, EndY, and EndZ represent the same information for the ending point of each vector.

Any vector can be edited by clicking on the desired cell of an already selected vector. If the "Keep Vectors Connected" checkbox is checked, any edited point will change the start/end pair with the previous or next vector.

In the vector data table, the span between a green start point and a red end point represents continuous laser operation. When the scanner will jump to an unconnected position, a new red/green break will be added. Compared to the continuous shape in Figure 4-72 above, the unconnected shape in Figure 4-73 below has two sections of green and red vectors.

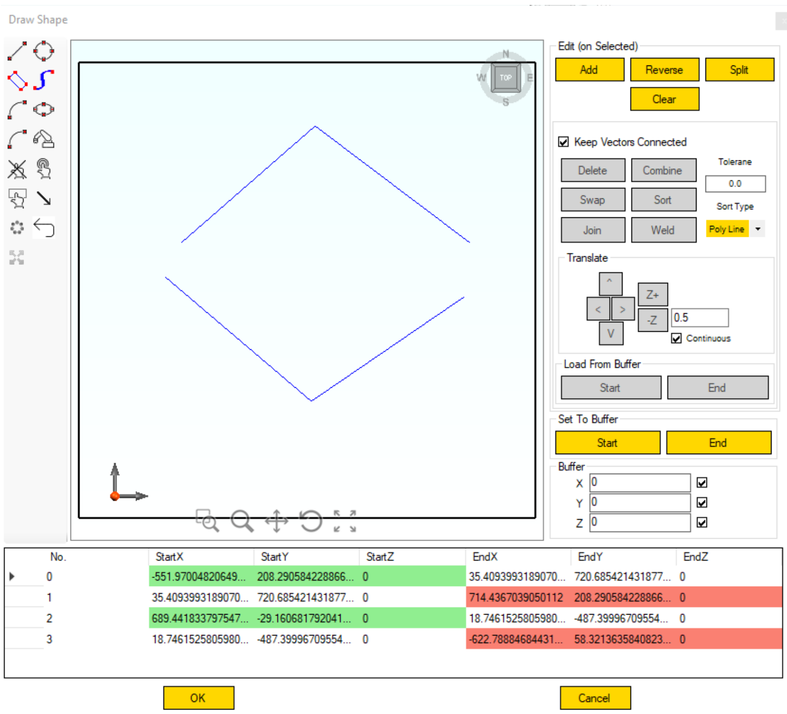



*Figure 4-73 - Multi Shape Disconnected Vectors*

### 4.10.3.3.3.3 Add

Select a vector and click "Add". This will insert a new vector of the same magnitude directly above the selected vector in the vector list. The new vector will be offset by +10 in each of the X and Y directions.

### 4.10.3.3.3.4 Reverse

Select a vector and click "Reverse" to switch the start point and the end point of the selected vector.

### 4.10.3.3.3.5 Split

Select a vector and click "Split" to turn the single vector into two vectors of equal magnitude. The start point of the first vector will be the start point of the original vector. The end point of the first vector and the start point of the second vector will be the midpoint of the original vector. The end point of the second vector will be the end point of the original vector. The figure below shows the same shape as Figure 4-74 after the first vector is split.

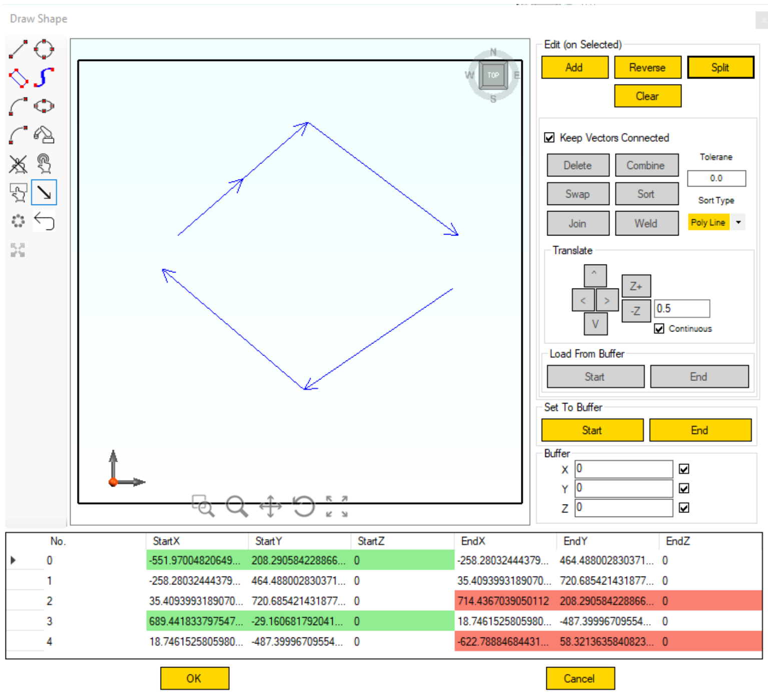


*Figure 4-74 - Multi Shape Split Vector*

### 4.10.3.3.3.6  Clear

The “Clear” button removes all of the vectors from the current Multi Shape.

### 4.10.3.3.3.7  Delete

The “Delete” button removes the selected vector. No other vectors are changed because of a deleted vector.

### 4.10.3.3.3.8  Combine

Select multiple vectors and click “Combine” to combine all vectors into a new vector. All selected vectors are removed and the new vector starts at the first vector’s starting point and ends at the second vector’s ending point. Figure 4-75 shows the selection of multiple vectors. Figure 4-76 shows the Multi Shape after the selected vectors were combined.

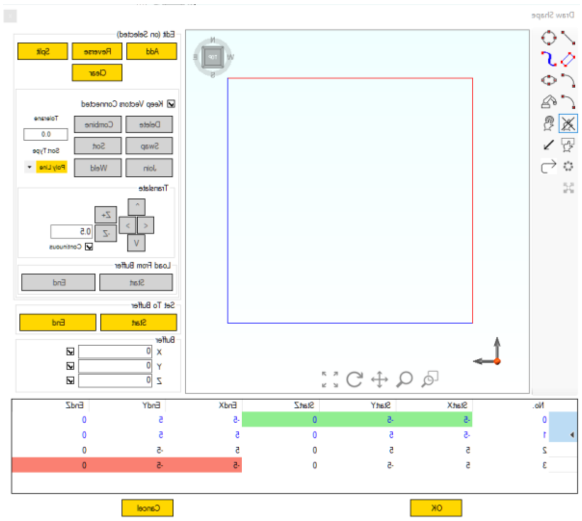


Figure 4-75 - Multi Shape Before Combine

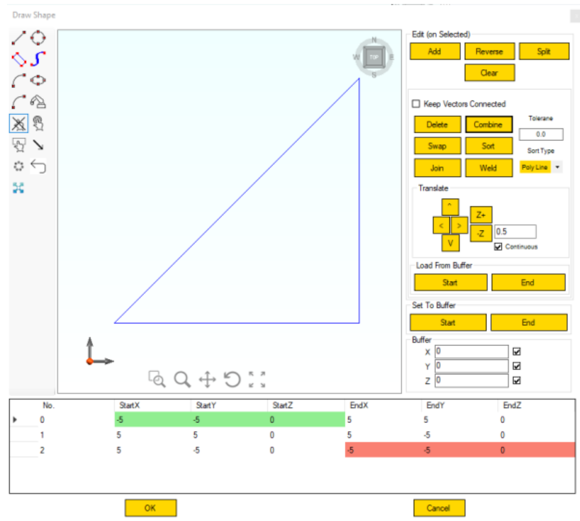



Figure 4-76 - Multi Shape After Combine

## 4.10.3.3.3.9 Swap

Select two vectors and click “Swap" to swap their order in the Vector Data Table. It does not change the position of the vectors.

## 4.10.3.3.3.10 Join

Select two vectors and click “Join” for the vectors to be joined tip-to-tail at the shortest distance. This operation only works on two vectors at a time. Figure 4-77 shows a Multi Shape before joining. Figure 4-78 shows the same Multi Shape where the “Join” button was used to close the opening in the shape.

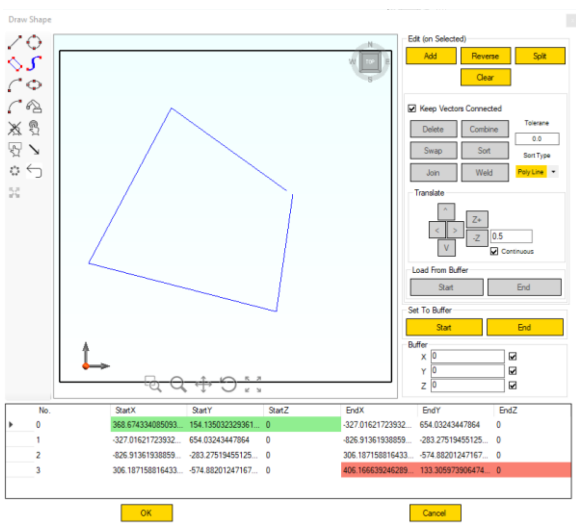



Figure 4-77 - Multi Shape Before Join

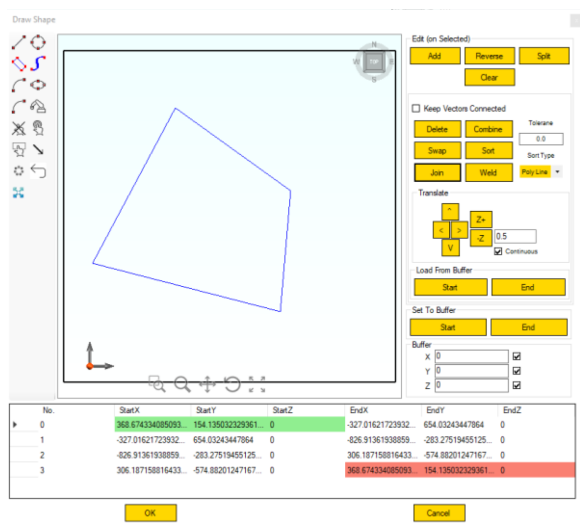



Figure 4-78 - Multi Shape After Join

## 4.10.3.3.3.11 Sort

The “Sort” button sorts the vectors of the current Multi Shape based on the algorithm in the “Sort Type” drop down.

## 4.10.3.3.3.12 Weld

The “Weld” button will connect vectors separated by less than the distance specified in the “Tolerance” box.

##### 4.10.3.3.3.13 Translate

The “Translate” group functions in the same manner as the Nudge tool. The “Translate” group operates on an entire vector compared to the start or the end point.

##### 4.10.3.3.3.14 Set From Buffer

The “Set From Buffer” group acts as a paste operation for an entire point. This group of buttons will set the values of the start or end point of the selected vector to the checked values in the buffer. The start or end point will be changed depending on which button is selected, “Start” or “End”.

##### 4.10.3.3.3.15 Load From Buffer

The “Load From Buffer” group acts as a copy operation for an entire point. This group of buttons takes the values from the start or end point of the selected vector and places these values into the checked values in the buffer. The buffer is set from either the start or end point depending on which button is selected, “Start” or “End”.

##### 4.10.3.3.3.16 Buffer

The buffer acts like a clipboard for a single point. Values can be manually edited or loaded from the “Load From Buffer” group. If a buffer value is unchecked, then it will not be used for “Set From Buffer” or “Load From Buffer” operations.

### *4.10.3.4 Points*

After creating a new Points object, a new window will open to create a collection of point processing objects (see Figure 4-79). Existing objects can also be converted to an editable Points object like a Multi Shape (see Section 4.10.3.3.2).

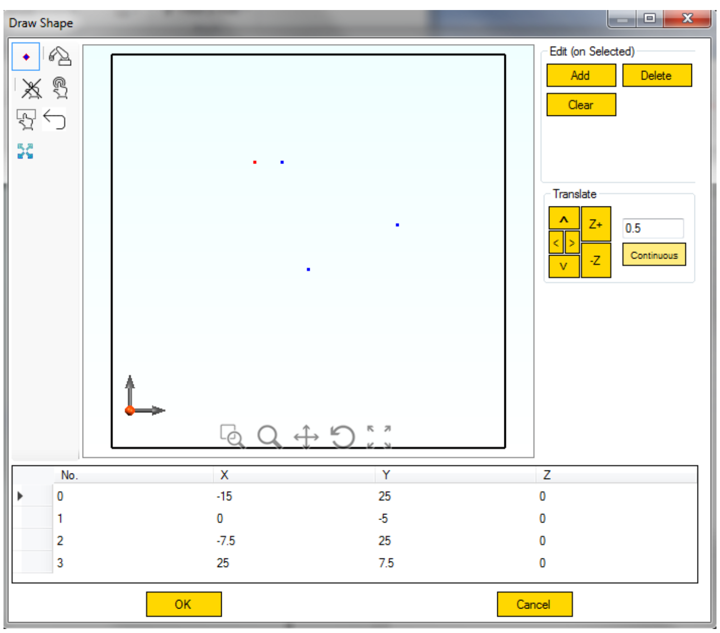



*Figure 4-79 - New Points Window*

#### 4.10.3.4.1 Points GUI

Table 4-10 describes the tools available on the left side of the Points window. Table 4-11 describes the buttons available on the right side of the Points window.

In the “Translate” group on the right side of the Points window there is a group of buttons to move selected point(s). It works the same way as the Nudge tool.

At the bottom of the Points window is the Points Table. The table shows the points in the current Points object in their execution order and with their coordinate positions.

*Table 4-10 - Points Tools*

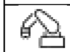
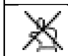
| Icon | Shape | Description |
|---|---|---|
|  | Point | Add a new point to the Points object. |
|  | Load From Robot Trajectory | When the job has a robot trajectory imported, the trajectory is loaded into the Points editor as a series of points. |
|  | No Select | Turn off selection by mouse. |
|  | Select By Pick | Select individual points with the mouse. |
|  | Select By Rectangle | Select a group of points with a reference rectangle drawn by dragging the mouse. |
|  | Undo | Undo the last action. |
|  | Translate | Move a point by dragging the mouse. |

*Table 4-11 - Points Buttons*

| Button | Description |
|---|---|
| Add | Creates a new point at a position offset from the currently selected point of (10, 10, 0). |
| Delete | Deletes the currently selected point. |
| Clear | Deletes all points in the current Points object. |

## 4.11 Process Properties

Process properties are the properties of a job or an object which specify the laser settings for an object’s vectors or points. IPGScan jobs can be a welding job type, cleaning job type, or marking job type. The job type changes the process properties available because different control of the laser is needed for each process.

### 4.11.1 Laser Optimization

Laser Optimization is an option in all IPGScan jobs starting with version 1.0.0.7176 and ScanPack version 0.1.7280. It is found in the Parameters Window when the job is selected in the Job Tree.

When Laser Optimization is disabled, the laser will be enabled and disabled at the start and end of each object, respectively. Also, the galvos will return to position (0,0,0) in between each processing object. This is the default operation for IPGScan.

When Laser Optimization is enabled, the laser will remain enabled throughout the duration of the job. Also, the galvos will only move as required for the job. The laser will be disabled and re-enabled when switching beam types.

The additional laser operations add tens of milliseconds to the overall job cycle time, depending on the length of the job. Enabling Laser Optimization will decrease the cycle time of the job. However, in the event of a fatal error or malfunction in the controller, using laser optimization increases the chances that the laser will remain on.

### 4.11.2 Welding Process Properties

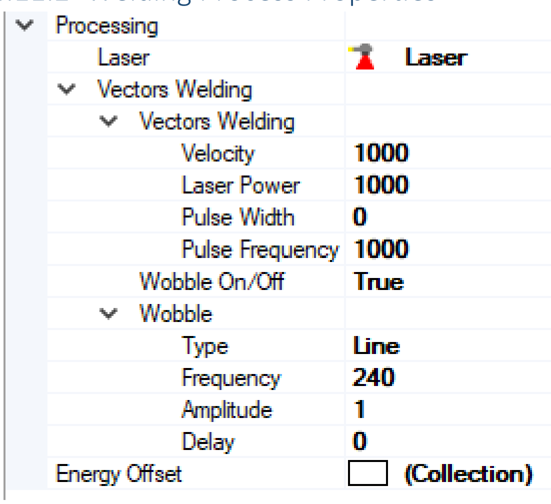



*Figure 4-80 - Welding Vectors Process Properties*

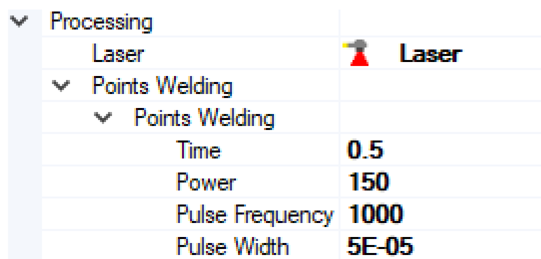



*Figure 4-81 - Welding Points Process Properties*

Figure 4-80 and Figure 4-81 shows an example of the process properties for a vector object and point object, respectfully, in a welding IPGScan job type. Table 4-12 and Table 4-13 show the properties for welding vector objects and welding point objects, respectively.

*Table 4-12 - Welding Vector Properties*

| **Property** | **Description** |
|---|---|
| Laser | type of laser that will be used to output the selected object. |
| Velocity | speed of the focal point of the beam |
| Laser Power | power of the laser |
| Pulse Width | Width of each laser pulse (0 if using CW laser) |
| Pulse Frequency | Frequency of the laser pulses (ignored if using CW laser) |
| Wobble On/Off | True if using wobble, false if not using wobble |
| Type | Type of the wobble |
| Frequency | Frequency of the wobble |
| Amplitude | Amplitude of the wobble |
| Delay | Delay of the wobble |
| Energy Offset | See Section 4.11.2.2 |

*Table 4-13 - Welding Point Properties*

| Property | Description |
| --- | --- |
| Laser | type of laser that will be used to output the selected object. |
| Time | Total length of each point |
| Power | Power of the laser for each point |
| Pulse | |
| Pulse Width | Width of each laser pulse (0 if using CW laser) |
| Pulse Frequency | Frequency of the laser pulses (ignored if using CW laser) |

##### 4.11.2.1 Wobble

The Wobble feature allows users to add a wobble to the weld seam. The wobble causes the TCP speed of the beam to speed up, but still allowing the weld feature to maintain a commanded linear speed. The wobble will never change the amount of time it takes to finish a weld. Refer to Table 4-14 for all wobble types. Table 4-15 discusses the different configuration parameters of a wobble.

*Table 4-14 - Wobble Types*


*Figure 4-82 - Circle Wobble*


*Figure 4-83 - Line Wobble*

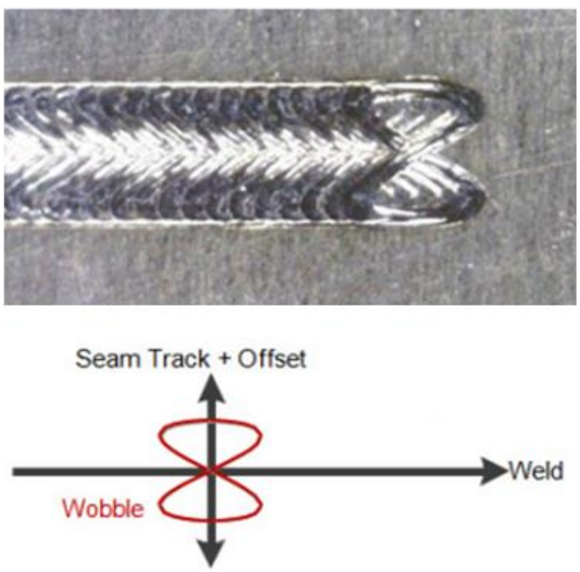



*Figure 4-84 - Eight Wobble*

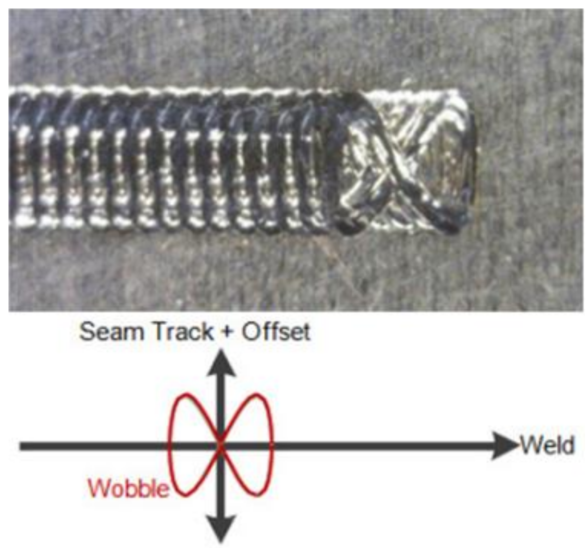



*Figure 4-85 - Infinity Wobble*

*Table 4-15 - Wobble Parameters*

| **Property** | **Description** |
|---|---|
| Amplitude | This value specifies the width of the wobble. For example, if a circle was used then the amplitude would be the diameter of the wobble. |
| Delay | This value specifies the amount of time it takes wobbling to begin. The use of a negative number causes the wobble to start early. |
| Frequency | This value is the rate at which one complete wobble shape is run. |

### 4.11.2.2 Wobble Power Modulation

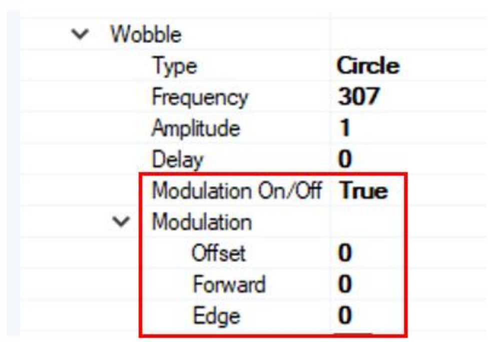
Wobble Power Modulation functionality is a compensation on the laser power control signal on each wobble cycle. The purpose behind this is to decrease or increase the laser power in areas where there may be an overlap of laser energy due to the wobble motion.

Three parameters are available (along with all other standard wobble parameters): Edge, Offset, and Forward. Additional details on each are provided below.

The Wobble Power Modulation is part of the wobble parameter section in IPGScan (when IPGScan is working in weld mode) and is not enabled by default.

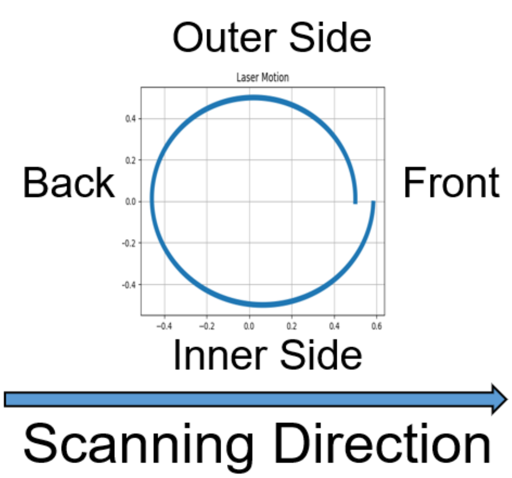
To enable it, wobble must be enabled for a particular weld, a wobble type selected, and “Modulation On/Off” must be set to True. See Figure 4-86.

*Figure 4-86 Wobble Power Modulation Parameters*




To better understand the given parameters uses can refer to Figure 4-87 for a definition of the beam movement during a wobble cycle.

*Figure 4-87 Definition of Wobble Cycle Terms*



#### 4.11.2.2.1 Edge Parameter

The Edge parameter can be set anywhere from 0 to 1. This parameter reduces the power at the sides of the wobble cycle (by the same amount).

For example, a value of 0.1 reduces the power on each side by 10%. A value of 1.0 would reduce the power by 100%. See Figure 4-88 and Figure 4-89.

*Figure 4-88 Heatmap of Wobble Weld with Different Edge Parameter Values*

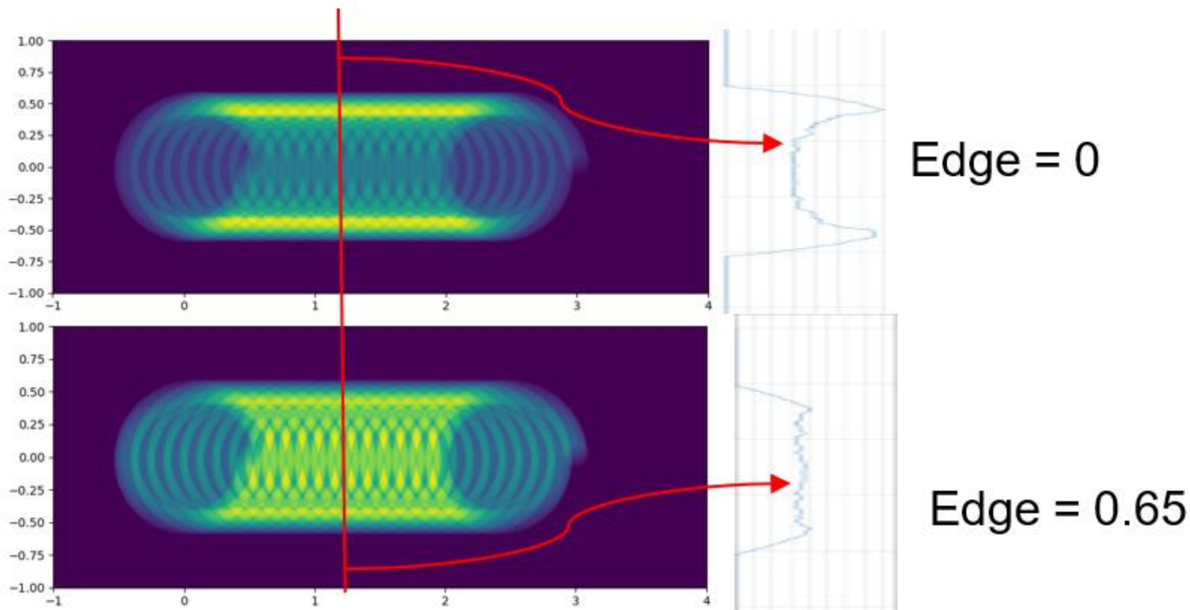
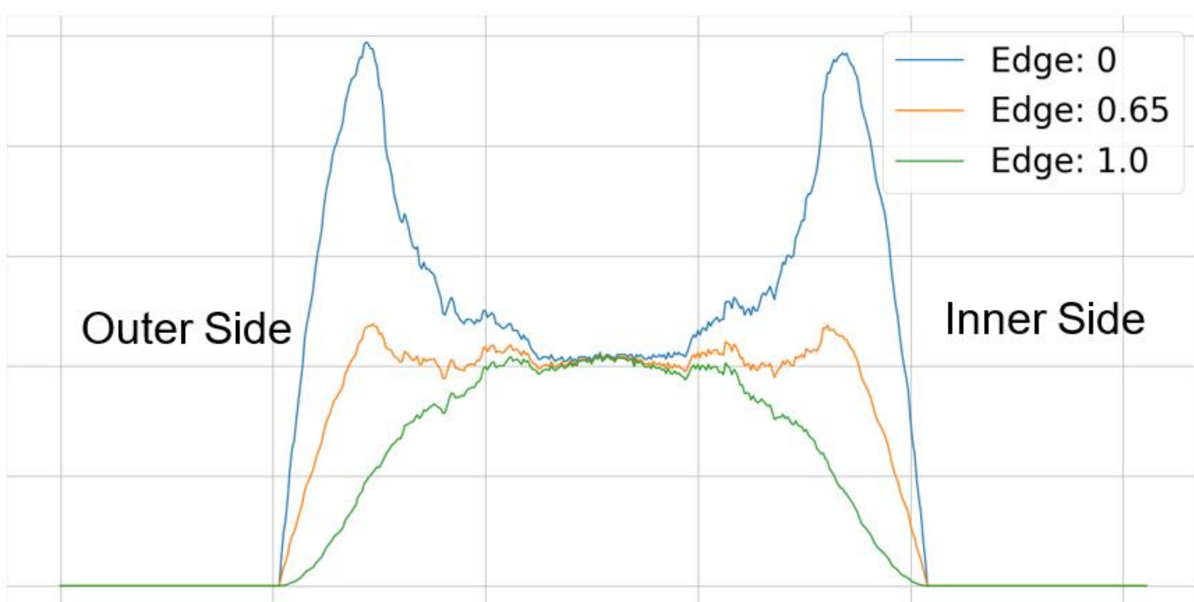



*Figure 4-89 Simulated Cross Section of Weld with Different Edge Values*



#### 4.11.2.2.2 Offset Parameter

The Offset parameter can be set anywhere from -1 to 1. This parameter reduces the power at one side of the wobble cycle, while increasing on the other. Adding a negative sign reverses the side power is decreased/increased.

For example a value of 0.1 (or -0.1) increases the power on one side by 10%, while decreasing the other by 10%. See Figure 4-90 and Figure 4-91.

*Figure 4-90 Heatmap of Wobble Weld with Different Offset Parameter Values*

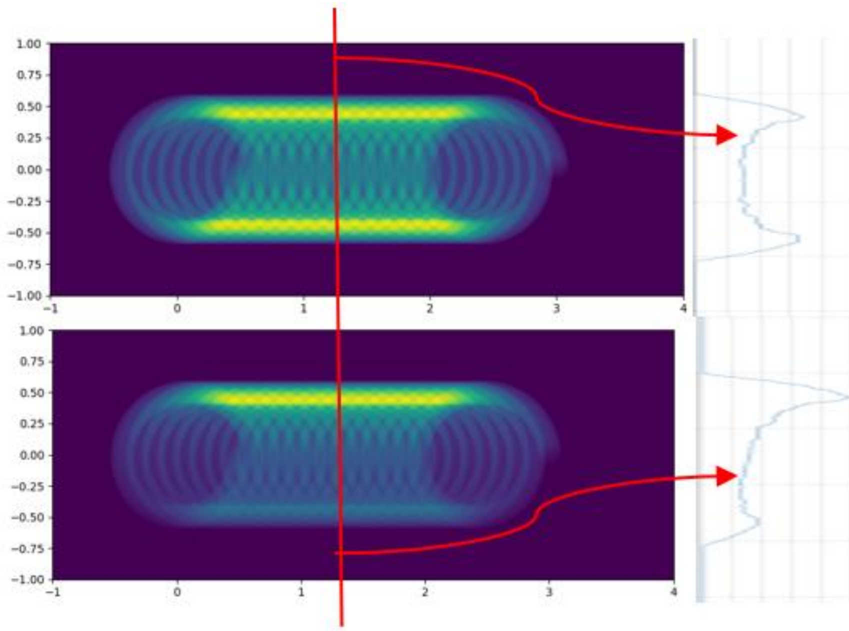
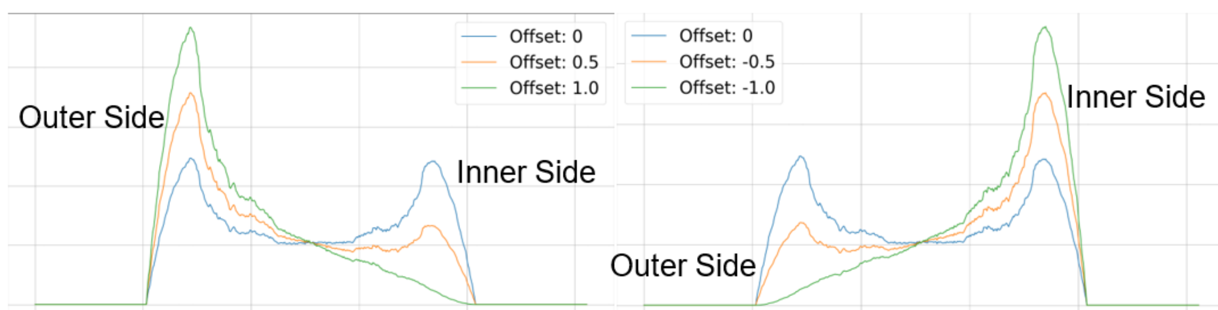



*Figure 4-91 Simulated Cross Section of Weld with Different Offset Values*




#### 4.11.2.2.3 Forward Parameter

The Forward parameter can be set anywhere from -1 to 1. This parameter reduces the power at the back of the wobble cycle, while increasing on the front (or vice versa depending on the sign). Adding a negative sign reverses the side power if decreased/increased.

For example, a value of 0.1 (or -0.1) increases the power on the front by 10%, while decreasing the back by 10%. See Figure 4-92 and Figure 4-95.

*Figure 4-92 Heatmap of Wobble Weld with Different Forward Parameter Values*

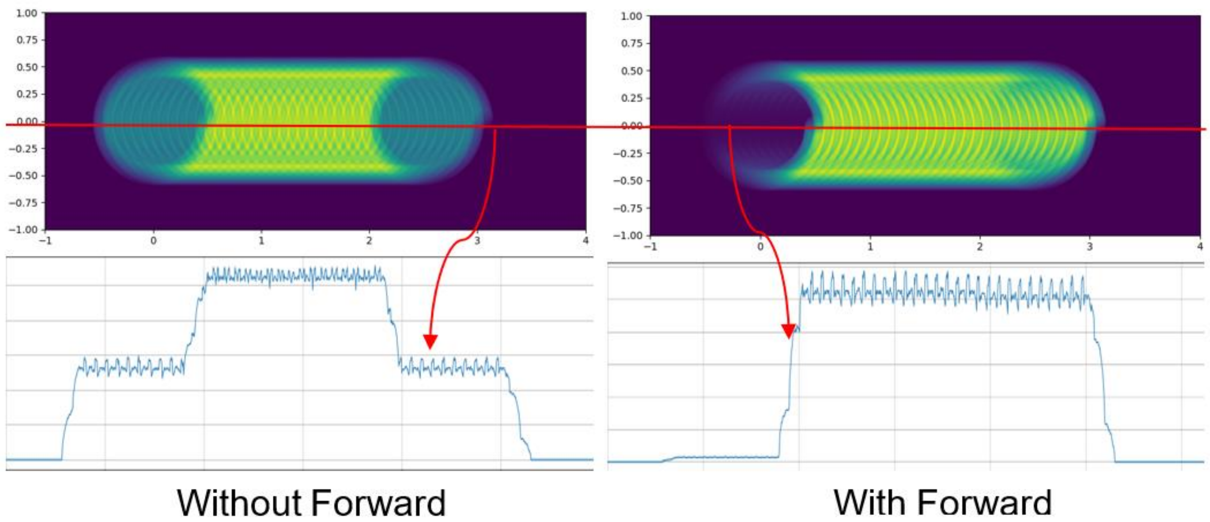
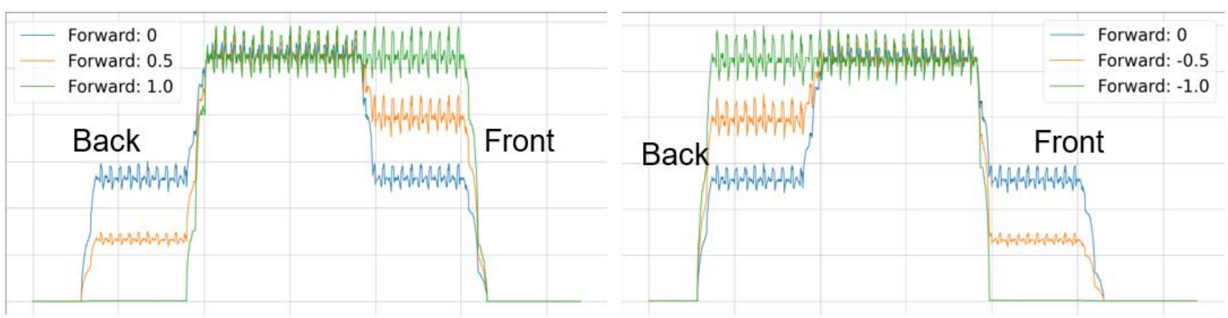



*Figure 4-93 Simulated Cross Section of Weld with Different Forward Values*




### *4.11.2.3 Energy Offset*

The Energy Offset feature allows users to ramp up or down the lasers power within an object.
The percentage is based on the laser power that is specified in the processing parameters for the object. An energy offset of 0% will result in a laser output of the power specified in the parameters. Figure 4-94 shows the Energy Offset Window.

In the Energy Offset Window, users can set offsets for points by either:

1. selecting points with the “Pick” or “Rectangle” tools and then typing in an offset under “Chart”
2. by dragging points on the lower graph

Users can create a linear progression of power between multiple points using the Ramp functionality. Users specify the starting and ending points in the progression and the values for those points. After clicking “Generate,” a linear progression will be created between the points specified and the values specified.

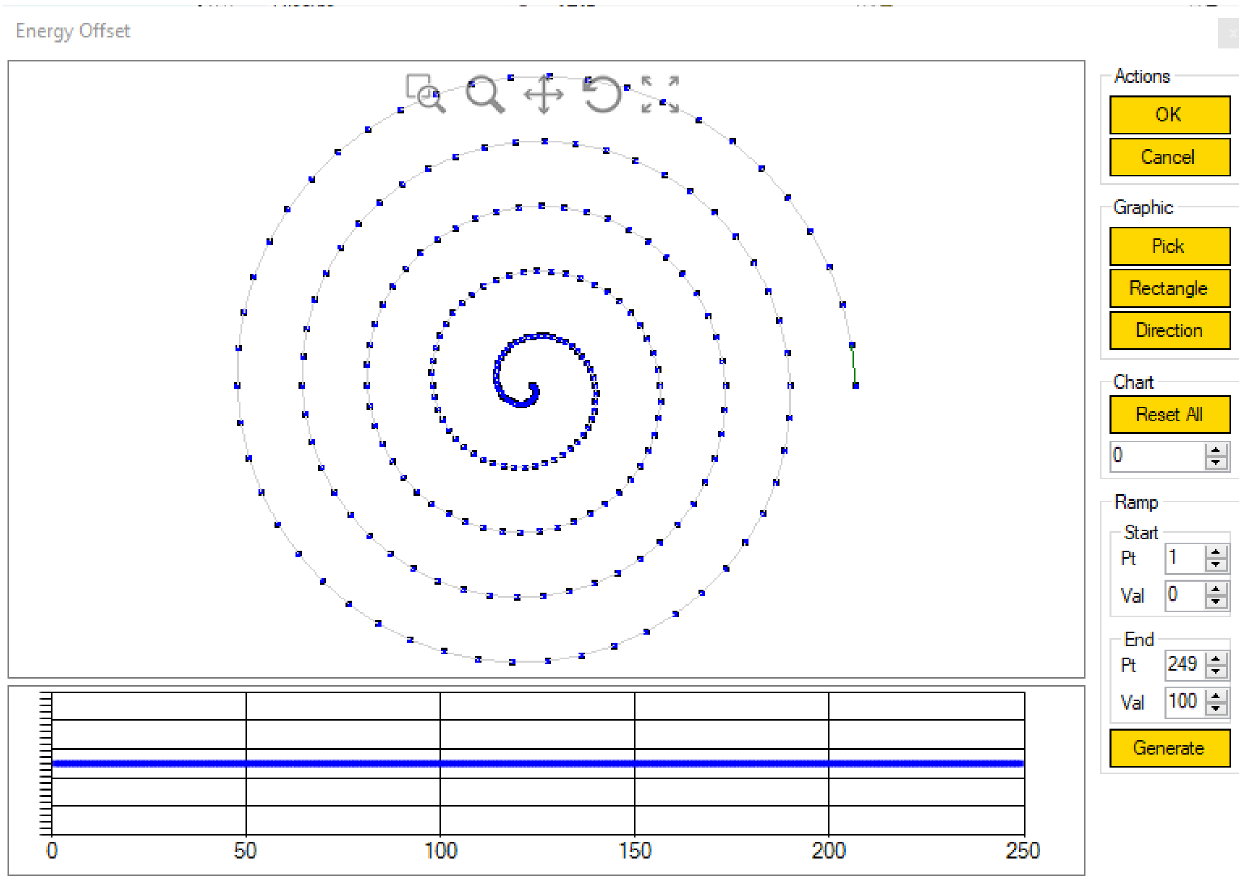



*Figure 4-94 - Energy Offset Window*

### *4.11.2.4 AMB Ring and Power Control*

In order to change the power settings for core and ring separately, the “Enable AMB” setting in IPGScan must be set to true.

**IMPORTANT**

**By default, “Enable AMB” is set to False and only the ring analog signal will be used.**

**IPGScan Version 1.0.0.14937 and ScanPack Version 0.1.14913 or higher are required for AMB ring and core power control.**

**Scan Controllers built after December 2021 should have the proper hardware for this functionality.**

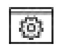
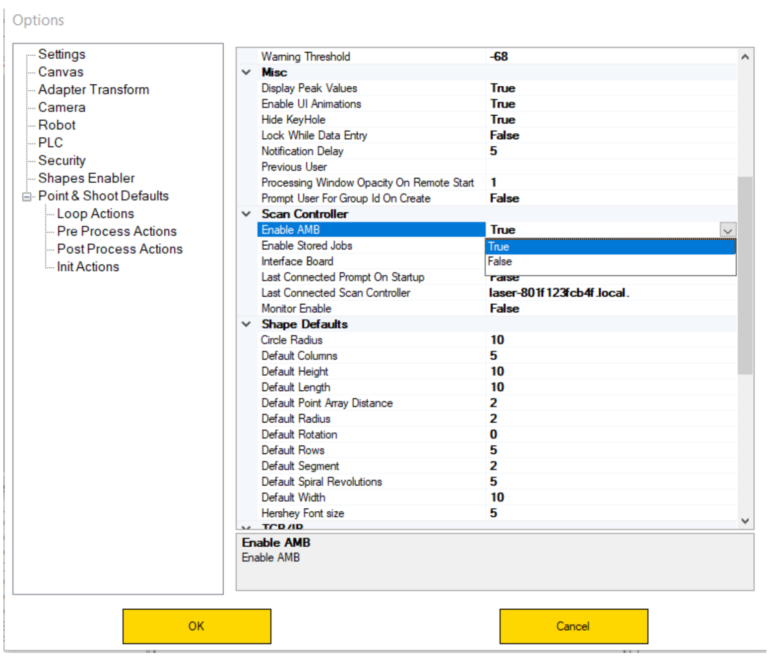
To change the “Enable AMB”, click on the Option button  (View-Options, Alt+O), under Settings->Scan Controller, set “Enable AMB” to True. See Figure 4-95.

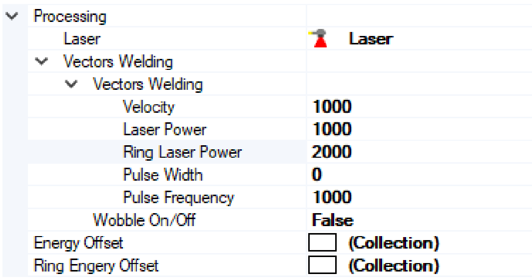
*Figure 4-95 Enabling AMB Power Control*




**IMPORTANT** **Users may need to restart IPGScan for the changes to take effect.**

Once “Enable AMB” is set, two additional field for laser parameters will appear in the object’s property area (see Figure 4-96):

*Figure 4-96 AMB Dual Power Control Properties*




The "Ring Laser Power" sets the laser power value (in watts) for ring beam and Ring Energy Offset opens the Energy Offset profile editor for the ring beam. Different Energy Offset profiles can be created for core and ring beams, however the length of each segment in the profile is common for both (please refer to the IPGScan manual for details on Energy Offset).

### 4.11.3 Marking Process Properties

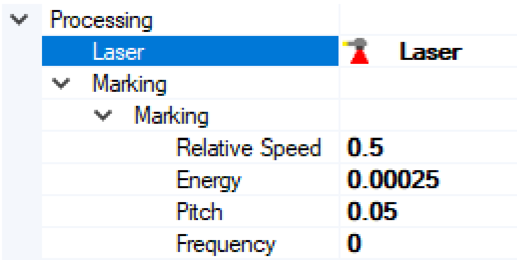



*Figure 4-97 - Marking Vector Process Properties*

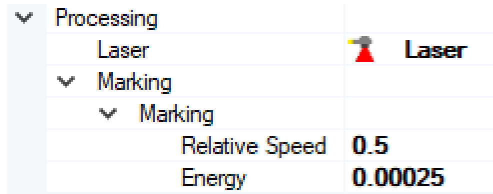



*Figure 4-98 - Marking Point Process Properties*

Figure 4-97 and Figure 4-98 shows an example of the process properties for a vector object and point object, respectfully, in a marking IPGScan job type. Table 4-16 shows the different processing properties of an IPGScan Marking job. When marking points, only Relative Speed and Energy are applicable.

*Table 4-16 - Marking Properties*

| Property | Description |
|---|---|
| Laser | Type of laser that will be used to output the selected object |
| Relative speed | Relative speed of the focal point of the beam from 0 to 1 |
| Energy | Energy of the laser in [J] |
| Frequency | Frequency of the laser |
| Pitch | Distance between laser pulses on the work piece |

### 4.11.4 Cleaning Process Properties

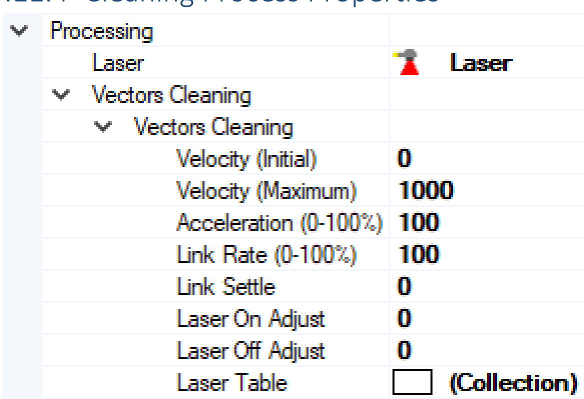



*Figure 4-99 - Cleaning Vector Process Properties*

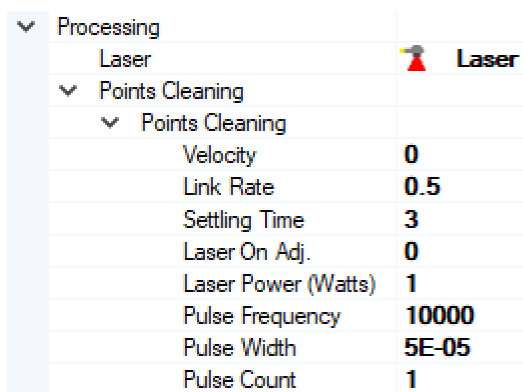



*Figure 4-100 - Cleaning Point Process Properties*

Figure 4-99 and Figure 4-100 shows an example of the process properties for a vector object and point object, respectfully, in a marking IPGScan job type. Table 4-17 and Table 4-18 show the properties for welding vector objects and welding point objects, respectively.

*Table 4-17 - Cleaning Vector Properties*

| Property | Description |
|---|---|
| Laser | Type of laser that will be used to output the selected object |
| Velocity (Initial) | Speed of the laser focal point at the start of the object |
| Velocity (Maximum) | Maximum speed of the laser focal point |
| Acceleration (0-100%) | percentage of the maximum acceleration allowed by the scanner. A lower value increases time between Velocity (Initial) to Velocity (Maximum). |
| Link Rate (0-100%) | Percentage of maximum speed allowed by the scanner for a jump (e.g. when the scanner moves from one point to another). |
| Link Settle | Specifies how long until the scanner settles on a given position in units of scanner time constants. The software will limit the value to a safe maximum. Usually, a value between 3 and 6 should be used. |
| Laser On Adjust | Adjustment on the calculated time between when the scanner reaches the start of a vector/a point and the laser begins to fire. Only positive values are allowed. |
| Laser Off Adjust | Adjustment on the calculated time between when the scanner leaves the end of a vector/a point and the laser begins to fire. Only positive values are allowed. |

*Table 4-18 - Cleaning Point Properties*

| Property | Description |
|---|---|
| Laser | Type of laser that will be used to output the selected object |
| Link Rate | See "Link Rate" in Table 4-17 |
| Settling Time | See "Link Settle" in Table 4-17 |
| Laser On Adj. | See "Link On Adjust" in Table 4-17 |
| Laser Power (Watts) | The output power of the laser in Watts |
| Pulse Frequency | See "Pulse Frequency" in Table 4-13 |
| Pulse Width | See "Pulse Width" in Table 4-13 |

#### 4.11.4.1 Laser Table

During an actual vector scanning, IPGClean has to manage three laser control signals. These signals are:

- Laser Power (in Watts)
- Pulse Frequency (in Hertz), assuming the pulse laser is being controlled
- Pulse Width (in Seconds), assuming the pulse width can be varied

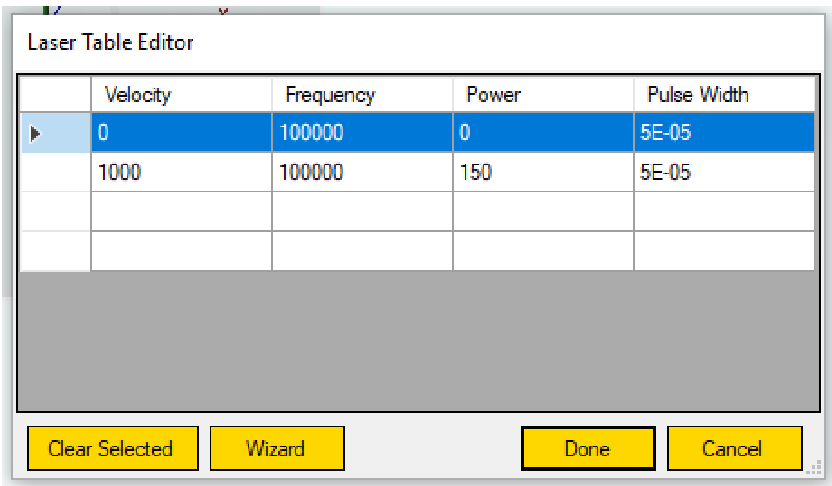
These signals may change with time and they are synchronized with the galvo mirrors movement. All signals above have to specified in terms of the scanning velocity. This is to ensure that either:

- A constant pulse spacing OR
- A constant deposition of energy per linear unit

Is maintained even if the galvos are accelerating. This is achieved by the Laser Table Feature. The Laser Table feature specifies the Power, Frequency, and Pulse Width for specific Velocities.

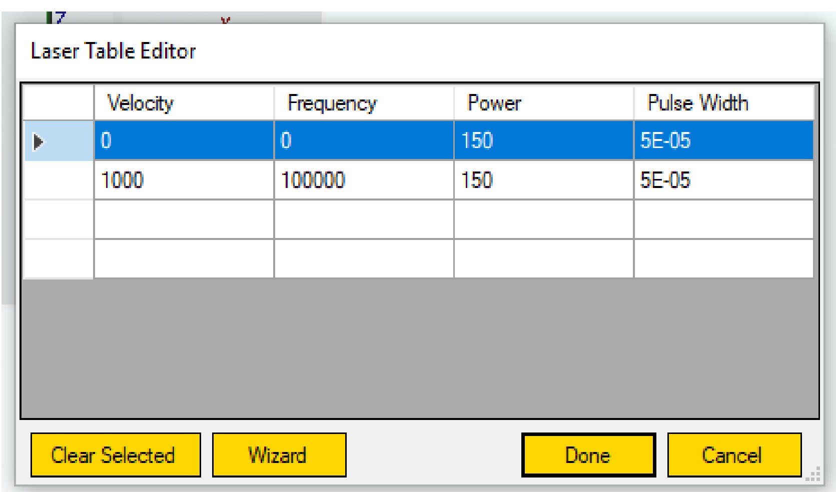
For example, for constant laser energy per linear distance, increase the laser power from 0 to 150W at a constant pulse width (50% duty cycle) and constant frequency (100 kHz). See Figure 4-101.

*Figure 4-101 IPGClean Laser Table Example 1*




For equally spaced pulses increase the pulse frequency from 0kHz to 100kHz at a constant power and pulse width. See Figure 4-102.

*Figure 4-102 IPGClean Laser Table Example 2*



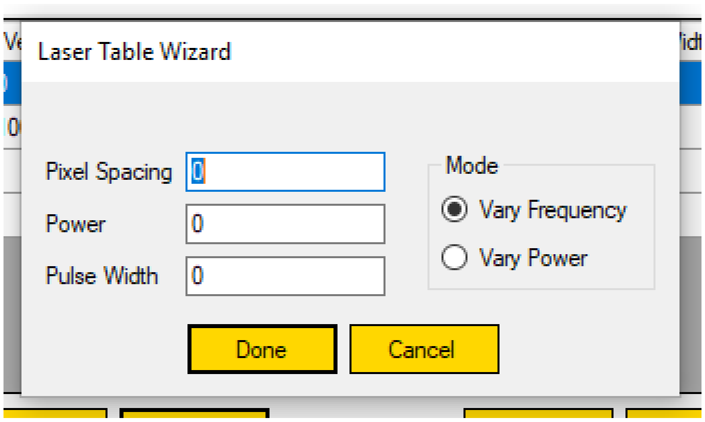

**IMPORTANT** **The pulse distance on the surface being cleaned is simply the raio between velocity and frequency.**

**The initial and maximum velocities may not need to be equal to those on the laser table. The software will interpolate and find correspondent values in the line that passes through all values in the laser table.**

The laser table editor also offers a wizard that populates the laser table based on Maximum Velocity parameter and uses the pulse distance as the input parameter.

Clicking on the Wizard button, the following dialog appears (see Figure 4-103):

*Figure 4-103 IPGClean Laser Table Wizard*




- Vary Frequency - a table will be created in which the pulse spacing is kept constant by varying the frequency as in Example 2 above.
- Vary Power - a table will be created in which the energy per linear distance iskept constant by increasing power with increasing velocity as in Example 1 above.

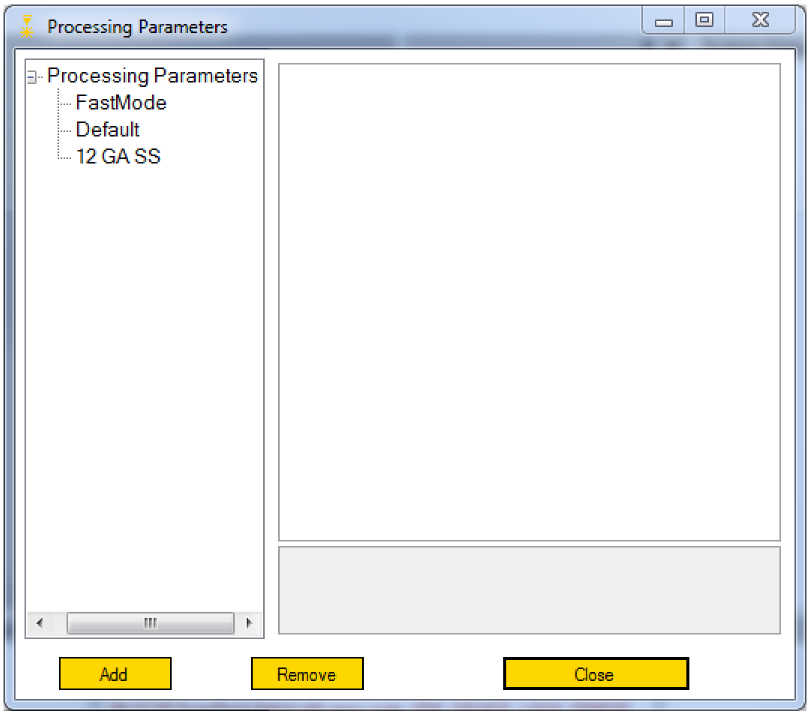
### 4.11.5 Creating a Parameter Profile

Users can create a parameters profile for frequently used materials/parameters.

To create a Parameter Profile:

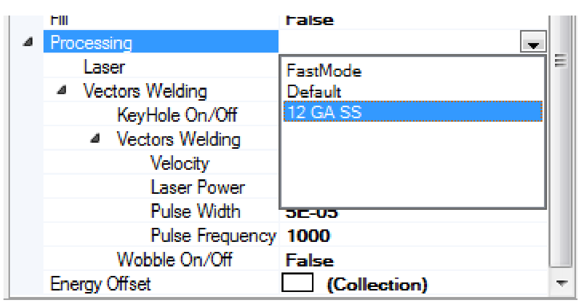
1. Select View → Processing Parameters. See Figure 4-104.

*Figure 4-104 Welding/Marking/Cleaning Parameters*




2. Click “Add” to add a new profile.
3. Rename the profile to a desired custom name as needed.
4. Modify the process parameters as needed.
5. Click Ok to save your changes. Close the Parameters window.
6. In the Data/Parameter Window, after selecting an object, under Processing, select the profile from the drop-down menu as shown in Figure 4-105.

*Figure 4-105 Processing Dropdown Menu*



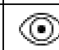
**IMPORTANT** **The fields for the parameters are auto-filled based on the template you selected. Further modifying the shapes parameters after selecting a profile will not change the profile.**

## 4.12 Process Features and Tools

### 4.12.1 Viewing Tools

There are three tools in the toolbar which can be used to adjust the view in IPGScan. These tools are shown in Table 4-19.

*Table 4-19 - Viewing Tools*

| Tool Name | Description | Icon |
|---|---|---|
| Show Selected | Show only the selected object in the job tree. All other objects on the canvas will disappear. Note that this does not affect processing; this is only for assisting the users visually. |  |
| Show Curve Direction | Shows the direction in which object vectors will be output by the scanner. |  |
| Show Vertices | Shows the vertices in all objects. |  |

### 4.12.2 Moving Objects

There are several methods for moving IPGScan objects.

#### *4.12.2.1 Pose Configuration*

A pose is a description of an object’s position and orientation in space. It is comprised of a “Center” or position and a “Rotation” or orientation. The Center is the object’s translation along the X, Y, and Z axes; the Rotation is the object’s rotation about the X, Y, and Z axes.

There are three poses for each object: Object pose, Group pose, and Global pose. The “Object pose” modifies the object’s position relative to the center of the object. The “Group pose” is the pose of the group and is applied to each object within the group. The “Global pose” is only available in On-The-Fly type jobs and modifies the object’s position relative to the IPGScan origin.

#### *4.12.2.2 Nudge Tool*

Above the properties menu, select the nudge button ( ) to display the Nudge Window as seen in Figure 4-106. The radio buttons at the bottom select which pose will be nudged.

When a button is pressed, the specified pose of the selected objects will be adjusted by the number at the bottom in the axis and direction of the button pressed. The “Global” pose is only available in On-The-Fly type jobs.

If the continuous button is selected, a nudge button can be held down and the object will keep being nudged. If the continuous button is not selected, only one nudge will be done per click of a button.

The local button applies the nudge to the selected objects’ object pose using the frame of the object. For example, if the selected object is rotated at a 45° angle and a local nudge is applied parallel to the rotation then the object will translate along the 45° angle.

When the Translate tab is selected, the nudge is applied to the selected poses’ “Center.” When the Rotate tab is selected, the nudge is applied to the selected poses’ “Rotation.”

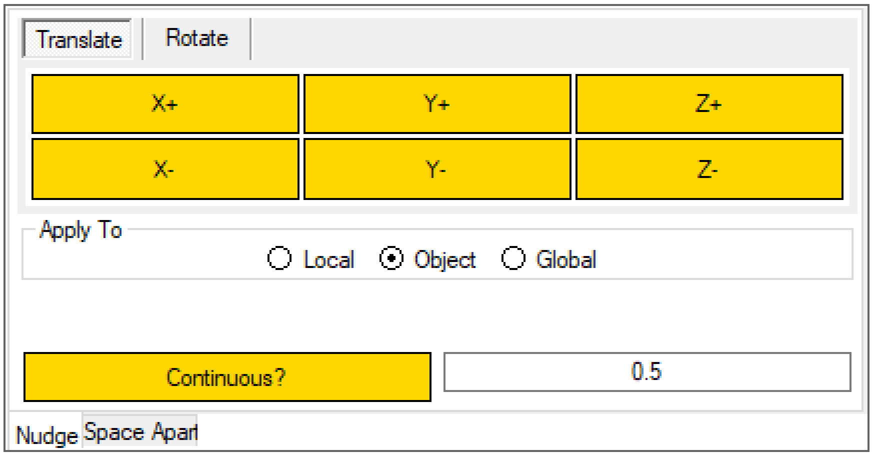



*Figure 4-106 - Nudge Window*

The Space Apart tab will equally space apart multiple objects by the specified amount. This will only apply to translation. Figure 4-107 shows the Space Apart window.

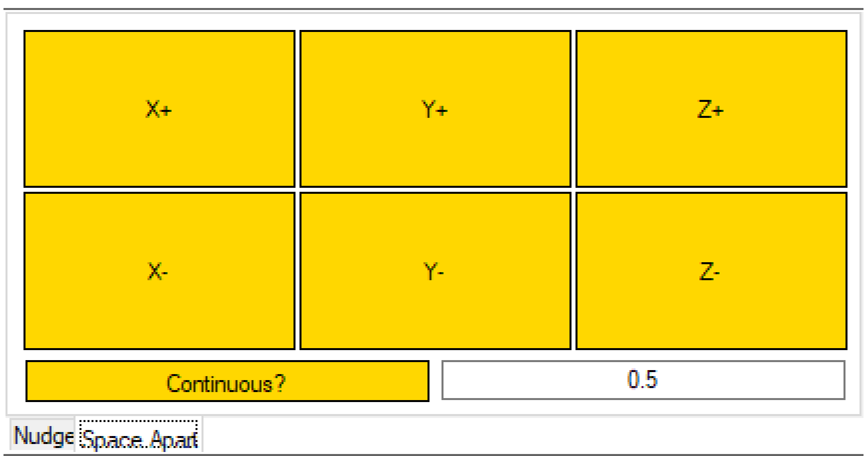



*Figure 4-107 - Space Apart Window*

#### *4.12.2.3 Select with Mouse*

The Select by Pick feature allows users to manually manipulate the coordinates of the object on the canvas. It does not affect the rotation of the object. The button (  ) can be found in the IPGScan toolbar.

The Select by Rectangle feature allows the user to drag and drop the rectangle in the field of view to select multiple objects. The other Move tools can then be used on the set of selected objects. The button (  ) can be found in the IPGScan toolbar.

#### *4.12.2.4 Keyboard Offset*

With keyboard offset, users have the ability to preview objects in the scanner field of view and make adjustments to object location, live time, using the computer keyboard. This feature is designed to assist with placing objects in a work area during process setup.

To use keyboard offset:

1. Select the desired object(s) in the Job Tree.
2. Click Start Processing.
3. Check Guide, Loop, and Offset. Refer to figure Keyboard Offset below.
    a. Fast Mode can also be checked if desired.

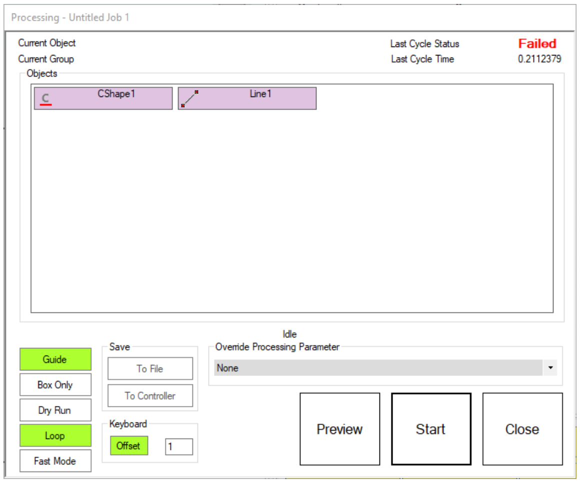



*Figure 4-108 - Keyboard Offset*

Set the desired distance that each keystroke will cause the object(s) to shift. Figure Offset Distance shows an offset distance of 1.0mm per keystroke.


*Figure 4-109 - Offset Distance*

4. Click Start.
    a. It should now be possible to see the object(s) with the guide laser.
5. While observing the guide laser, move the position of the object(s) using the keystrokes outlined in Table Keyboard Offset Keystrokes

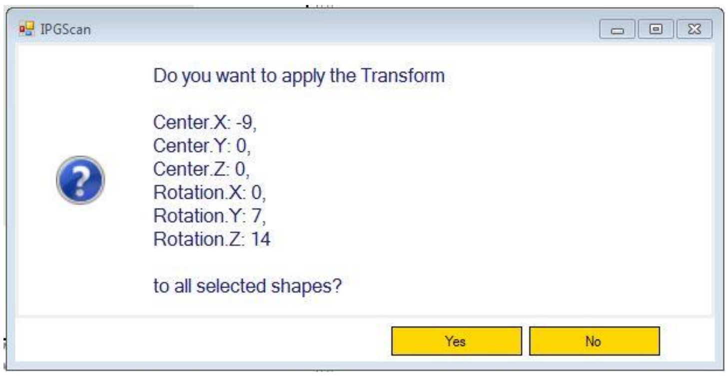
*Table 4-20 - Keyboard Offset Keystrokes*

| IPGScan Direction | Keyboard Key Stroke |
|---|---|
| Translation in X+ / X- | Right Arrow / Left Arrow |
| Translation in Y+ / Y- | Up Arrow / Down Arrow |
| Translation in Z+ / Z- | PgUp / PgDn |
| Rotation about X+ / X- | Ctrl + Right Arrow / Ctrl + Left Arrow |
| Rotation about Y+ / Y- | Ctrl + Up Arrow / Ctrl + Down Arrow |
| Rotation about Z+ / Z- | Ctrl + PgUp / Ctrl + PgDn |

6. Once the object(s) is in the desired location, click Stop.

7. Click Close to close the Processing Window.
8. Click Yes to apply the Transform. Click No to revert the object(s) to the starting location. See Figure 4-110 below.

*Figure 4-110 Confirm Transform*




### 4.12.3 Projection

Projection allows users to project an object onto an STL model. This allows users to contour features according to STL model.

To use Projection:

The following steps detail the method for which users can project processing objects onto an STL model.

1. Open IPGScan.
2. Create a new IPGScan Job.
3. Import an STL model. See Figure 4-111.
    a. (Optional) Apply any offsets and/or rotations to the model. The models color can also be adjusted in the Parameter Tree


*Figure 4-111 - Importing an STL Model*

4. Create the desired processing object that will be projected onto the STL model.
    a. The desired object must have a “Max Length” value that is greater than 0. The smaller the defined “Max Length,” the more vectors the processing object will consist of. As the number of vectors increase, the longer it will take the object to project. Projection can only move existing vectors. The benefit of having more vectors is that the object will contour the STL model at a higher resolution.
    b. For this example, a 200mm diameter circle with a fill pattern will be created.
5. Once the desired object is created, it must be converted into a Multishape. This can be done by right clicking on the objects name in the Job Tree and then clicking on “Convert to Multi Shape.” See Figure 4-112.

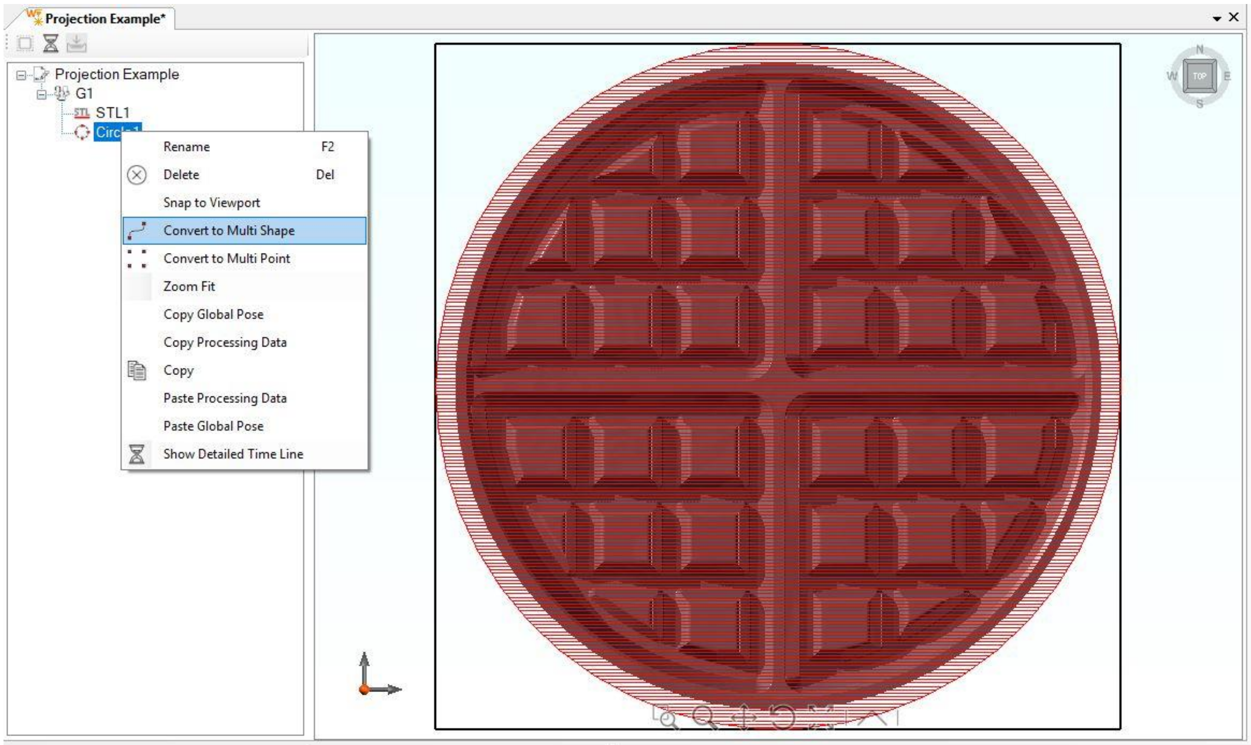



*Figure 4-112 - Converting the Process Object into a Multi Shape*

6. Next, the user can right click on the Multi Shape and click on “Project Multishape to STL.” This will bring up the “Project Multi Shape to STL” window. See Figure 4-113.


*Figure 4-113 - Project Multi Shape to STL Window*

7. In the “Project Multi Shape to STL” window, the user should select the STL model that the processing object will be projected on.
8. Next, the user needs to align the projection point. The projection point appears as a blue dot in the IPGScan Canvas (see Figure 4-114).
    a. The projection point can be thought of as a light source. This needs to be positioned so that the processing object would cast a “shadow” onto the STL model. The placement of the projection point will have different effects on how the processing object is projected onto the STL model.

        i. A projection point that is directly perpendicular to the projection surface from at a large distance to the surface will result in a projection with the least distortion.

b. The user can adjust the projection point in a number of manners.
    i. Direct Entry Method – The user can directly type values into the X, Y, and Z boxes.
    ii. Arrow Keys – The user can increment X, Y, and Z. The amount adjusted per arrow click is the amount that the Increment radio selection is set to.
    iii. Snap Point to View – The user can adjust the IPG Canvas view and then click "Snap Point To View." This will cause the projection point to snap to coordinates that pertain to the IPGScan Canvas view.

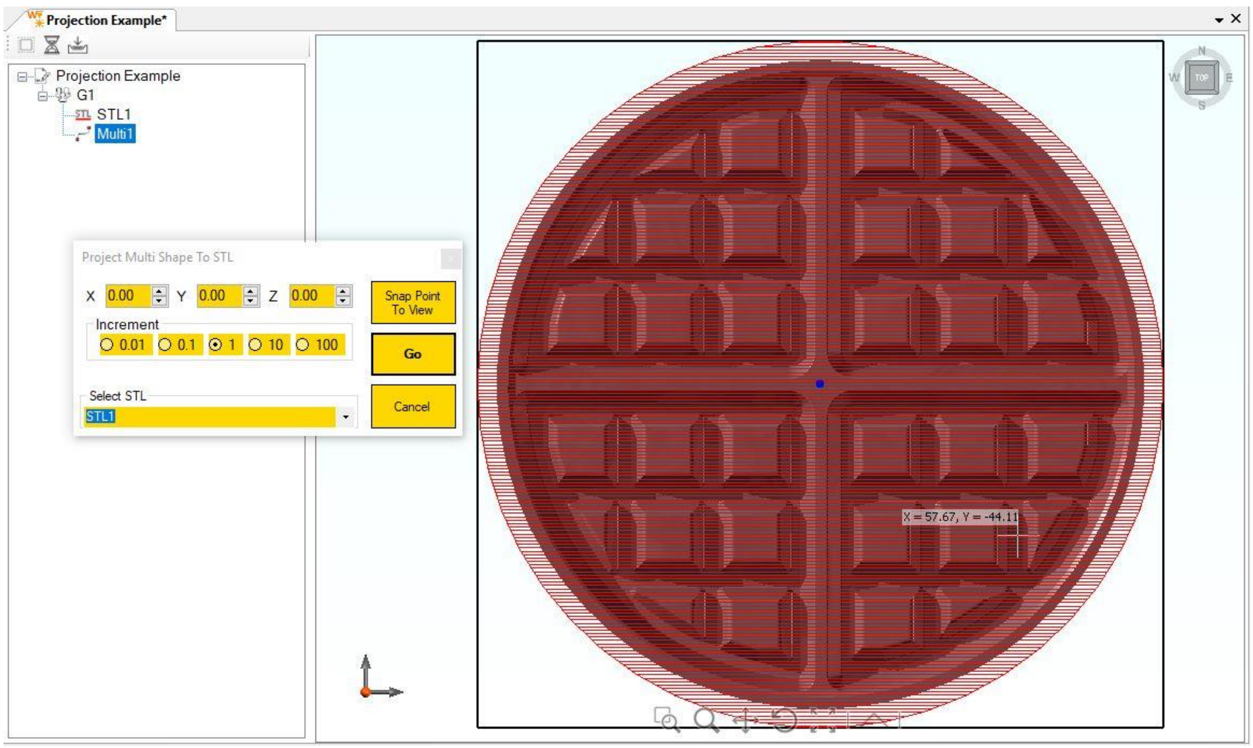



*Figure 4-114 - Projection Point*

9. Once the projection point is aligned as desired (see Figure 4-115), the user can click "Go" to project the process object.
    a. Keep in mind that any vectors that do not project directly onto the STL will be removed from the object.

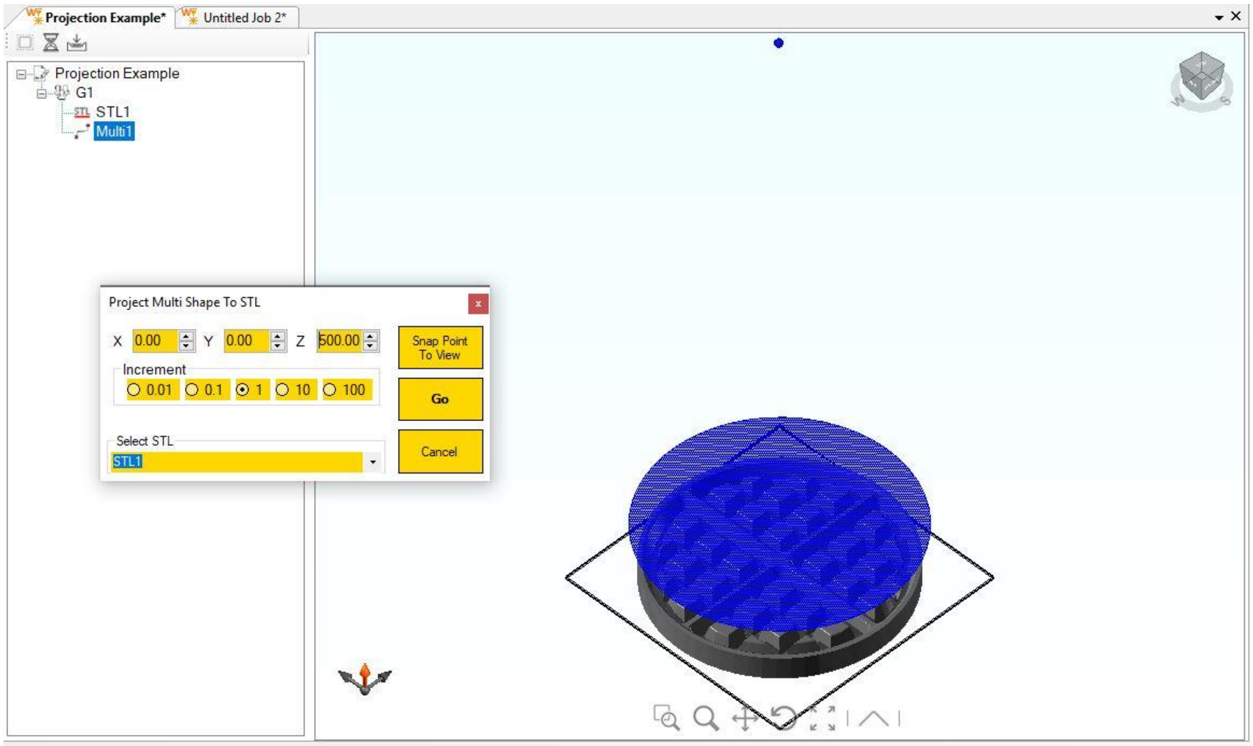



*Figure 4-115 - Aligning the Projection Point, Process Object, and STL Model*

10. Depending on the number of vectors in the process object, this process can take some time. Figure 4-116 details the warning that will appear in IPGScan while the projection process is taking place.

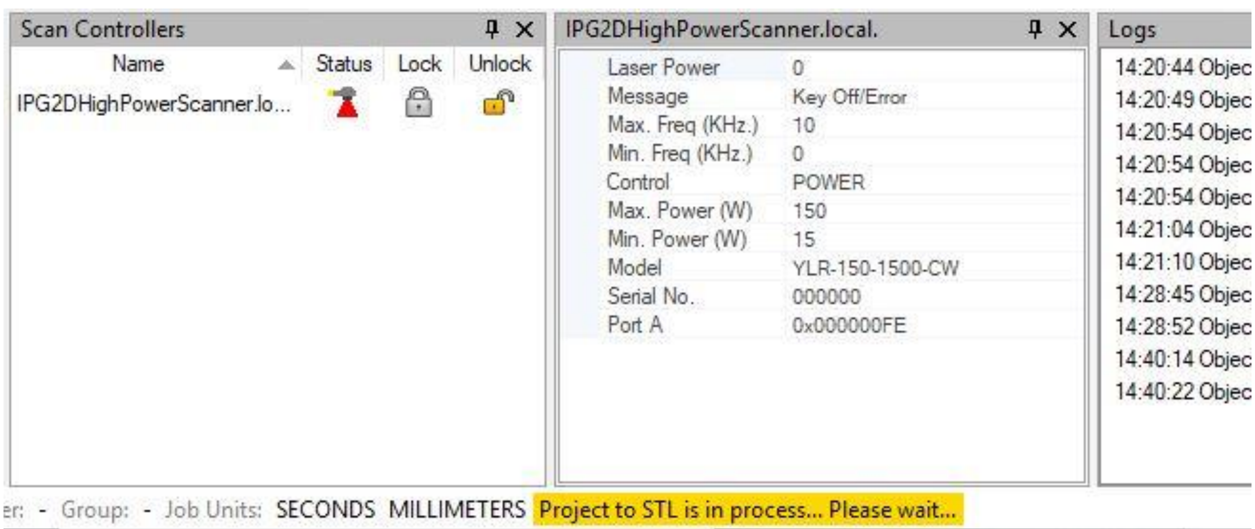



*Figure 4-116 - Warning during Projection Process*

11. Once the projection process is complete, the object will now be projected onto the STL model accordingly. See Figure 4-117.

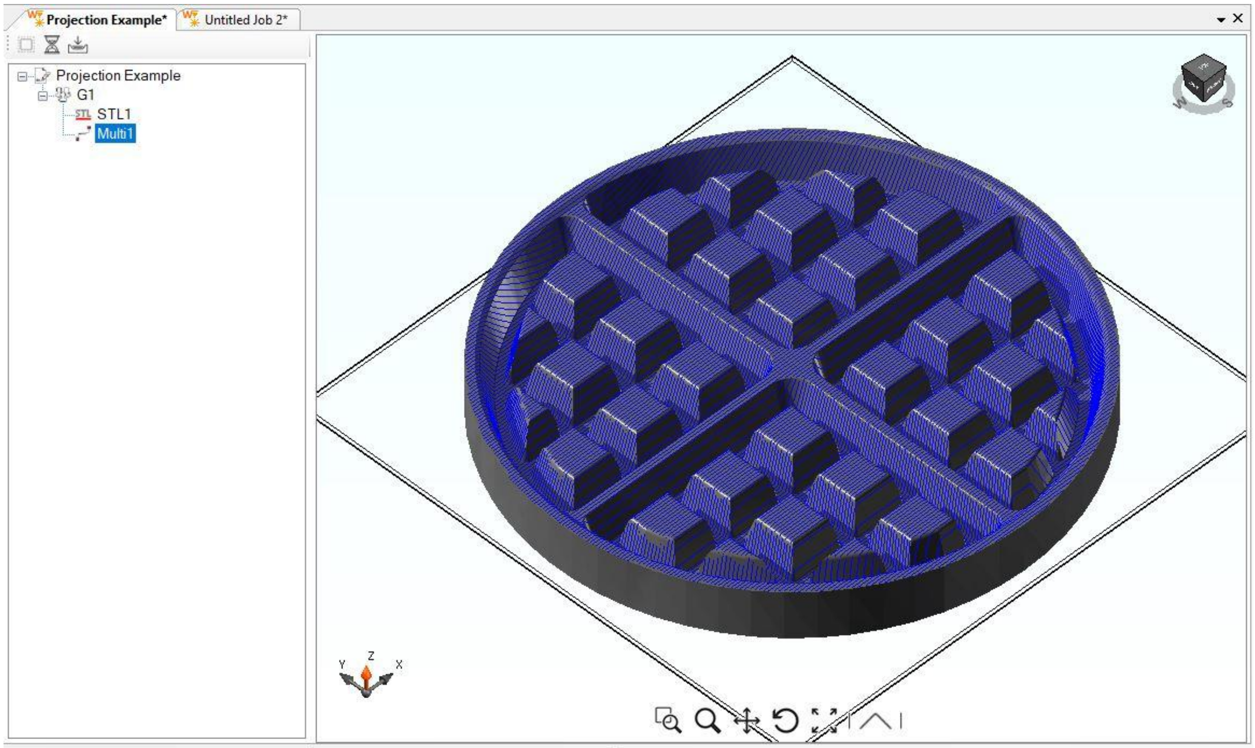



*Figure 4-117 - Process Object Projected onto an STL Model*

a. Note that because the object is projected onto the STL model by casting a “shadow,” there is a high likelihood that the original pitch defined (line to line fill spacing) will no longer be equal throughout the process object. See Figure 4-118.

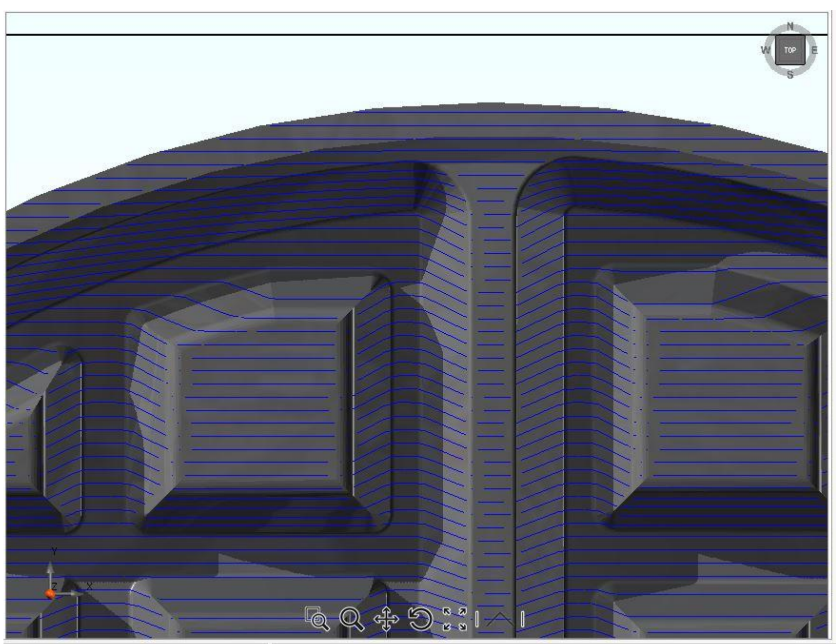


*Figure 4-118 - Uneven Fill Pitch after Projection*

b. It’s also important to know that once a process object is projected onto a STL model, the process object will maintain the contour of the model even if the model or process object is moved. In other words, the process object does not maintain an active mold of the STL model. See Figure 4-119.

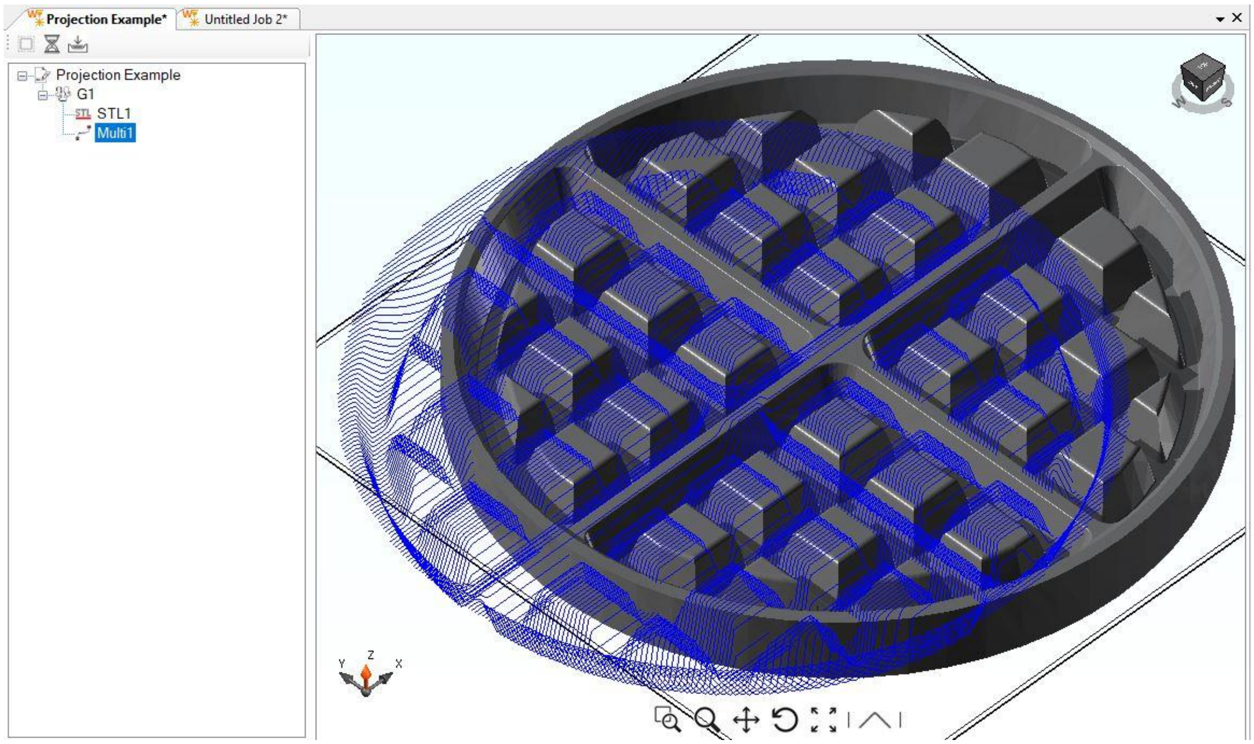



*Figure 4-119 - Process Object Moved after Projection*

### 4.12.4 Wrap Circle

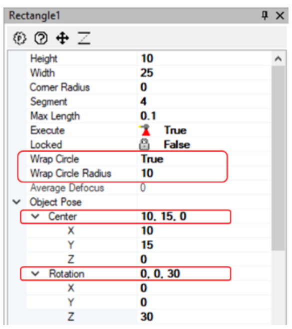
The 'Wrap Circle' feature in IPGScan can be utilized to wrap vector objects around a circle of user-specified radius. Vector objects that can be wrapped around a circle in IPGScan include text and all vector shapes that are directly defined in the IPGScan GUI as well as vector shapes that are imported from DXF files.

IPGScan treats 'Wrap Circle' as a property of a vector object, just as position and rotation. Among these object properties, 'Wrap Circle' takes priority over rotation and position.

When 'Wrap Circle' is set to true for a vector object, IPGScan first wraps that object to an imaginary circle centered at (0, 0) with the user-specified radius. After the vector object is wrapped and positioned accordingly, IPGScan then applies position and rotation settings to the 'wrapped' vector object.

For example, this rectangle object has the following three properties (see Figure 4-120):

*Figure 4-120 Example Object Wrap Parameters*




1. Wrap circle is 'True' and Wrap Circle Radius is '10'.
2. Center position is (10, 15, 0).
3. Rotation is around z-axis by 30 degree.

| | |
|---|---|
| **IMPORTANT** | **No matter at what order the above properties are being entered in IPGScan, IPGScan will apply the 'Wrap Circle' property first, followed by rotation and then the center position is adjusted. Therefore, the resultant shape and its position/orientation with respect to the origin are as shown in the picture below (**Figure 4-121**).** |

*Figure 4-121 Resulting Object with Wrap*

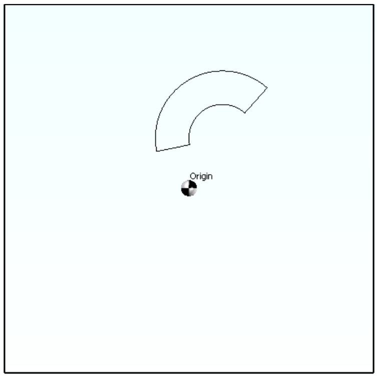
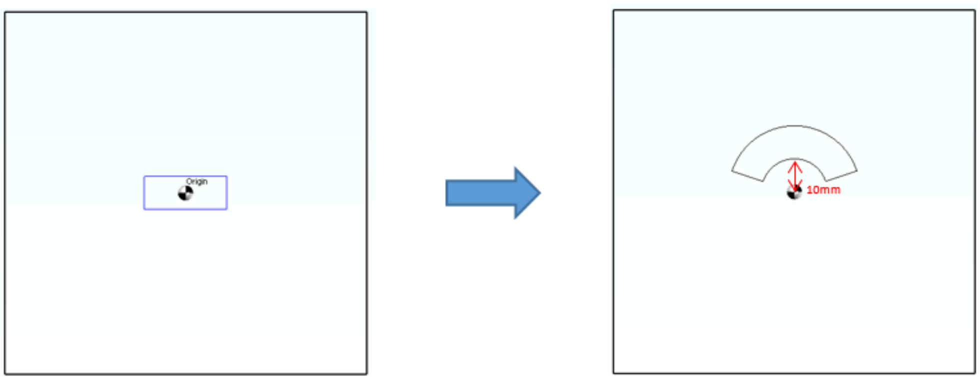



Figure 4-122 illustrates how each individual property affects the rectangle based on the priority order of the properties, although all the changes occur to the shape at the same time in the canvas.

First, ‘Wrap Circle’ and ‘Wrap Circle Radius’ is 10.

*Figure 4-122 Individual Property Effects of Wrap*



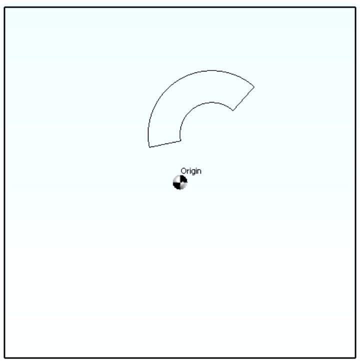

| | |
|---|---|
| **IMPORTANT** | **IPGScan always assumes the circle that a vector object wraps around is centered at origin and with user specified radius, so the resultant vector object after wrapping will be positioned around that circle (the circle will not be shown in IPGScan). If the user wants the post-wrapping vector object at a different location and/or a different orientation, the user must adjust the object’s coordinates and/or rotation accordingly.** |

Secondly, rotation around z-axis by 30 degrees. See Figure 4-123.

*Figure 4-123 Wrapped Object with Applied Rotation*

Finally, adjust position to (10, 15, 0). See Figure 4-124.

*Figure 4-124 Wrapped Object with Applied Translation*




#### *4.12.4.1 Wrap Procedure*

The following sections outline example applications of the Wrap feature.

##### 4.12.4.1.1 'Wrap Circle' to Vector Object Defined in IPGScan

The following describes the steps to set Wrap Circle property to a vector object which is defined in IPGScan.

1. In the IPGScan job file, select the desired vector object to set the Wrap Circle property.
2. Change its Max Length to a non-zero value, e.g. 0.1 (mm).

**IMPORTANT** **The default value for Max Length is 0. Failure to change Max Length to a non-zero value before "Wrap Circle" will produce an incorrect result.**

3. Set 'Wrap Circle' to True, and set Wrap Circle Radius to the desired value. The vector object will then appear wrapped around a circle of that radius centered on origin.
4. Adjust rotation and positon of the wrapped vector object accordingly, if needed.

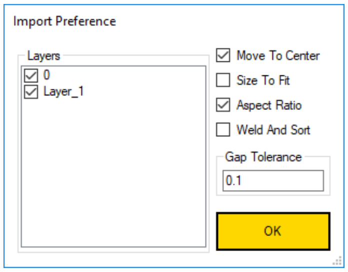
##### 4.12.4.1.2 'Wrap Circle' to Vector Object Imported from DXF

If the vector object to be imported is at (0, 0) in the DXF file, users can import the DXF file as it is. Users can then follow the same procedure described in the above section to set 'Wrap Circle'.

If the vector object to be imported is NOT at (0, 0) in the DXF file, here are the steps to ensure correct 'Wrap Circle' result.

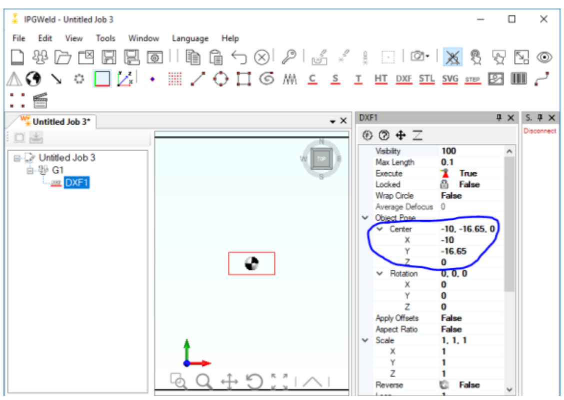
1. Import DXF and select 'Move To Center' (see Figure 4-125). The DXF example used here includes a rectangle located at (10, 16.65) in the DXF coordinate.

*Figure 4-125 DXF Import for Wrap*



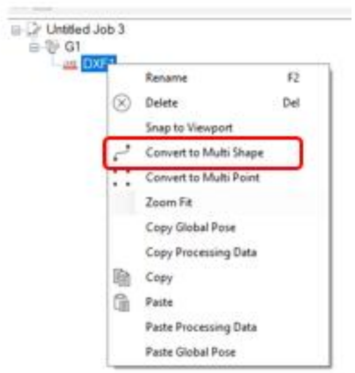

After the DXF is imported with 'Move To Center' selected, you will find the rectangle shape centered around origin in the IPGScan canvas. However IPGScan treats this rectangle as part of the DXF object, whose coordinate in the IPGScan canvas is now (-10, -16.65). See Figure 4-126. If you proceed with 'Circle Wrap' now, an incorrect result will occur. Additional steps are needed before 'Wrap Circle' is applied, as described below.

*Figure 4-126 Move to Center Applied Coordinates*



2. Select this DXF object, right click and select 'convert to Multi Shape'. See Figure 4-127.

*Figure 4-127 Convert to Multi Shape for Wrap*

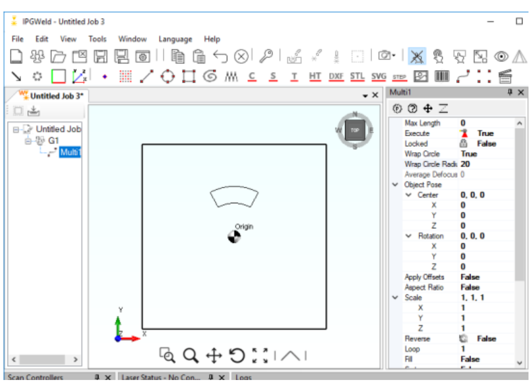



3. Select this multi-shape object and set its Max Length to a non-zero value, e.g. 0.1 (mm).

| | |
|---|---|
| **IMPORTANT** | **The default value for Max Length is 0. Failure to change Max Length to a non-zero value before "Wrap Circle" will produce an incorrect result.** |

4. Set 'Wrap Circle' to True, and set Wrap Circle Radius to your desired value. The vector object will then appear wrapped around a circle of that radius centered on origin. In this example, Wrap Circle Radius is set to 20 (mm), the resulted object is shown in Figure 4-128.

*Figure 4-128 Wrapped Circle*



5. Adjust rotation and positon of the wrapped vector object accordingly, if needed.

## 4.13 Action Controls

Action Controls are IPGScan objects which perform an action rather than fire the laser (like a circle or a point).

Most Action Controls have an "Action TimeOut" property. If an Action Control has not finished in that amount of time, IPGScan will skip that Action Control and continue. If the "Action TimeOut" is set to {-1}, no timeout will be used.

There are two types of Action Controls: those which run in IPGScan and those which are sent to the buffer (See section 4.8.1). If IPGScan is running an Action Control, it will not continue to the next object until that Action Control has finished. If an Action Control is sent to the buffer, IPGScan will send the information to the buffer and continue to the next object regardless of the status of the last Action Control. Actions which are executed in the buffer will only execute after commands which contain an End of Object (EOO). Processing Objects and some of the Action Controls specified below contain an EOO.

### 4.13.1 No Action

This performs no action; the program will not be affected in any way.

### 4.13.2 User Action

1. Displays a message box on the computer screen. An "OK" button must be pressed to continue.
2. Parameters
    a. Prompt
        i. The message to display
3. Runs in IPGScan

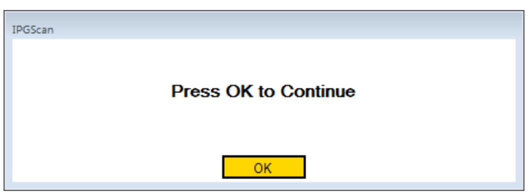



*Figure 4-129 - User Action Message*

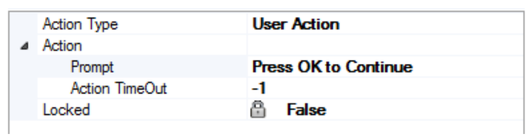



*Figure 4-130 - User Action Configuration*

### 4.13.3 Delay Action

1. Delays the scanner processing
2. Parameters
    a. Delay
        i. Time to delay the scanner processing in seconds.
    b. Absolute?
        i. Only used with On-The-Fly job types
        ii. If true, scanner processing will be delayed until the time specified in Delay based on the start of the robot trajectory.
3. Sent to the buffer.
4. Adds an EOO (See section 4.8.1)

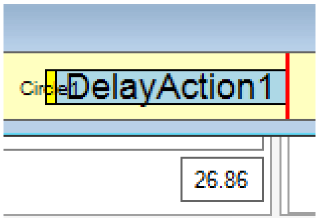



*Figure 4-131 - Delay Action; Absolute False Timeline*

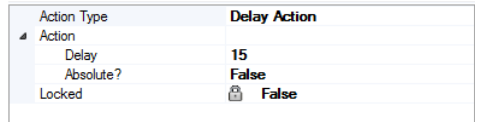



*Figure 4-132 - Delay Action; Relative False Configuration*

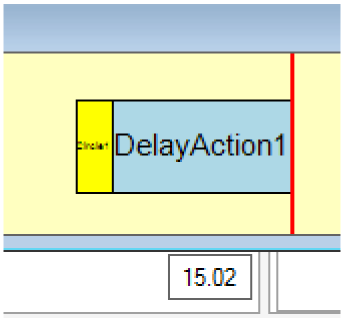



*Figure 4-133 - Delay Action; Absolute True Timeline*

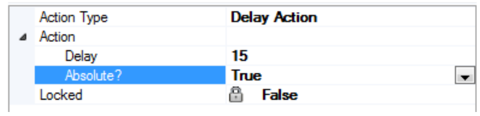



*Figure 4-134 - Delay Action; Relative True Configuration*

## 4.13.4 Streaming Data Action

All of these Action Controls are sent to the buffer and do not add an EOO, unless otherwise mentioned (See section 4.8.1).

### *4.13.4.1 Set Wait On Event*

1. Causes the scanner processing to pause until a specified input is active. Figure 4-135 shows the configuration for a wait until the Start signal is active.
2. Parameters
    a. Event
        i. Choose the input to wait to become active. Options: Start, GPIO0, GPIO1

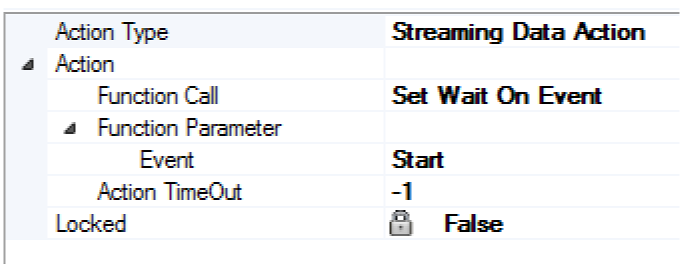



*Figure 4-135 - Set Wait On Event Start*

### *4.13.4.2 Set Wait Invert*

1. Sets the logic level for inputs for the Set Wait On Event action
2. It is possible to set different logic levels for different inputs
3. Parameters
    a. Event
        ii. The input to adjust. Options: Start, GPIO0, GPIO1

b. State
   i. Set: active low
   ii. Clear: active high

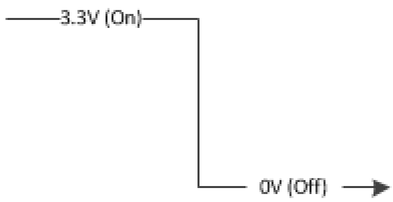



*Figure 4-136 - Set Wait Invert: Clear; Set Wait Edge: Clear*

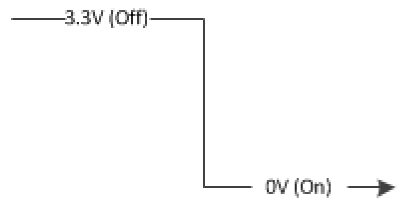



*Figure 4-137 - Set Wait Invert: Set; Set Wait Edge: Clear*

#### *4.13.4.3 Set Wait Edge*

1. Sets the trigger level for inputs for the Set Wait On Event action
2. Parameters
   a. Event
      i. The input to adjust. Options: Start, GPIO0, GPIO1.
   b. State
      i. Set: edge detection
      ii. Clear: level detection

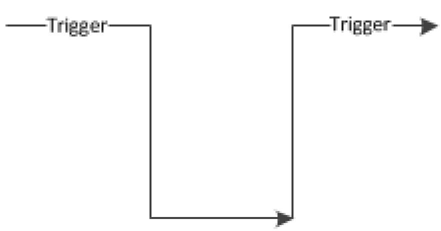



*Figure 4-138 - Set Wait Invert: Clear; Set Wait Edge: Clear*

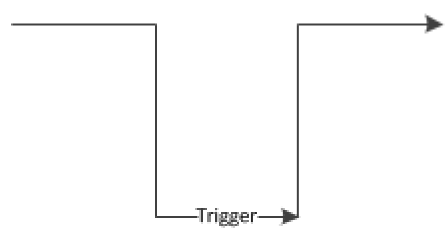



*Figure 4-139 - Set Wait Invert: Set; Set Wait Edge: Clear*

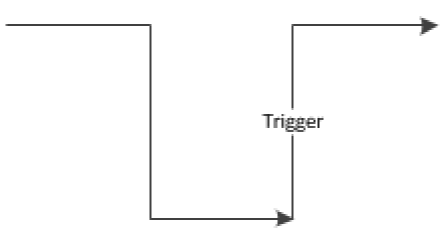



*Figure 4-140 - Set Wait Invert: Clear; Set Wait Edge: Set*

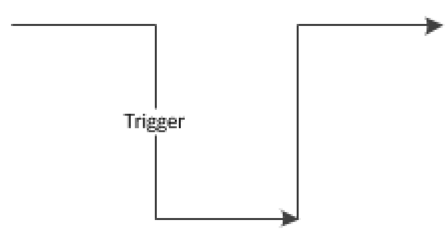



*Figure 4-141 - Set Wait Invert: Set; Set Wait Edge: Set*

#### *4.13.4.4 Set Port C*

1. Sets specific bits of Port C to 1
2. To turn on GPIO[3:0], the corresponding enable must be set to 1. If the corresponding enable is 0, the GPIO will act as an input on Port A.
3. Use of GPO[24:16] requires the Extended IO Board or the Fieldbus Base Station(see External Interface Manual.)
4. Figure 4-142 shows a configuration which will enable and turn on GPIO[0].
5. Parameters

a. Mask

i. Check the specific bits to set to 1 (on). Bits unchecked will not be changed.

| Port C Bit | Description | Port C Bit | Description |
|---|---|---|---|
| 0 | GPIO[0] | 17 | GPO[17] |
| 1 | GPIO[1] | 18 | GPO[18] |
| 2 | GPIO[2] | 19 | GPO[19] |
| 3 | GPIO[3] | 20 | GPO[20] |
| 8 | Output enable for GPIO[0] | 21 | GPO[21] |
| 9 | Output enable for GPIO[1] | 22 | GPO[22] |
| 10 | Output enable for GPIO[2] | 23 | GPO[23] |
| 11 | Output enable for GPIO[3] | 24 | GPO[24] |
| 16 | GPO[16] | | |

*Table 4-21 - Bits for Port C*

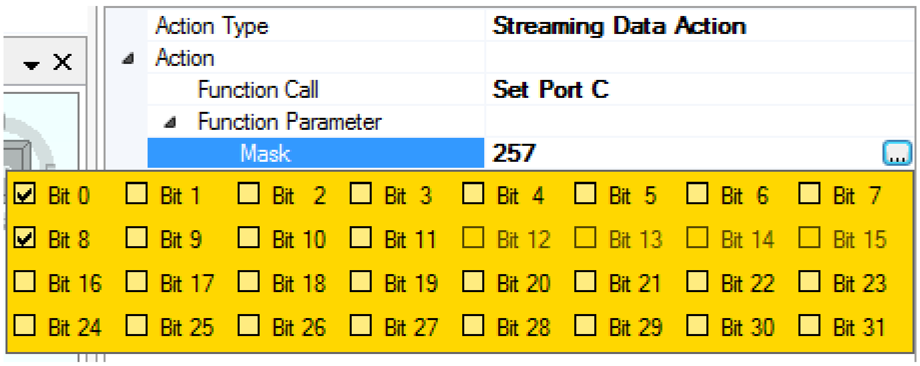



*Figure 4-142 - Set Port C*

### 4.13.4.5 Clear Port C

1. Sets bits of Port C to 0. This can turn off GPIO or turn off the output enable for a GPIO. Figure 4-143 shows a configuration which will turn off the output enable for GPIO[1], turn off GPIO[0], and not adjust the output enable for GPIO[0].
2. Parameters

a. Mask

i. Check the specific bits to set to 0 (off). Bits unchecked will not be changed.

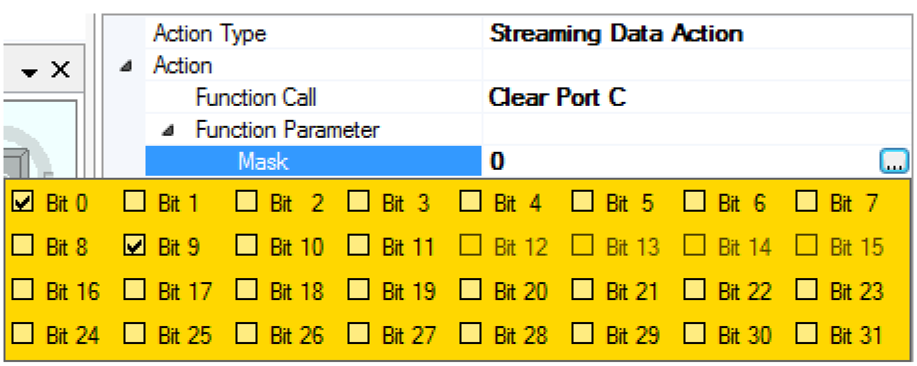



*Figure 4-143 - Clear Port C*

#### 4.13.4.6 Set Port F

1. Setting Port F can lock the signals Ready, Active, or Error on. Figure 4-144 shows a configuration which will turn on the Ready signal and will maintain the current settings for Active and Error.
2. Parameters
    a. Mask
        i. This signals to lock on. Unchecked signals are unchanged.

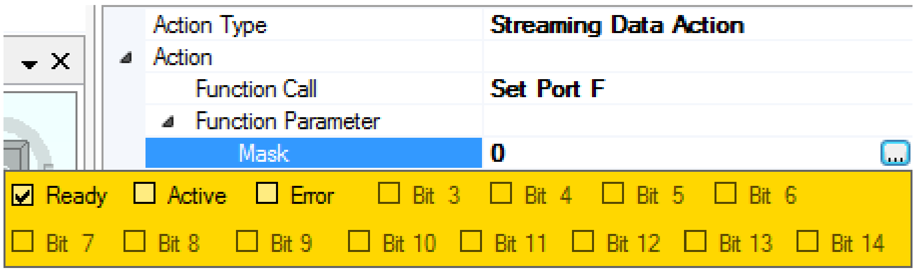



*Figure 4-144 - Set Port F*

#### 4.13.4.7 Clear Port F

1. Clearing Port F returns Ready, Active, or Error to their default logic. Figure 4-145 shows a configuration which will return the Error signal to its default logic and will not change the Ready or Active signals.
2. Parameters
    b. Mask
        i. This signals to return to default logic. Unchecked signals are unchanged.

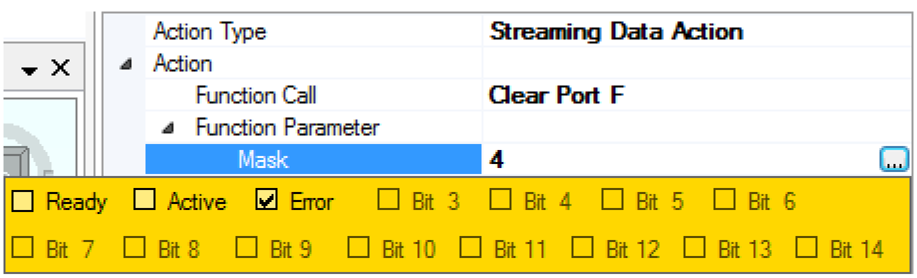



*Figure 4-145 - Clear Port F*

#### 4.13.4.8 Clear Stage

1. Clears the count of the specified stage axes. Figure 4-146 shows a configuration which will clear only Axis Z.
2. Parameters
    a. X, Y, Z
        i. True: clear the count of this axis
        ii. False: do not clear the count of this axis

| | |
|---|---|
| Action Type | Streaming Data Action |
| Action | |
| Function Call | Clear Stage |
| Function Parameter | |
| X | False |
| Y | False |
| Z | True |
| Action TimeOut | -1 |
| Locked | False |

*Figure 4-146 - Clear Stage*

#### 4.13.4.9 Laser Shutdown

1. Disables the laser.
2. Adds an EOO to the buffer.

#### 4.13.4.10 Laser On

1. Enables a type of laser at 0W power. Figure 4-147 shows a configuration which will turn on the guide beam.
    a. Parameters
        i. Laser type
            1. Main – main laser
            2. Guide – guide beam
            3. Focus – focus assist beams
            4. Focus without Guide – focus assist without the guide beam
            5. Guide without Focus – guide beam without focus assist beam (3D Scan Heads)
2. Adds an EOO to the buffer.

| | |
|---|---|
| Action Type | Streaming Data Action |
| Action | |
| Function Call | Laser On |
| Function Parameter | |
| Laser | GUIDE |
| Action TimeOut | -1 |
| Locked | False |

*Figure 4-147 - Laser On*

#### 4.13.4.11 Laser Off

1. Disables the laser.
2. Adds an EOO to the buffer.

### 4.13.5 Reset Tracking

1. If using an On-The-Fly job, this does two things. First, it will add a Set Wait On Event with the parameter Start. Second, it will reset the tracking location of a robot trajectory to 0. This is automatically called before every run of an On-The-Fly job.
2. Adds an EOO

3. Sent to the buffer.

### 4.13.6 Set Coordination Flags

1. Sets the expected value of the coordination flags for a robot trajectory. See section 12.5.5. Figure 4-148 shows a configuration which expects flag 1 to be off and flag 2 to be on.
2. Parameters
    a. Coordination Flags: the expected coordination flag value in hexadecimal
3. Does not add an EOO
4. Sent to the buffer.

| | |
|---|---|
| Action Type | **Set Coordination Flags** |
| Action | |
| Coordination Flags | **0x00000002** |
| Action TimeOut | **-1** |
| Locked | **False** |

*Figure 4-148 - Set Coordination Flags*

### 4.13.7 Set Coordination Mode

1. Sets the motion coordination mode for ScanPack.
2. Parameters
    a. Coordination Mode. Possible choices:
        i. COORDINATION_OFF: no coordinated motion
        ii. STAGE_TRACKING: ScanPack will move a stage trying to output processing objects as close to the center of the scan head as possible.
        iii. ROBOT_TRACKING: ScanPack will follow a recorded robot trajectory. This is automatically called before each run of an On-The-Fly job.
        iv. ROBOT_STATIONARY: ScanPack will use some components of coordinated motion with an external motion device but does not follow a recorded trajectory. This is called automatically with Robot Alignment (see section 12.4.9.1.2).
        v. STAGE_AUTO: If the current processing vector can be output without moving the stage, the scan head will not be moved. If the vector continues to a larger area, the stage will be moved.
3. Figure 4-149 shows a configuration which sets the current coordination mode to STAGE_TRACKING.
4. Does not add an EOO.
5. Sent to the buffer.

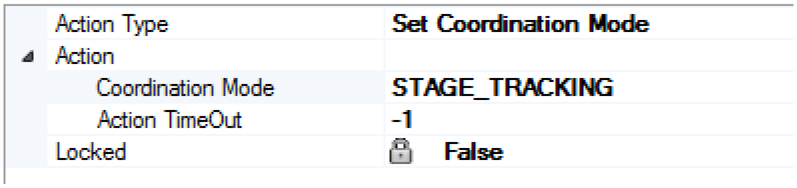



*Figure 4-149 - Set Coordination Mode*

## 4.13.8 Wait

### *4.13.8.1 Wait for Done*

1. Waits until the buffer of the connected scanner is empty.
2. Runs in IPGScan

### *4.13.8.2 Port A Bit Action*

1. Wait for a specific level of a bit on Port A. Port A corresponds to input signals with the scan controller. Figure 4-150 shows a configuration which will block execution in IPGScan until bit 16 on Port A is active.
2. Parameters
    a. Bit: the bit to check
    b. Wait For: the level to wait for. True corresponds to active; False corresponds to inactive.
3. Runs in IPGScan

| Bit | Description | Bit | Description |
|---|---|---|---|
| 0 | GPIO[0]/Strobe | 16 | GPI[16]/Select 0 |
| 1 | GPIO[1] | 17 | GPI[17]/Select 1 |
| 2 | GPIO[2] | 18 | GPI[18]/Select 2 |
| 3 | GPIO[3] | 19 | GPI[19]/Select 3 |
| 4 | GPI[4] | 20 | GPI[20]/Select 4 |
| 5 | GPI[5] | 21 | GPI[21]/Select 5 |
| 6 | GPI[6] | 22 | GPI[22]/Select 6 |
| 7 | GPI[7] | 23 | GPI[23]/Select 7 |
| 11 | Air Flow OK for High Power Heads | 24 | GPI[24]/Select 8 |

*Table 4-22 - Bits for Port A*

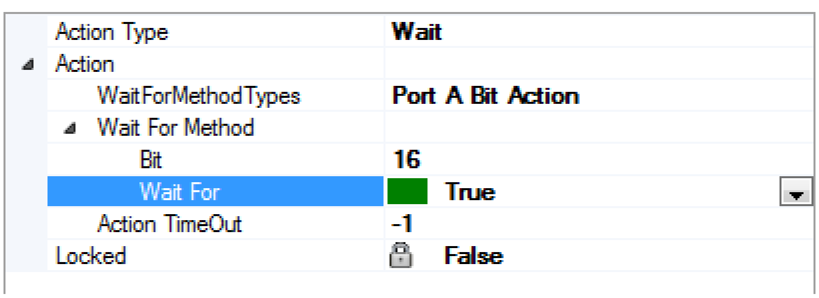



*Figure 4-150 - Port A Bit Action*

### *4.13.8.3 Wait for Strobe*

1. Waits for the Strobe input to be active with the Ready signal handshaking for Point and Shoot processing.

a. The Ready signal will be active while the Strobe signal is inactive; then the ready signal will be inactive.

2. Runs in IPGScan.

### 4.13.9 Go To Group

1. Moves the program pointer of IPGScan to a different group based on the GroupID.
2. Parameters
    a. Constant
        i. Go To: a constant value to use as the destination GroupID.
        ii. Figure 4-151 shows a configuration which will jump to the group with GroupID 5.
    b. Register
        i. Go To Group At: the register to use as a source for the destination GroupID.
        ii. Figure 4-152 shows a configuration which will jump to the group with the same GroupID that is in Variable 2.
    c. Conditional – Chooses one of two GroupIDs based upon a condition.
        i. Operand 1: the first value to use in the condition comparison
        ii. Condition: choice of > (greater than), = (equal to), < (less than)
        iii. Operand 2: the second value to use in the condition comparison
        iv. Pass GoTo: the destination GroupID if the comparison is true
        v. Fail GoTo: the destination GroupID if the comparison is false
        vi. Figure 4-153 shows a configuration that will jump to Group 7 if Variable 3 is greater than 3 or Group 2 if Variable 3 is less than 3.
3. Runs in IPGScan

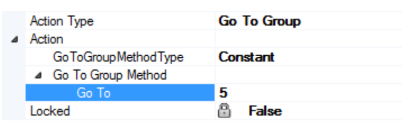



*Figure 4-151 - Go To Group Constant*

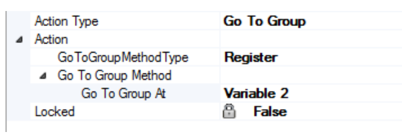



*Figure 4-152 - Go To Group Register*

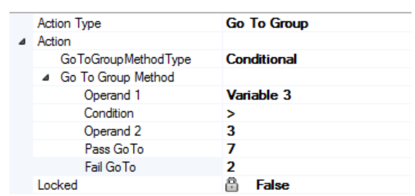



*Figure 4-153 - Go To Group Conditional*

### 4.13.10 Load Register

All of the following Load Register methods set or modify the value of IPGScan registers. These methods all run in IPGScan.

#### *4.13.10.1 Port A*

1. Reads the value of the Port A inputs as a binary number, applies a bit shift, limits the width of the bits (bit mask), and converts it to a decimal number. An active input corresponds to a value of 1.
2. Parameters
    a. Shift: the right bit shift applied to the value read from Port A. The parameter is specified in hexadecimal.

b. Width: the number of bits to convert to decimal after applying the shift, counting from the right (least significant bit). Similar to a bit mask. The parameter is specified in decimal.
c. Destination Register: the IPGScan register to store the converted decimal number

#### *4.13.10.2 Serial Port*

1. Reads information from a Serial port on the computer running IPGScan and stores the value in an IPGScan register.
2. Parameters:
    a. COM Port Settings: standard Serial communication settings
    b. Command: the string sent by IPGScan to the external device to request information. Can be used to distinguish which information is being requested.
    c. End Delimiter: IPGScan will read data from the Serial port until it receives this string
    d. Acknowledgement: a string sent by IPGScan to inform the external device that it has successfully received the information. *Optional.*
    e. Destination Register: the IPGScan register to store the information received from the external device.

#### *4.13.10.3 User*

1. Creates a prompt box that asks the user for the data to store in the register.
2. Parameters
    a. Prompt: The message to show in the prompt box.
    b. Destination Register: the IPGScan register to store the information received from the user.

#### *4.13.10.4 Ethernet*

1. Reads information from a TCP client and stores the value in an IPGScan register. This uses the TCP settings and Actions Port in the IPGScan Options.
2. Parameters
    a. Command: the string sent by IPGScan to the external device to request information. Can be used to distinguish which information is being requested.
    b. $1^{st}$ Response: Expected message from the TCP client before the message with the register content. End Delimiter is expected with this message as well.
    c. End Delimiter: IPGScan will parse each packet of information from the TCP client looking for this End Delimiter. When IPGScan receives the End Delimiter, it will stop reading the TCP client. In the packet that includes the End Delimiter, if there is data after the End Delimiter, IPGScan will raise an error.
    d. Acknowledgement: a string sent by IPGScan to inform the TCP client that it has successfully received the information. *Optional.*
    e. Destination Register: the IPGScan register to store the information received from the TCP client.

#### *4.13.10.5 Constant*

1. Stores a preset value in a register. This value is set by the IPGScan programmer and not by the user.
2. Parameters
    a. Constant: the value to store
    b. Destination register: the IPGScan register to store the value

#### *4.13.10.6 Concatenate*

1. Combines the value of two IPGScan registers into a third IPGScan register. For example, "A" + "B" = "AB".
2. Parameters
    a. Operand 1: the first IPGScan register to use
    b. Operand 2: the second IPGScan register to use
    c. Destination Register: the IPGScan register to store the combined value

#### *4.13.10.7 RegEx*

1. Applies a Regular Expression (RegEx) pattern to an IPGScan register and stores the resulting value in a different IPGScan register.
2. Parameters
    a. Pattern: the RegEx pattern to apply
    b. Source Register: the IPGScan register to use as a source string
    c. Match Index: index of all matches found to store. This index is zero based.

#### *4.13.10.8 Math*

1. Applies a mathematical operation to two IPGScan registers and stores the result.
2. Parameters
    d. Operand 1: the first IPGScan register to use
    e. Operand 2: the second IPGScan register to use
    a. Operand: Mathematical operation to apply. Choice of:
        i. Addition
        ii. Multiplication
        iii. Sine
            1. Calculated as sin(Operand 1)
            2. Calculated in radians
        iv. Cosine
            1. Calculated as cos(Operand 1)
            2. Calculated in radians
        v. Tangent
            1. Calculated as tan(Operand 1)
            2. Calculated in radians
        vi. atan2
            1. Calculated as atan2(y,x) where y is Operand 1 and x is Operand 2
            2. Calculated in radians

vii. Square Root
b. Destination Register: the IPGScan register to store the calculated value

#### *4.13.10.9 Increment*

1. Increments the value in a register and stores the new value in the same register.
2. Parameters
    a. Increment By: the value to increment the register by
    b. Minimum Digits: if the resulting number of digits in the register is less than the minimum number of digits, zeros are padded in front of the register
    c. Destination Register: the IPGScan register to increment and store the new value in

#### *4.13.10.10 Date*

1. Stores the current date in a register.
2. Parameters
    a. Date Type
        i. Default
            1. Generated date will follow the format of the host computer
        ii. DayOfYear
            1. Generated date will be an integer representing the day of the year in the format "ddd".
        iii. Julian
            1. Generated date will be formatted in the Julian date format of "yyddd".
        iv. Formatted
            1. Generated date based upon the additional parameter "Format." Format string follows "Custom Date Time."
    b. Destination Register: the IPGScan register to store the date in

#### *4.13.10.11 Time*

1. Stores the current time in a register. The time is represented with 24-hour clock.
2. Parameters
    a. Destination Register: the IPGScan register to store the date in

#### *4.13.10.12 Custom Date Time*

1. Stores a custom configuration of time and date into a register.
2. Parameters
    a. Format: the format pattern that should be used to render the custom date and time. This follows the C# DateTime formatting convention (https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings).
3. Destination Register: the IPGScan register in which to store the Date Time
4. Example
    a. "yyyy/MM/dd" will render as "2019/06/27" on June 27, 2019
    b. A few common patterns are:
        i. 'MM': the month, from 01 through 12

ii. 'd': the day of the month, from 1 to 31
iii. 'dd': the day of the month, from 01 to 31
iv. 'yyyy': the year as a four digit number
v. 'hh': the hour, using a 12-hour clock from 01 to 12
vi. 'HH': the hour, using a 24-hour clock from 00 to 23
vii. 'mm': the minute, from 00 through 59
viii. 'ss': the second, from 00 through 59

##### *4.13.10.13 Timer*

1. Runs a software timer and sets an IPGScan register to the elapsed time when stopped. Multiple timers can be used at the same time.
2. Parameters
    a. Action: the timer action to perform
        i. Reset: reset the value of the timer
        ii. Start: start running the timer
        iii. Stop: stop running the stopwatch and store the time of the timer
    b. Wait For Done On Stop: Wait for the buffer to be empty before stopping the timer. This parameter is only used with the Stop action. It is useful when timing actions or objects that execute from the buffer.
    c. Destination Register: the IPGScan register to store the time in. This is also used to specify which timer act on.

##### *4.13.10.14 From XML*

1. Gets a value from an XML file and stores the value in an IPGScan register.
2. Parameters
    a. XML File Path: the location of the XML file. It can be a local file on the computer running IPGScan or a remote location accessible by the computer running IPGScan.
    b. XPath: the path to the specific tag which contains the desired value
    c. ID Value: value of the identifying attribute to look for
    d. ID Attribute: identifying attribute name
    e. Value Attribute: name of the attribute to retrieve a value from
    f. Destination Register: the IPGScan register to store the value in
3. Examples
    a. Given the following XML file (Figure 4-154) and IPGScan configuration (Figure 4-155), the destination register (Variable 1) would contain "-0.5".

```xml
<root>
    <Item id="0" x="0" y="0" />
    <Item id="1" x="-0.5" y="10" />
</root>
```

*Figure 4-154 - Example XML File 1*

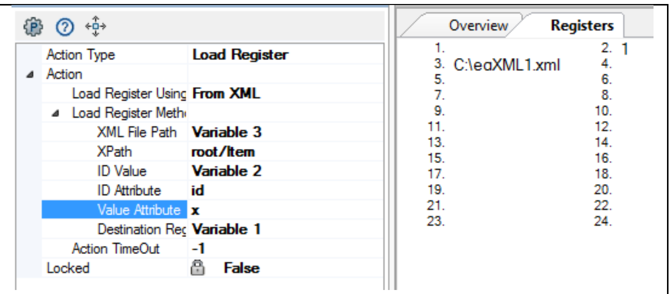



*Figure 4-155 - Example XML Configuration 1*

b. Given the following XML file (Figure 4-156) and IPGScan configuration (Figure 4-157), the destination register (Variable 7) would contain “Foo”.

```xml
<root>
    <Item id="0">Foo</Item>
    <Item id="1">Bar</Item>
</root>
```

*Figure 4-156 - Example XML File 2*

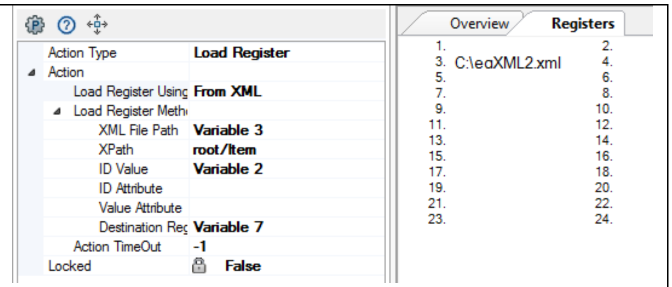



*Figure 4-157 - Example XML Configuration 2*

c.

### *4.13.10.15 TCP Client*

1. Similar to Ethernet, this Action Control asks a remote device for a value to store in an IPGScan register. However, in this Action Control, IPGScan starts a TCP Client and asks a TCP server. The TCP Client is closed after the action is complete.
2. Parameters:
    a. Command: the string sent by IPGScan to the external device to request information. Can be used to distinguish which information is being requested.
    b. 1st Response: Expected message from the TCP server before the message with the register content. End Delimiter is expected with this message as well.
    c. End Delimiter: IPGScan will parse each packet of information from the TCP server looking for this End Delimiter. When IPGScan receives the End Delimiter, it will stop reading the TCP server. In the packet that includes the End Delimiter, if there is data after the End Delimiter, IPGScan will raise an error.
    d. Acknowledgement: a string sent by IPGScan to inform the TCP server that it has successfully received the information. *Optional.*
    e. IP Address: the IP Address of the TCP server
    f. Port: the port which the TCP server is listening to
    g. Destination Register: the IPGScan register to store the value in

### 4.13.10.16 Siemens PLC

1. Reads values from a Siemens PLC DataBlock and stores that value in an IPGScan register. The PLC has to be configured in the PLC section of the Options. The PLC has to be connected before running this Action Control.
2. Parameters
    a. PLC Index: which of the configured PLCs to communicate with (defined in the Options Section)
    b. Data Type: the type of data to read. Choice of: Counter, Timer, Input, Output, Memory, DataBlock.
    c. Data Block ID: ID of the Data Type option above (for example DB1, T45, etc)
    d. Variable Type: The type of Siemens variable used. Choice of: Bit, Byte, Word, DWord, Int, Dint, Real, String, StringEx, Timer, Counter.
    e. Start Byte Address: memory address offset for the item within the data structure.
    f. Destination Register: the IPGScan register to store the value in

### 4.13.10.17 Find Focus

1. Runs the Find Focus routine with the camera at a specified location. The focus offset result is stored in an IPGScan register.
2. Parameters
    a. At Point: the location to run Find Focus, specified as (X, Y) in the Scan Head's frame of reference.
    b. Destination Register: the IPGScan register in which to store the Z offset

### 4.13.10.18 Script

1. Runs a custom JavaScript script. The example in Figure 4-158 shows a script which generates a random number and stores that value in Variable1.
2. Parameters
    a. Data Type: the data type in which to load registers into the script and set registers out of the script.
    b. Script: The JavaScript script to run. All IPGScan registers can be addressed as "Variable*n*.Value" where '*n*' is the variable number. Only modified registers will be overwritten when the script is finished running.

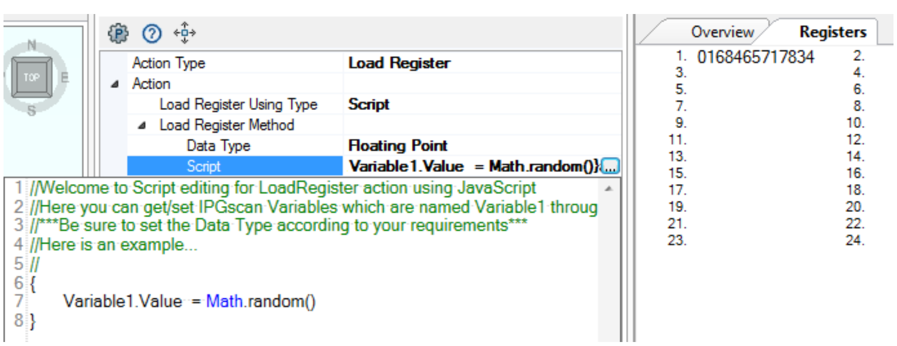



*Figure 4-158 - Load Register Script Example*

### 4.13.11 Write Register

All of the following Write Register methods export value of IPGScan registers. These methods all run in IPGScan.

#### *4.13.11.1 Ethernet*

1. Writes the value of an IPGScan register to a TCP Client. This Action Control uses the TCP settings and Actions Port from the IPGScan Options. Figure 4-159 and Figure 4-160 show an example of this Action Control.
2. Parameters
    a. Write From: The IPGScan register from which to get the value to write. Empty registers will not be written.
    b. End Delimiter: A delimiter appended to the end of the value from the IPGScan register. *Optional.*

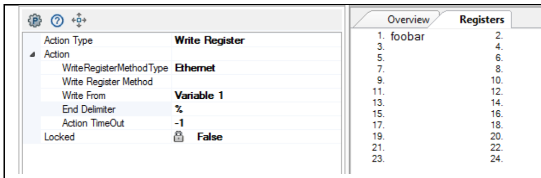



*Figure 4-159 - Write Register Ethernet Configuration*

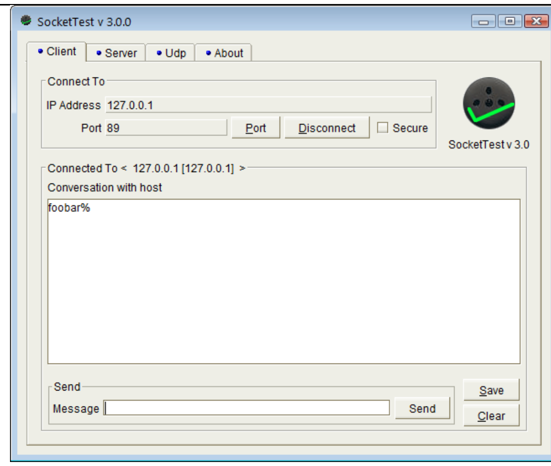



*Figure 4-160 - Write Register Ethernet Output*

#### *4.13.11.2 File*

1. Writes the value of an IGPScan register to a file. The value of an IPGScan register will be written to a new file if the file does not exist or will be appended to the file if it does exist. Each time the value is written to the file, the date and time will be written to the file first.
2. Parameters
    a. Write To File: The name of the file to use. The starting directory for the file is “C:\IPGP\IPGScan\RegisterValues\”.
    b. Write From: the IPGScan register to write to the file
    c. End Delimiter: a string to add to the file to signify the end of the contents of the register. *Optional.*

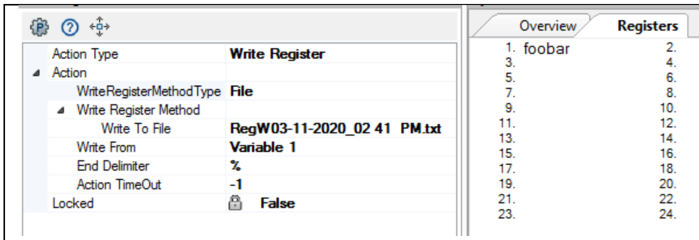


Figure 4-161 - Write Register File Configuration

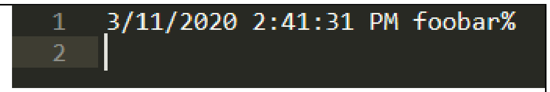


Figure 4-162 - Write Register File Output

### 4.13.11.3 Siemens PLC

1. Writes the value of an IPGScan register to a Siemens PLC variable.
2. Parameters
   a. PLC Index: which of the configured PLCs to communicate with
   b. Data Type: the type of data to write to. Choice of: Counter, Timer, Input, Output, Memory, DataBlock.
   c. Data Block ID: ID of the Data Block
   d. Variable Type: The type of Siemens variable used. Choice of: Bit, Byte, Word, DWord, Int, Dint, Real, String, StringEx, Timer, Counter.
   e. Start Byte Address: Address of the starting byte.
   f. Write From: the IPGScan register to write to the file
   g. End Delimiter: not used

### 4.13.11.4 To TCP Server

1. Writes the value of a register to a TCP Server
   a. Parameters
      i. IP Address: IP Address of the server
      ii. Port: Port of the server
      iii. Write From: the IPGScan register to write to the server
      iv. End Delimiter: Optional string to add to the end of the register

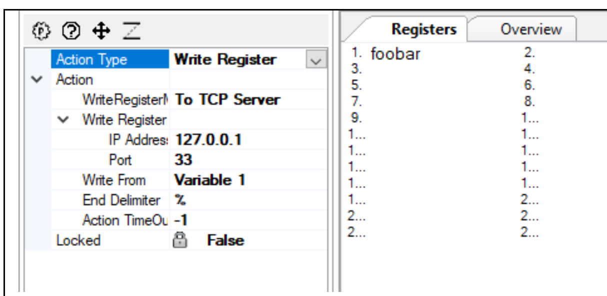


Figure 4-163 - Write Register To TCP Server Configuration

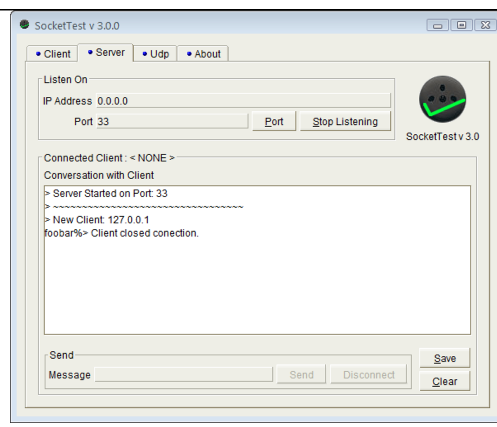


Figure 4-164 - Write Register To TCP Server Output

#### 4.13.12 Park At Action

1. This Action Control parks the galvos at the specified location.
2. Parameters
    a. Using Variables?: True if the position is specified by IPGScan registers. False if the position is specified by the IPGScan programmer.
    b. Park At: The location to park the galvos in the scanner frame represented as (X,Y,Z). Either numbers if not using variables or the IPGScan registers to use for X, Y, and Z.
3. Sent to the buffer.
4. Adds an EOO flag

#### 4.13.13 Stage Motion Action

1. This Action Control moves the stage to the specified location.
2. Parameters
    a. Move To: The location to move the stage in the stage frame represented as (X,Y,Z).
    b. Home X: True if the X axis should be homed before moving.
    c. Home Y: True if the Y axis should be homed before moving.
    a. Home Z: True if the Z axis should be homed before moving.
3. Sent to the buffer.
4. Adds an EOO flag

#### 4.13.14 Exit Action

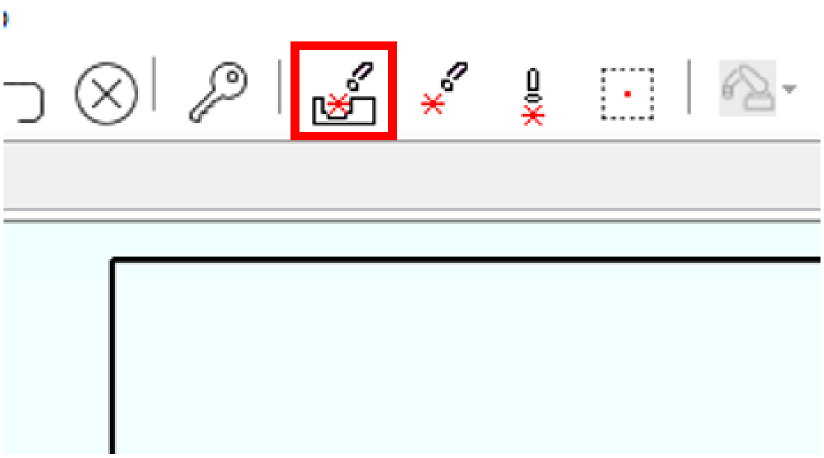
1. This Action Control stops the currently running IPGScan job. The processing window will remain open. The action waits for the buffer to be empty before stopping.

## 4.14 Previewing and Running an IPGScan Job

To preview or run a job in IPGScan, users must open the Processing Window. This can be done manually or in an automated manner (i.e. using the Remote API).

To open the Processing Window, users can click the “Start Processing” button in the Tool bar (see Figure 4-165) or select Tools → Start Processing.

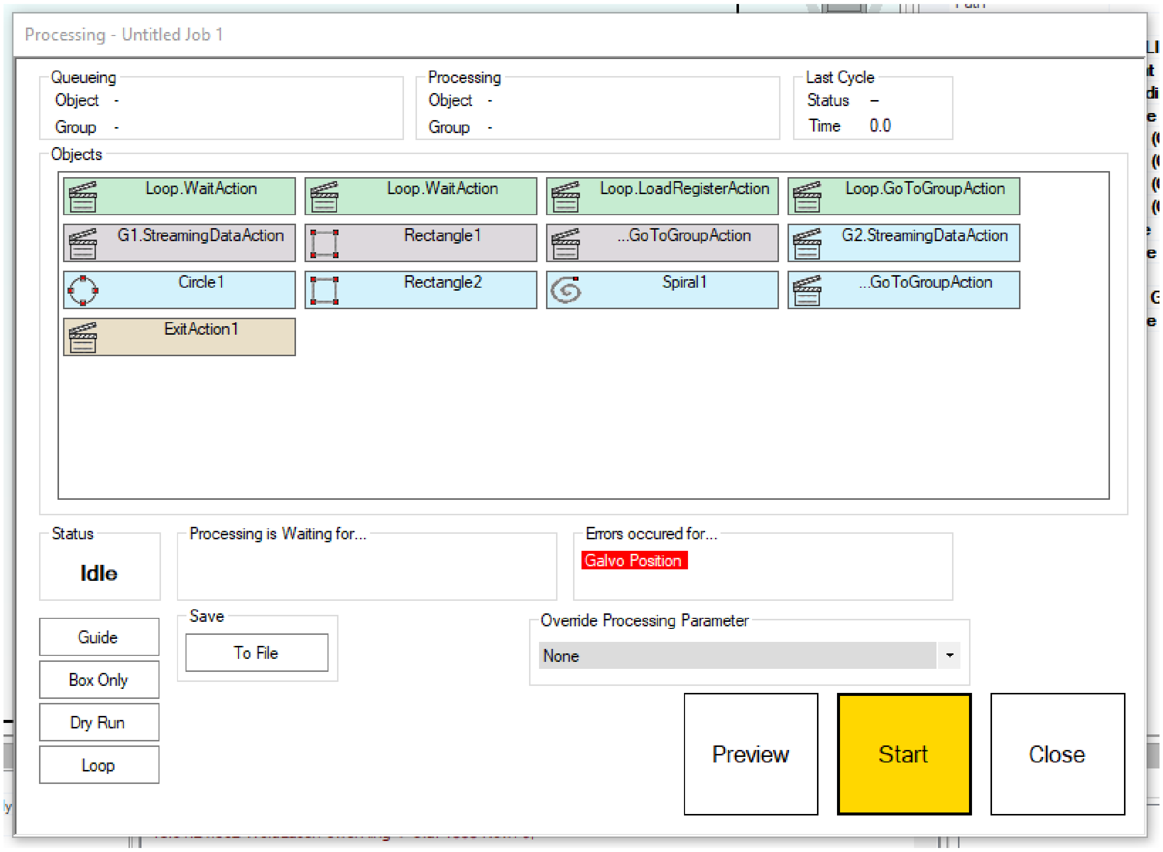
*Figure 4-165 Start Processing Button*



**IMPORTANT** **A connection with a Scan Controller is required to open the Processing Window.**

Figure 4-166 illustrates the Processing Window.

*Figure 4-166 Processing Window*

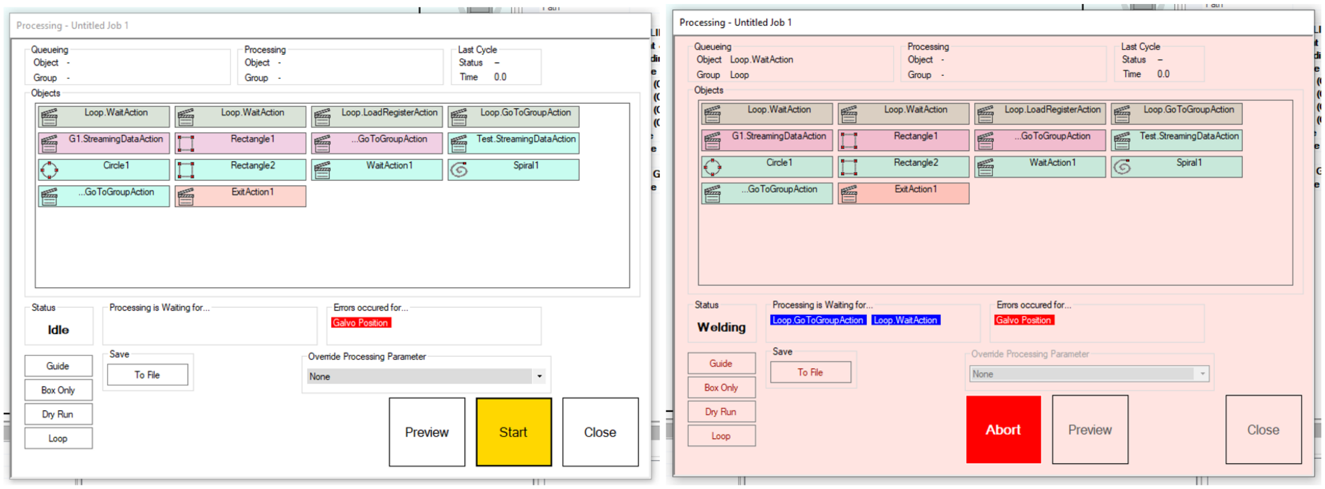



Within the Processing Window, users will find numerous status indicators:

- Queueing Box
  - This contains the Object name and Group name that IPGScan is actively sending or attempting to send to the scanners buffer.
- Processing Box
  - This contains the Object name and Group name of the currently processing object and Group.
- Last Cycle Box
  - This will display the last cycles completed time and the status of the cycle. For instance, if the process completes successfully, the status will state “OK.” If the cycle does not complete successfully (i.e. is aborted), the status will state “Failed.”
- Objects Box
  - A display of all the Process Objects and Action Controls that are selected and available for processing. If users, selected the job name in the Job Tree prior to opening the Processing Window, all of the Process Objects and Action Controls within the job will be listed in the Objects box. If users only select certain items within the Job Tree prior to opening the Processing Window, only the selected items will be within the Objects box.
- Status Box
  - This provides an indicator as to whether or not processing has started. For instance, prior to users clicking the “Start” button to start processing, this status will indicate

"Idle." Once processing has been started, it will indicate welding/cleaning/marking. See Figure 4-167.

*Figure 4-167 Idle and Actively Processing Process Windows*

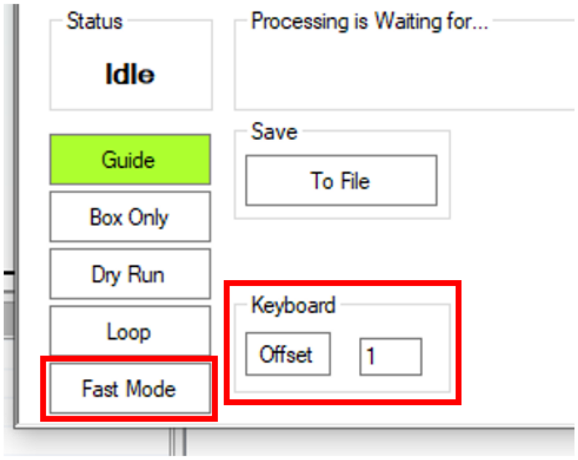



- "Processing is Waiting for…" Box
  - This provides an indication as to what Process Object or Action Control IPGScan is actively waiting on (i.e. which object is processing or which Action Control requires an action).
- "Errors occurred for…" Box
  - Lists any errors that are actively present.

In addition to statuses, the Processing Window also offers the following functionality.

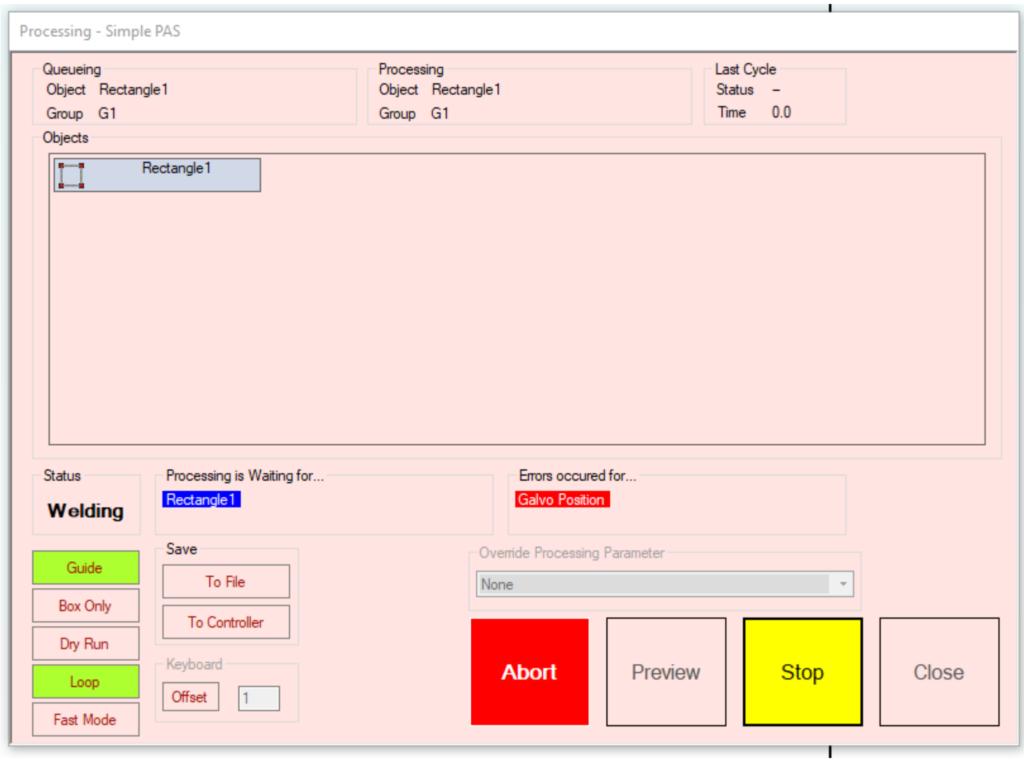
- Guide Button – When enabled, the selected Process Objects/IPGScan program will execute with the guide laser and not the processing laser. Enabling guide will also reveal the following buttons (see Figure 4-168):
  - Fast Mode – This will cause the Process Object to output at the speed defined for the Fast Mode process parameters profile.
  - Keyboard Offset – This allows users to nudge Process Objects using the computer keyboard. See section "Keyboard Offset" for additional details on functionality.

*Figure 4-168 Fast Mode and Keyboard Offset Buttons*




- Box Only Button – This will cause a box to be displayed around the place of any process objects using the guide laser. This functionality is typically used for approximate Process Object placement using the guide laser.
- Dryrun Button – Utilized for simulating a process in Robotic On-The-Fly processing. See section "Robotic On-The-Fly Processing" for additional detail.
- Loop Button – Will continuously loop the selected job when processing is started. Anytime the Loop function is enabled, two additional buttons become available when processing is started (see Figure 4-169).
  - Abort Button – This will stop processing, even if in the middle of a Process Object (it will interrupt the process).
  - Stop Button – This will stop the loop and allow objects loaded within the queue to complete prior to stopping the process.

*Figure 4-169 Processing Window Stop and Abort Buttons*




- Override Processing Parameter Selection – Allows users to select a parameter profile that will override any Process Objects during previewing or processing.
- Preview Button – Automatically enables the Guide and Loop buttons and starts processing.
- Start Button – Starts processing. Any functionality that is enabled (i.e. Guide and Loop buttons selected) will be applied. Once processing is started, an additional button becomes available:
    - Abort Button – This will stop processing, even if in the middle of a Process Object (it will interrupt the process).
- Close Button – Closes the Processing Window.

# 5 Laser On Monitor

## 5.1 Overview

Laser On Monitor provides users with a means of better detecting if unverified changes have been made to a process. For instance, once users have setup and verified that their process (i.e. welding process) produces the desired part quality, users would then implement Laser On Monitor functionality to help ensure that unauthorized changes to the job do not cause a deviation in process quality. Laser On Monitor helps to achieve this by comparing a trained recording of the job to the real process as it is executed. If any laser on/off deviations are detected between the recording and the real process, a fault condition will occur and the user will be notified.

The following sequence outlines the general process for setting up and utilizing Laser On Monitor.

1. Users should create a desired Point and Shoot or On The Fly job for processing a given part.
2. Conduct process development until results are as desired (i.e. welds meet penetration, interface, and profile requirements).
3. With process results developed, conduct a Laser On Monitor training by running the complete process one time through.
4. Verify the results from the one cycle were as desired.
5. Assuming the results from the training cycle were ok, enable Monitoring and begin to cycle the system for production.
    a. If an unauthorized change is ever made to the weld job and the system is not retrained (and quality verified by the user), IPGScan will set an error and stop the process if a deviation in laser on/off timing is detected between the real process and the training.

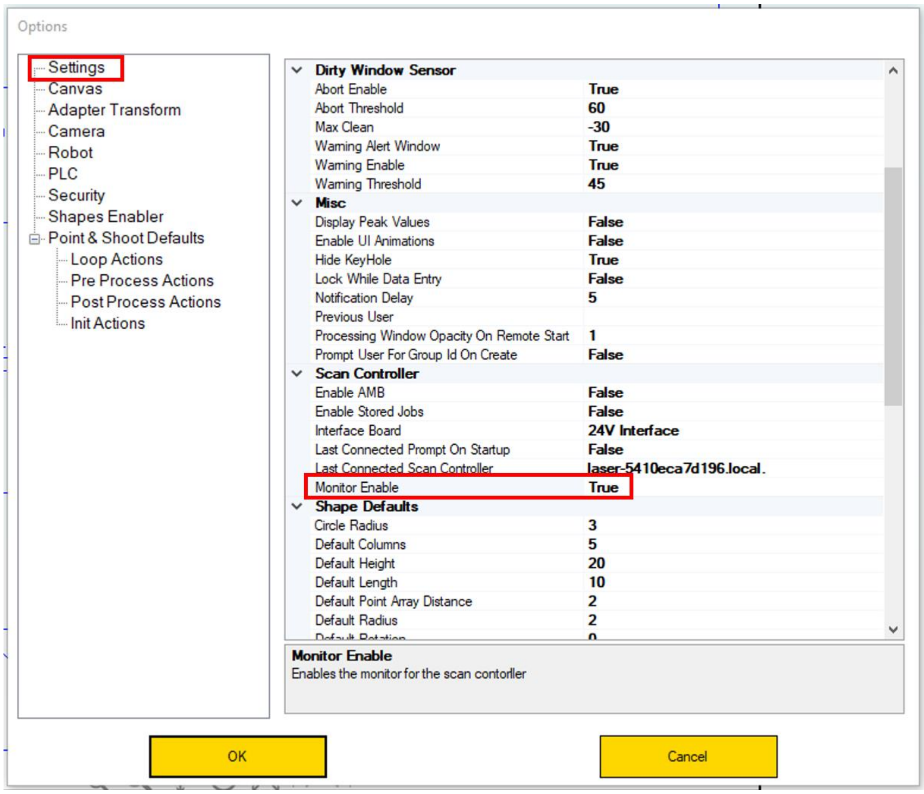
| | |
|---|---|
| **IMPORTANT** | **Laser On Monitor compares laser on/off timings of a training/recording to the real process when Monitor is enabled. Examples of process changes that could result in changes to laser on/off timings include changes to process object velocity, size, positon, and more.**<br><br>**Users should also utilize IPGScan System Security for additional process control measures.** |

The following sections detail the setup and use of Laser On Monitor.

## 5.2 Initial Setup

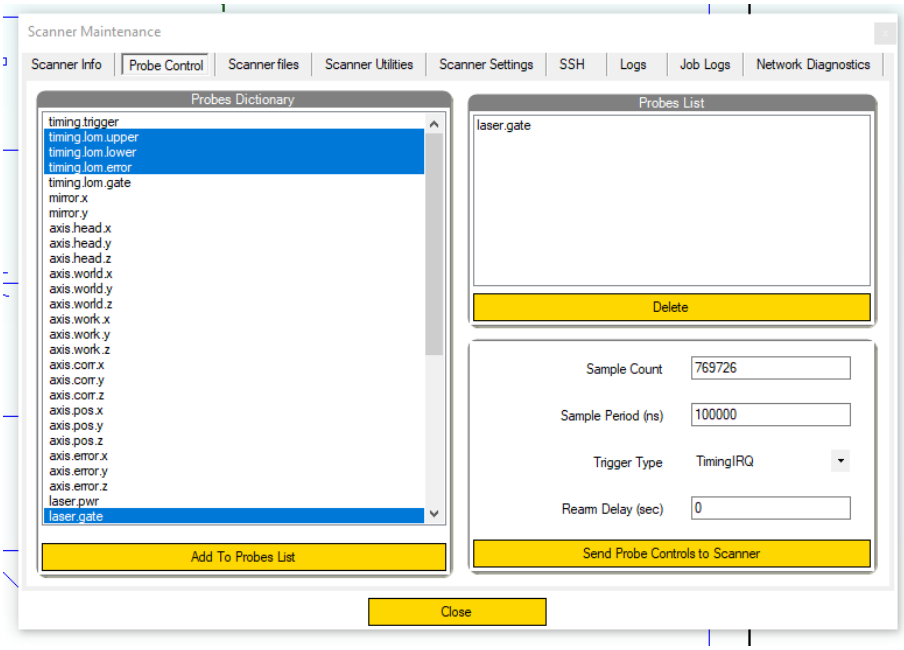
In order to utilize Laser On Monitor functionality, users must first set “Monitor Enable” to True in the IPGScan Options menu. See Figure 5-1.

*Figure 5-1 Monitor Enable Setting*



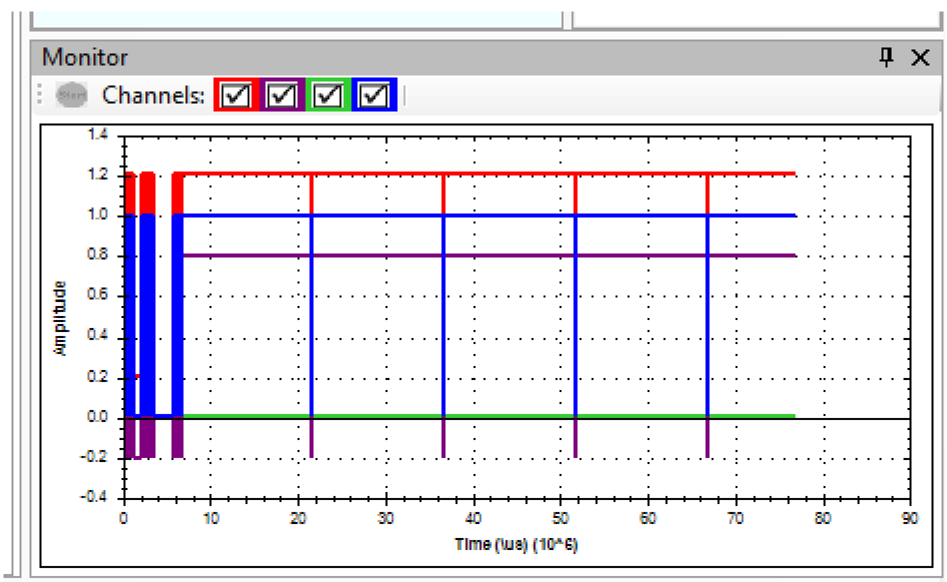

Optionally, users can upload Probes to the Scan Controller, which allow for the visualization of Laser On Monitor timings in the “Monitor Window.” Uploading probes can be performed within the Scanner Maintenance window, under the Probe Control tab. For Laser On Monitor, users should upload the “timing.lom.upper,” “timing.lom.lower,” “timing.lom.error,” and “laser.gate” probes. See Figure 5-2.

*Figure 5-2 Laser On Monitor Probes*




With the proper probes loaded, users will see the Monitor window graph update with the timings for each given signal. See Figure 5-3 as an example.

*Figure 5-3 Monitor Graph Example*



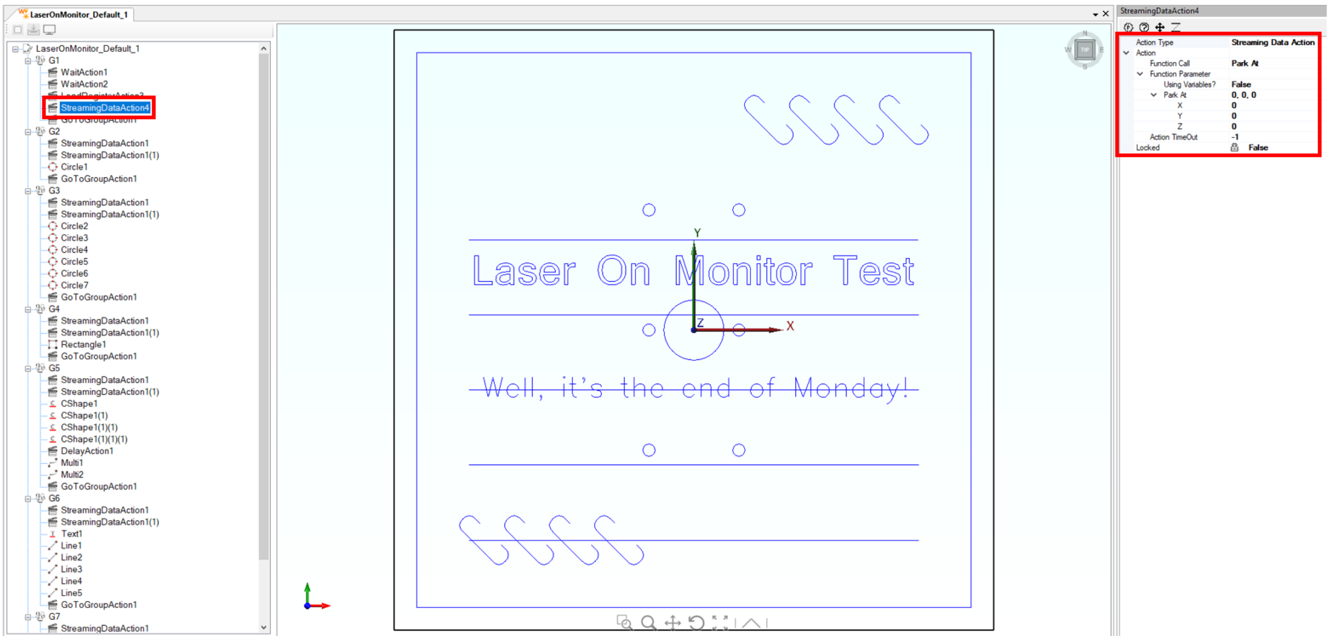

## 5.3 Job Requirements

When utilizing Laser On Monitor, users must incorporate certain Action Controls within a particular job. The following sections outline the required Action Controls for each job type.

### 5.3.1 Default Job Types

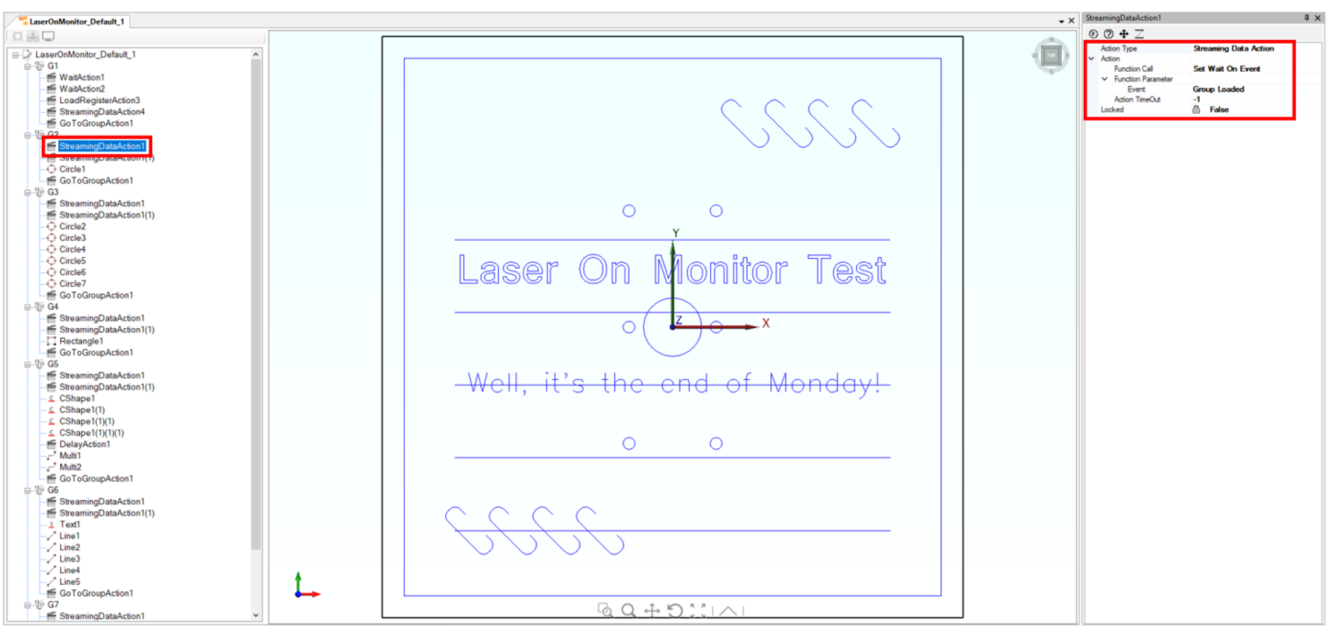
For Default jobs, users must add a non-blocking "park At" Action Control to the Loop Group. See Figure 5-4.

*Figure 5-4 Loop Group - Park At Action*




Users must also add a "Group Loaded" Action Control at the start of each Group within the job. See Figure 5-5.

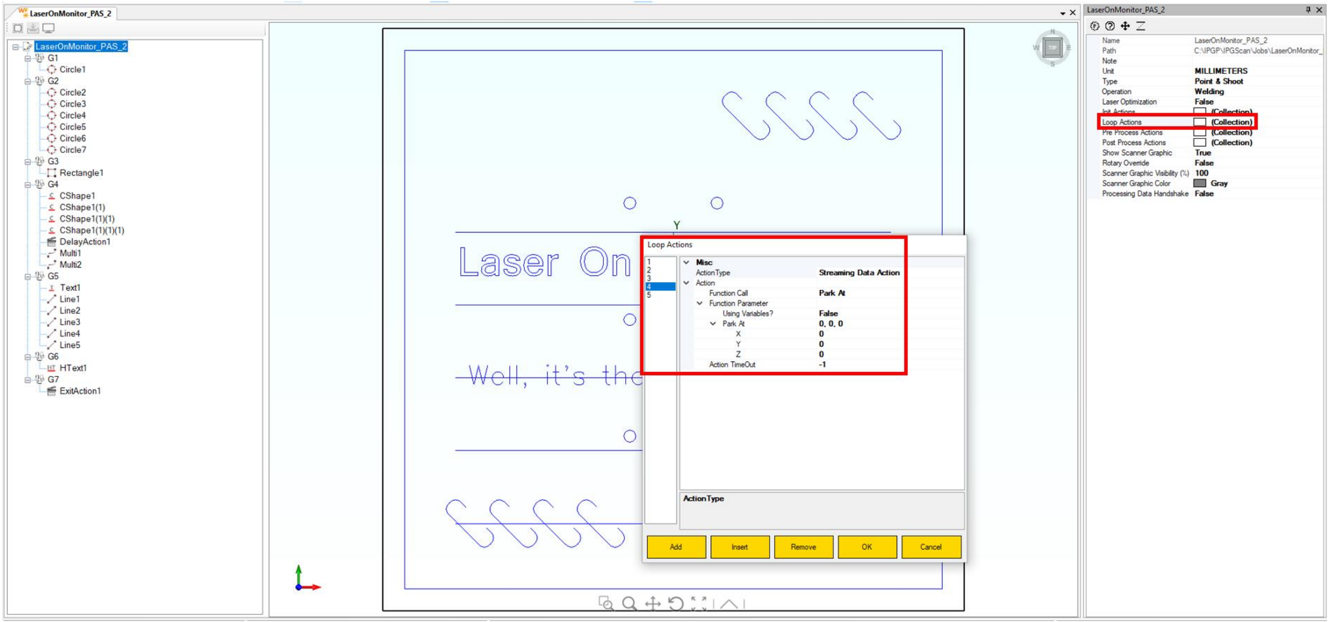
*Figure 5-5 Process Groups - Group Loaded*



### 5.3.2 Point and Shoot Job Types

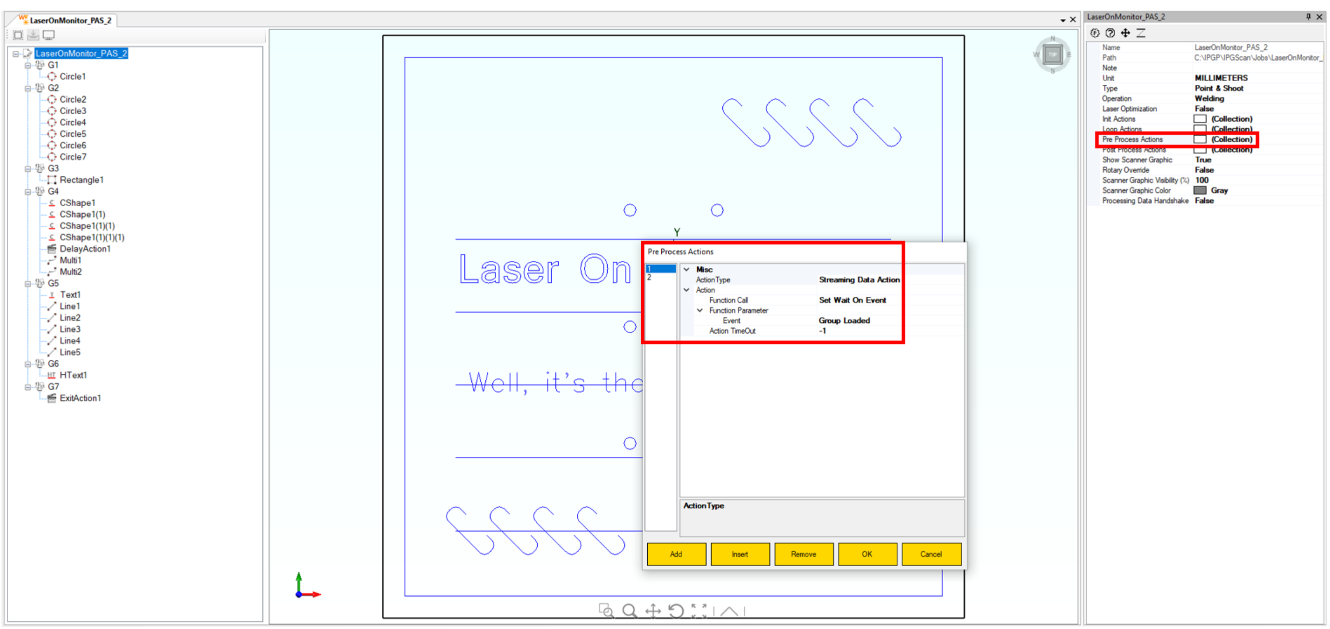
For Point and Shoot jobs, users must add a non-blocking “Park At” Action Control to the Loop Actions collection box. See Figure 5-6.

*Figure 5-6 Loop Actions Collection - Park At Action*



Users must also add a “Group Loaded” Action Control in the Pre Process Actions collection box. See Figure 5-7.

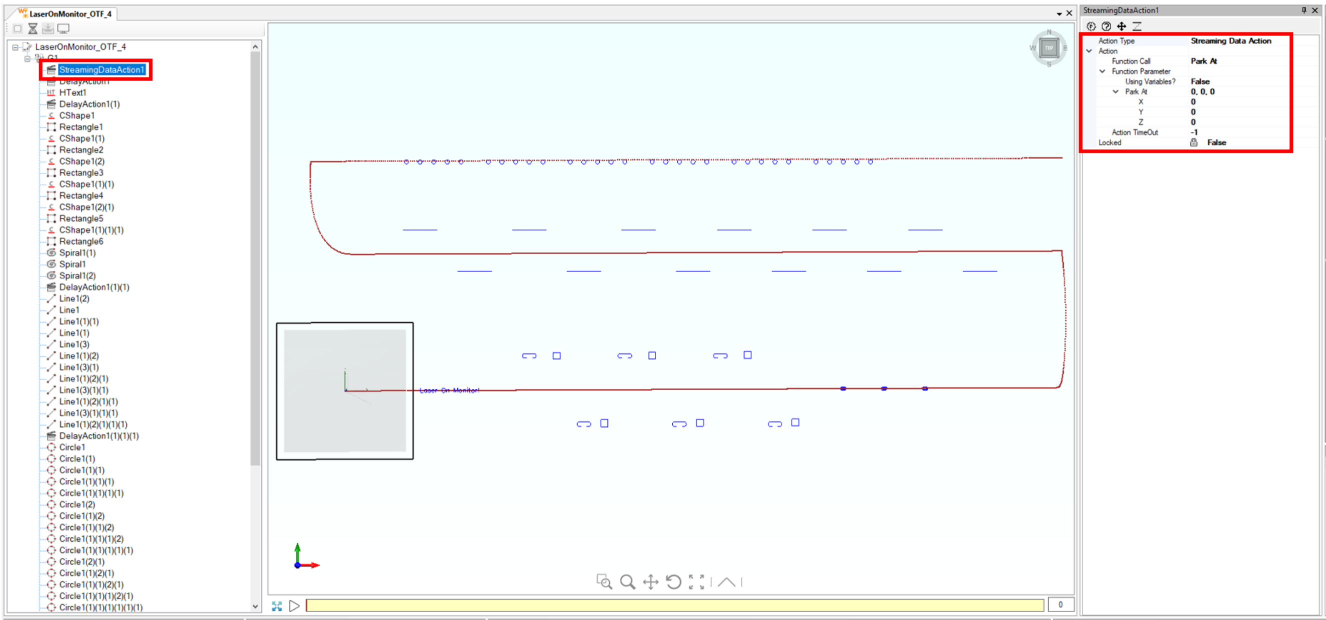
*Figure 5-7 Pre Process Actions Collection - Group Loaded*



### 5.3.3 On-The-Fly Job Types

For On The Fly jobs, users must add a non-blocking “Park At” Action Control to the top of the job. See Figure 5-8 as an example.

*Figure 5-8 Robotic On The Fly - Park At Action*

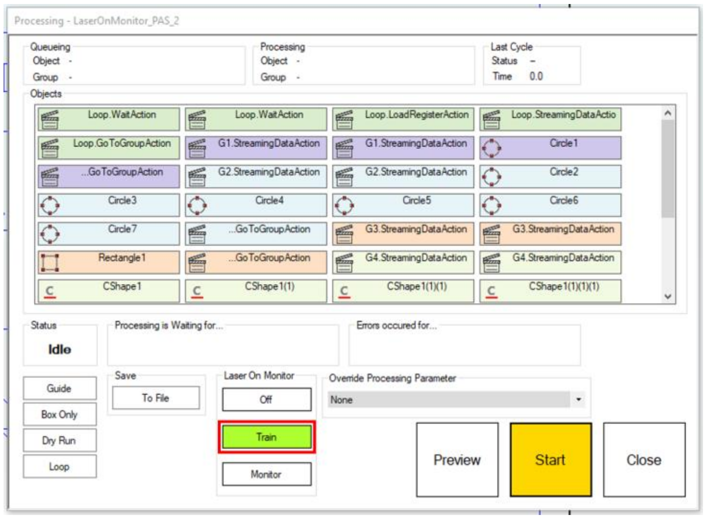



## 5.4 Training

Once users have setup a process that meets quality requirements, the next step for Laser On Monitor setup is to perform a training. Each Group within a job can be trained. In order to train Groups, users can run the entire process with Train enabled or each individual Group can be run separately. Prior to starting processing, users should enable the “Train” button in the Processing Window. See Figure 5-9.

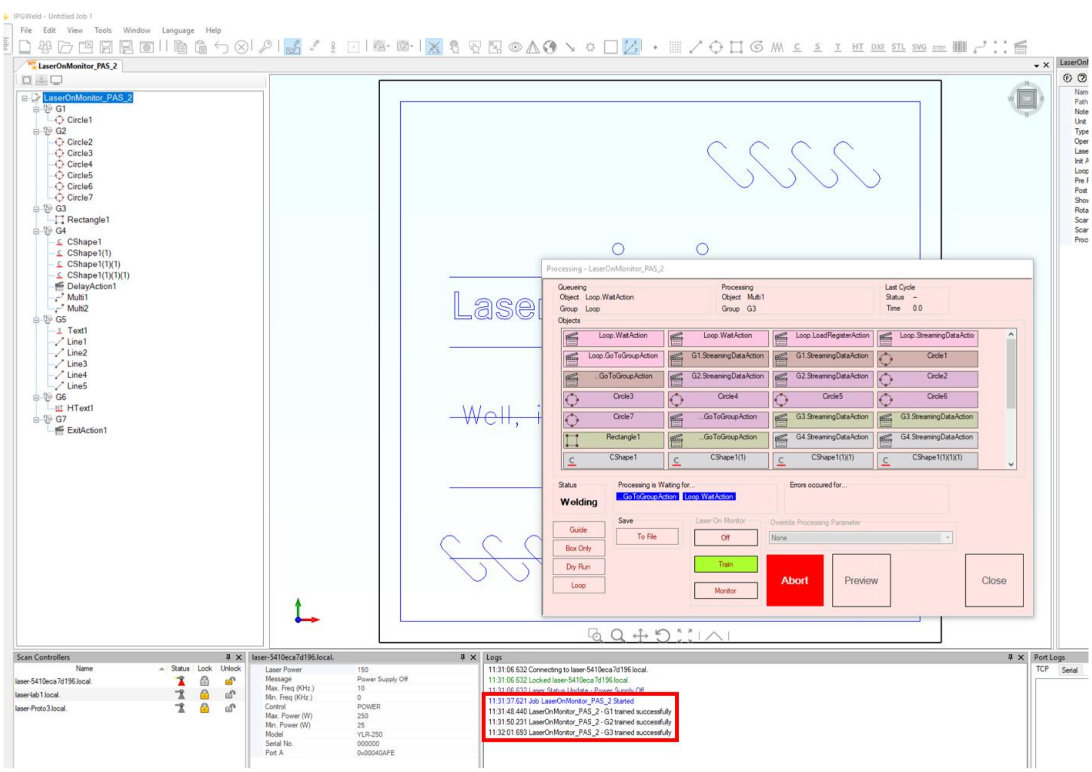
| | |
|---|---|
| **IMPORTANT** | **Running with Train enabled does not prevent the laser from firing. The purpose for this is so that users can verify the output of the process was correct during the training cycle.** |

*Figure 5-9 Training a Job/Group*




As the process is taking place, users will see that a Logs message is generated for each Group that is successfully trained. See Figure 5-10.

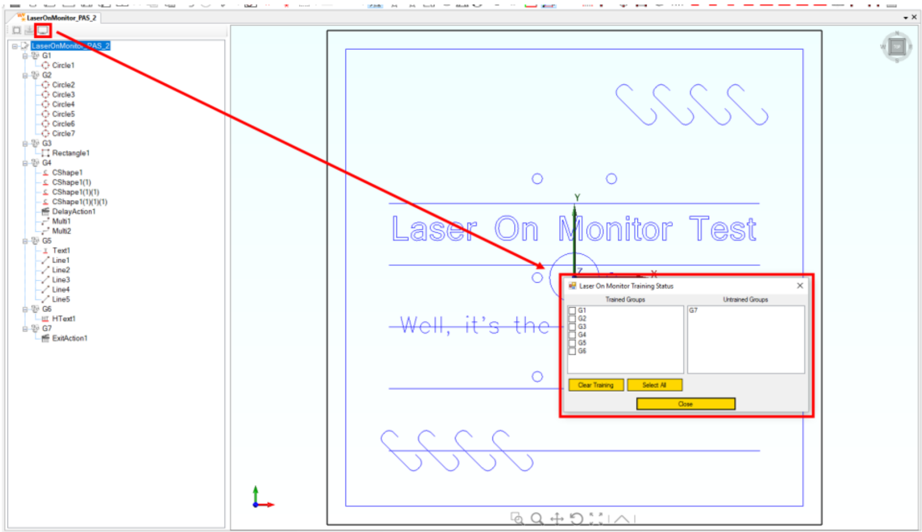
*Figure 5-10 Logs Message During Training*




Once an Exit Action Control is called for a complete Point and Shoot job or if a user only runs one Group during training, the Processing Window will close automatically once all trainings are uploaded to the Scan Controller.

To see which Groups within a job have been trained, users can click on the "Laser On Monitor Training Status" button or navigate to Tools → Scanner → Laser On Monitor Training Status. This will open the Laser On Monitor Training Status window, as seen in Figure 5-11.

*Figure 5-11 Laser On Monitor Training Status Window*




Within the Laser On Monitor Training Status window, users can select Groups within the "Trained Groups" column and clear the trainings if desired.

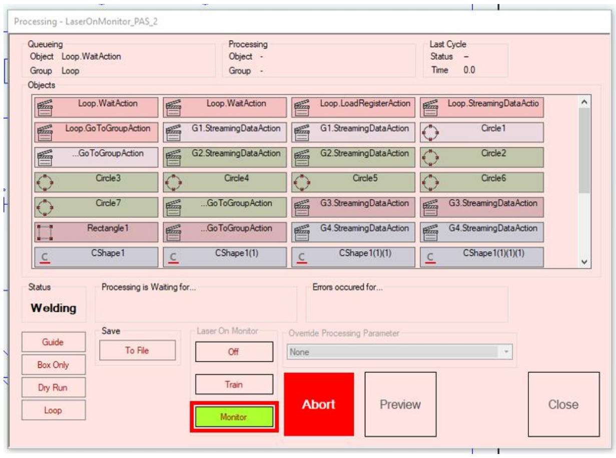
| | |
|---|---|
| **IMPORTANT** | **If users wish to retrain a job or Group, it is not necessary to clear trainings from the Laser On Monitor Training Status window.** |

Because training files are uploaded to the Scan Controller, users can close a job, reopen the job, and the training files will still exist (users will not need to retrain the job). Retraining a job would only be required if users change the job or if the job is utilized with a different Scan Controller that has not yet been trained.

## 5.5 Monitoring

After training a job or Group(s), users can enable Monitor in the Process Window and then start processing. See Figure 5-12. With Monitor enabled, the training file will be compared to the current Group being executed. If any differences exist in laser on/off timing between the training and executing Group, a Laser On Monitor error will be triggered and the process will abort.

*Figure 5-12 Enabling Monitor for Processing*




**IMPORTANT**

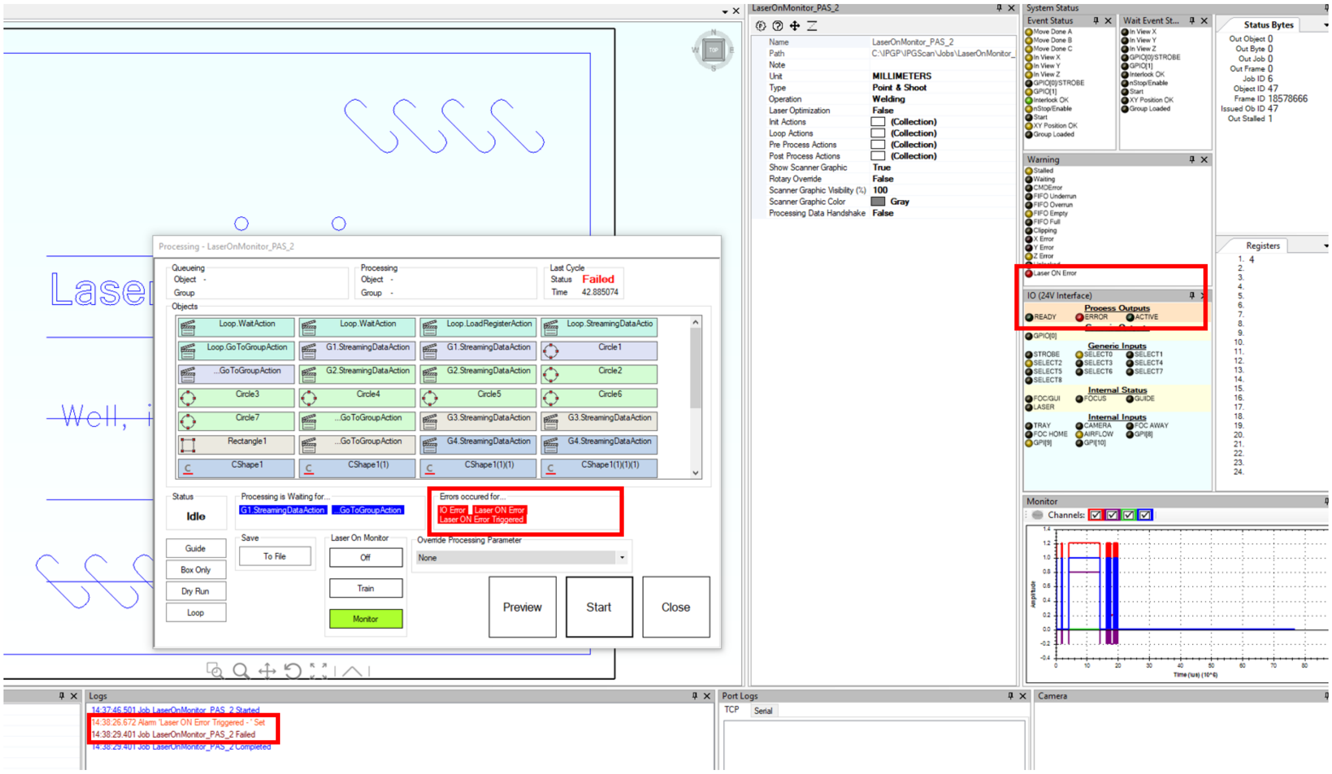
**For users that utilize the Remote API, the JobStart command can be sent with an additional argument to specify whether or not Laser On Monitor should be on/off when processing is started.**

Example Remote API Command for Laser On Monitor Off: JobStart [*Job Name*] –lomoff
Example Remote API Command for Laser On Monitor On: JobStart [*Job Name*] –lommonitor

### 5.5.1 Laser On Monitor Errors

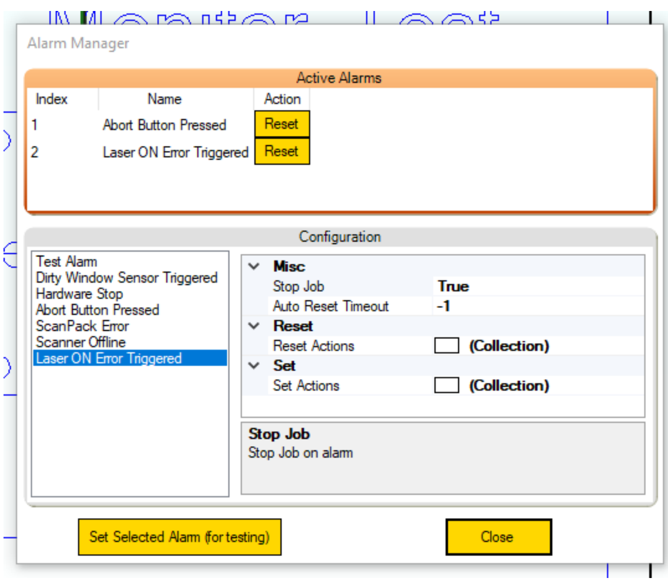
When Monitor is enabled and a change is detected in laser on/off timings, IPGScan will abort the process and the ERROR bit will be set active. Figure 5-13 Provides an example of the errors that are reported in IPGScan when a Laser On Monitor Error is triggered.

*Figure 5-13 Laser On Monitor Error Set*



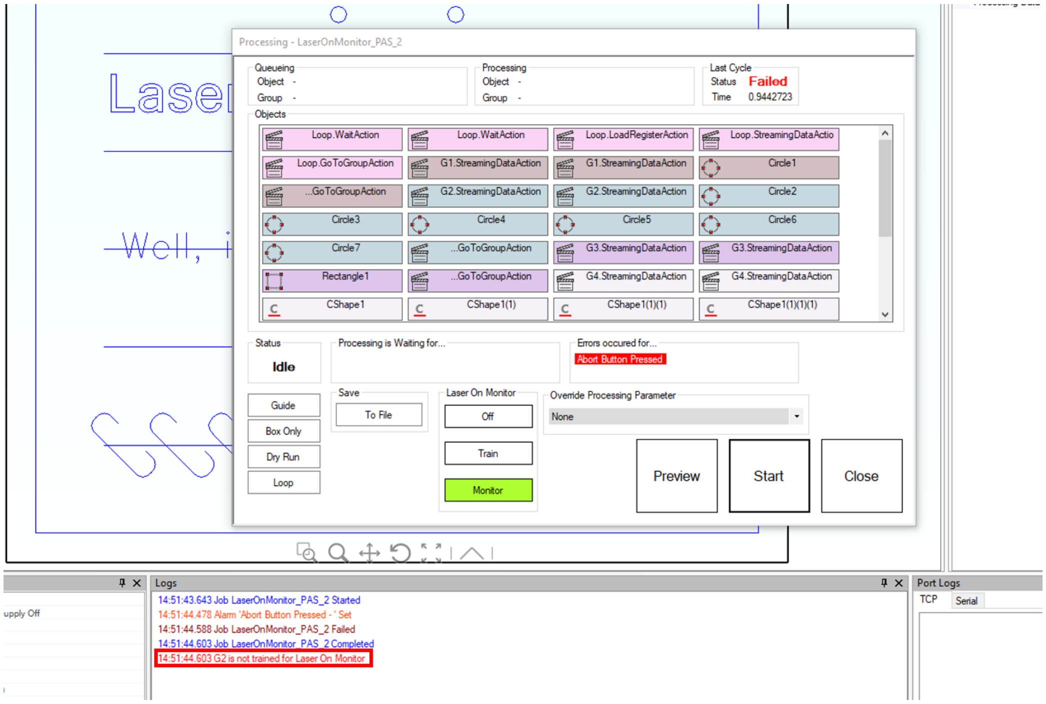
Optionally, users can setup additional actions to take place when a Laser On Monitor Error is triggered in IPGScan. For instance, users could have a General Purpose bit be set active. This can be done using the Alarm Manager. See Figure 5-14.

*Figure 5-14 Alarm Manager - Laser On Monitor Triggered*



While Monitor is enabled, if users attempt to process a Group which has not been trained, processing will be aborted and users will be presented with a message in the Logs window stating that the Group has not been trained. See Figure 5-15.

*Figure 5-15 Laser On Monitor Group Not Trained*



# 6 Maintenance Window

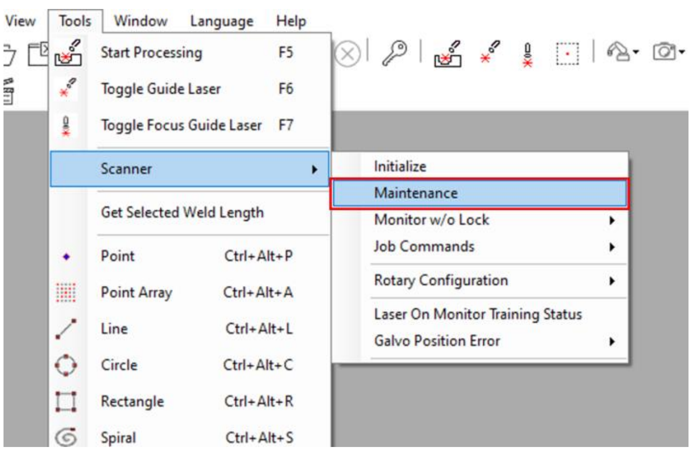
## 6.1 Overview

Within IPGScan, users can access a Scanner Maintenance window. Within this window, users can check system information as well as perform some of the basic functionality found within the Scan Controller Utility.

To access the Scanner Maintenance window, perform the following steps.

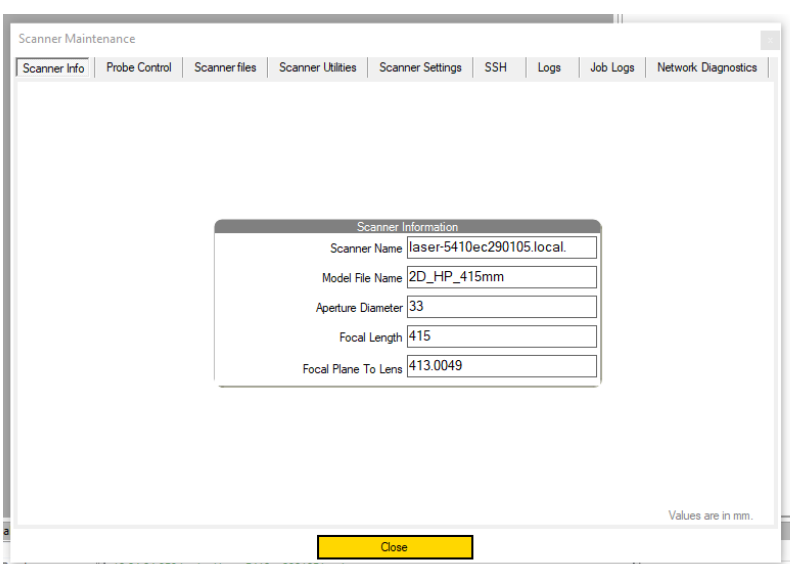
1. Open IPGScan and connect to a scanner.
2. Click “Tools.”
3. Navigate to “Scanner.”
4. Click “Maintenance.” See Figure 6-1.

*Figure 6-1 Opening the Scanner Maintenance Window*




Once the Scanner Maintenance window opens, users can begin to utilize various functions within each of the given tabs. See Figure 8-1.

*Figure 6-2 Scanner Maintenance Window*



## 6.2 Scanner Settings

Within the Scanner Settings tab, users have the ability to change the scanner name, set scanner Ethernet Adapter settings, configure a heartbeat signal, and enable Galvo Position Error.

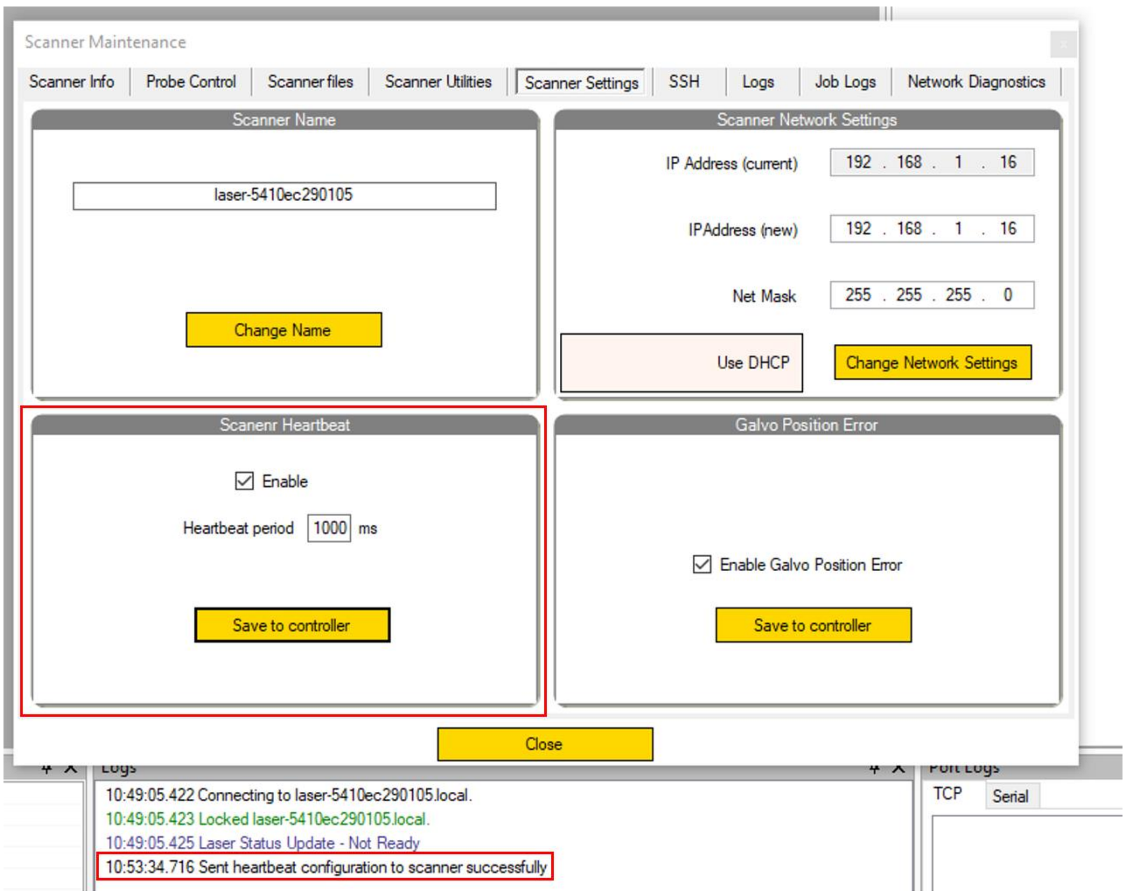
### 6.2.1 Scanner Heartbeat

Users can enable a Heartbeat signal to better help determine that the system is operational and that nothing has gone wrong.

If users wish to enable the Heartbeat signal, perform the following steps.

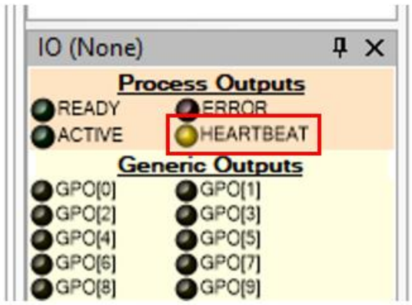
1. Check the "Enable" box.
2. Define a desired Heartbeat period.
3. Click the "Save to Controller" button. This will generate a log message stating, "Sent heartbeat configuration to scanner successfully." See Figure 6-3.

*Figure 6-3 Enabling the Heartbeat Signal*




4. Close the Scanner Maintenance Window.
5. Close and restart IPGScan. Once users reconnect to the scanner, a Heartbeat signal will be present in the IO Window of IPGScan. See Figure 6-4.

*Figure 6-4 Heartbeat Signal*




If users wish to disable the Heartbeat signal, simply uncheck the Enable box, save the settings to the controller, and restart IPGScan.

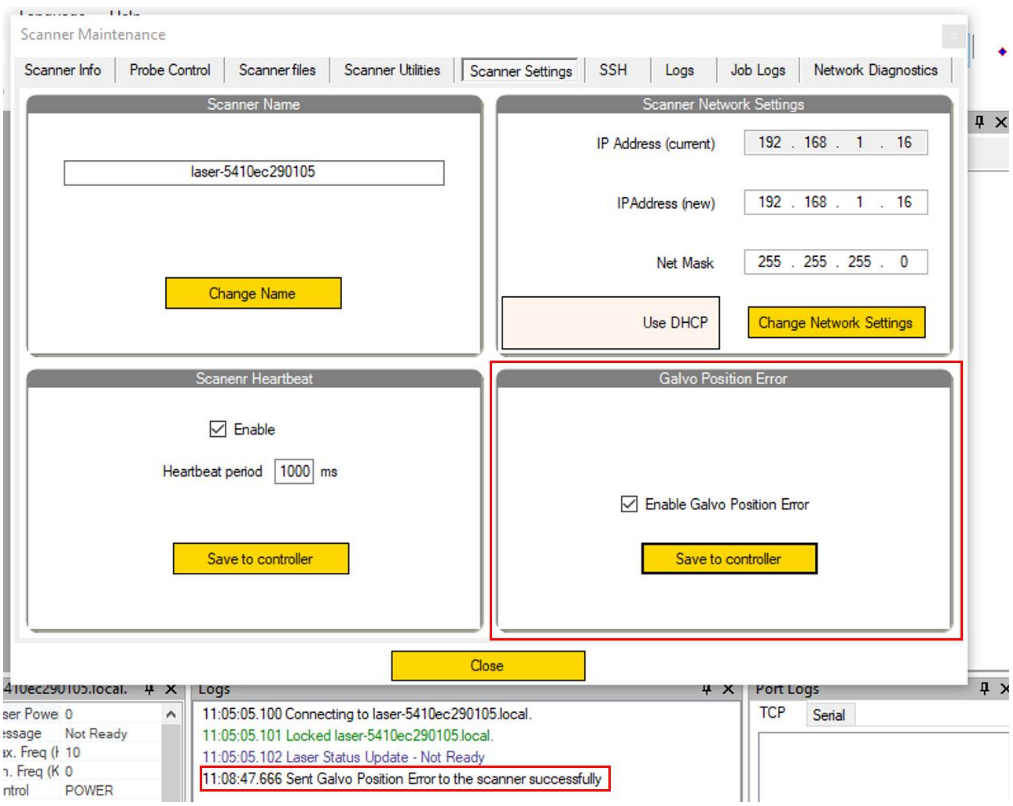
### 6.2.2 Galvo Position Error

Galvo Position Error can be enabled so that if a deviation in galvo position is detected during processing, the laser will be shutoff.

If users wish to enable Galvo Position Error functionality, perform the following steps.

1. Check the “Enable Galvo Position Error” box.
2. Click the “Save to Controller” button. This will generate a log message stating, “Sent Galvo Position Error to the scanner successfully.” See Figure 6-5.

*Figure 6-5 Enabling Galvo Position Error*



3. Close the Scanner Maintenance Window.
4. Close and restart IPGScan. Once reopened, Galvo Position Error functionality will now be active.

If users wish to disable Galvo Position Error functionality, simply uncheck the Enable Galvo Position Error box, save the settings to the controller, and restart IPGScan.

# 7 Alarm Manager

## 7.1 Overview

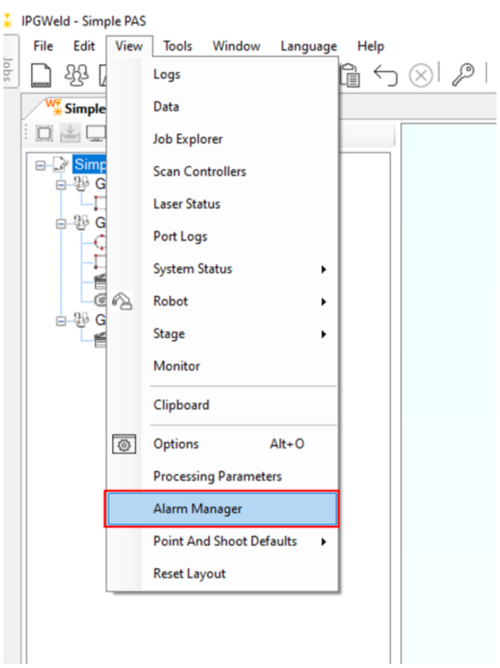
The Alarm Manger provides users with a tool to configure IPGScan to take desired actions based on alarms that may occur when operating the system. For instance, users could configure a particular error to cause a General Purpose bit to toggle active for a period of time. Another example could consist of having a prompt box appear if an error occurs. Within the Alarm Manger, users will find that actions can be configured for the following alarm conditions:

- Test Alarm – For users to test the Alarm Manager functionality
- Dirty Window Sensor Triggered – Triggered if Warning or Abort Threshold values are exceeded for the Dirty Window Sensor
- Hardware Stop – Triggered if processing is run while Enable is set inactive
- Abort Button Pressed – Triggered when the Abort button is pressed in the Processing Window
- ScanPack Error – Triggered anytime a ScanPack error occurs
- Scanner Offline – Triggered if a scanner goes offline
- Laser ON Error Triggered – Triggered when a Laser On Monitor error occurs

Users can open the Alarm Manger window by following the procedure below.

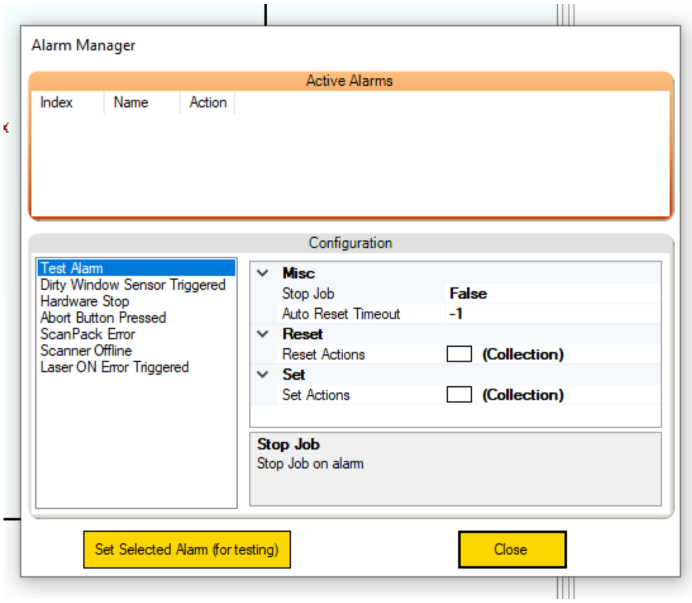
1. Click "View."
2. Click "Alarm Manager." See Figure 7-1.

*Figure 7-1 Opening the Alarm Manager Window*




Once the Alarm Manager window is open, users can select a desired alarm condition and configure any desired actions. See Figure 7-2.

*Figure 7-2 Alarm Manager Window*



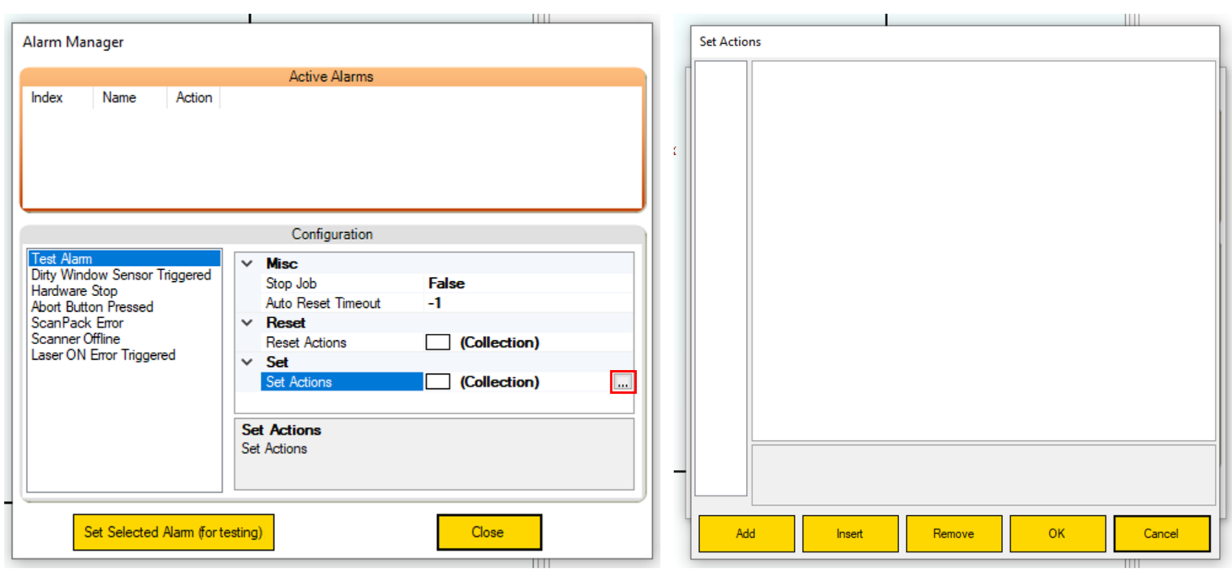

## 7.2 Set and Reset Actions

For each configurable alarm condition, users can define Set and Reset Actions.

- Set Actions – When the alarm occurs, any Action Controls defined within the Set Actions collection box will execute.
- Reset Actions – When the alarm is reset, any Action Controls defined within the Reset Actions collection box will execute.

To setup Action Controls for Set and Reset Actions, simply open the collection box associated with either action. See Figure 7-3.

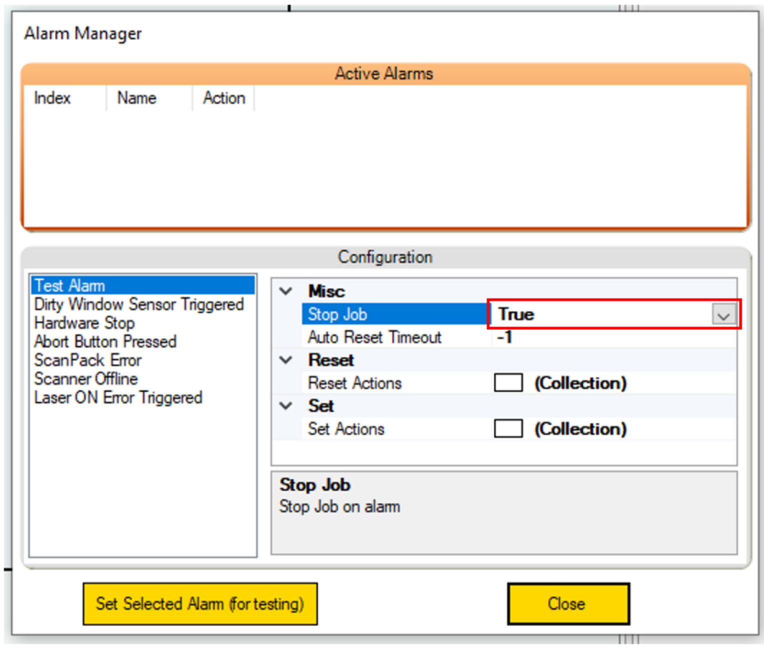
*Figure 7-3 Opening Set or Reset Actions Collection Boxes*



Within the Set Actions or Reset Actions collection box window, users can Add, Insert, and Remove any desired Action Controls that will be executed when the given action occurs. For additional detail on Action Controls, refer to section "Action Controls."

Finally, if users wish for a particular error to cause processing to be aborted when the error occurs, set the "Stop Job" option to "True." See Figure 7-4.

*Figure 7-4 Alarm Manager Stop Job Setting*

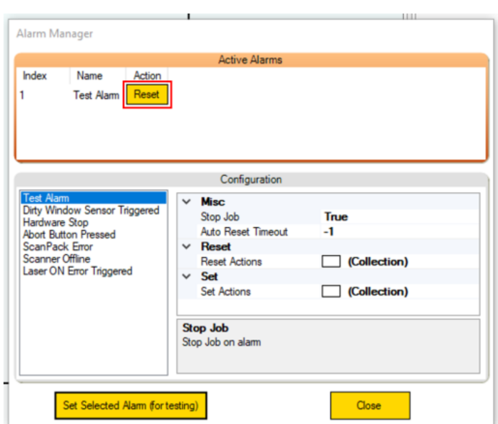



## 7.3 Resetting an Alarm

If an alarm occurs, users have a few methods available to reset the alarms. These methods are as follows.

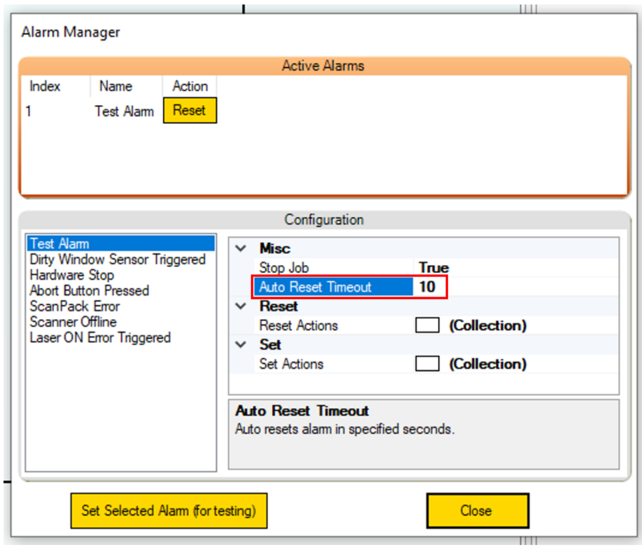
1. Reset via the Alarm Manager Window
    a. Users can click the "Reset" button for a given alarm within the Alarm Manger window. See Figure 7-5.

*Figure 7-5 Resetting an Alarm in the Alarm Manager Window*



2. Auto Reset Timeout
    a. Users can define a time period for which the alarm condition will remain active until IPGScan automatically resets the alarm. Simply define a period of time in seconds for the "Auto Reset Timeout" parameter. See Figure 7-6.

*Figure 7-6 Alarm Manager Auto Reset Parameter*



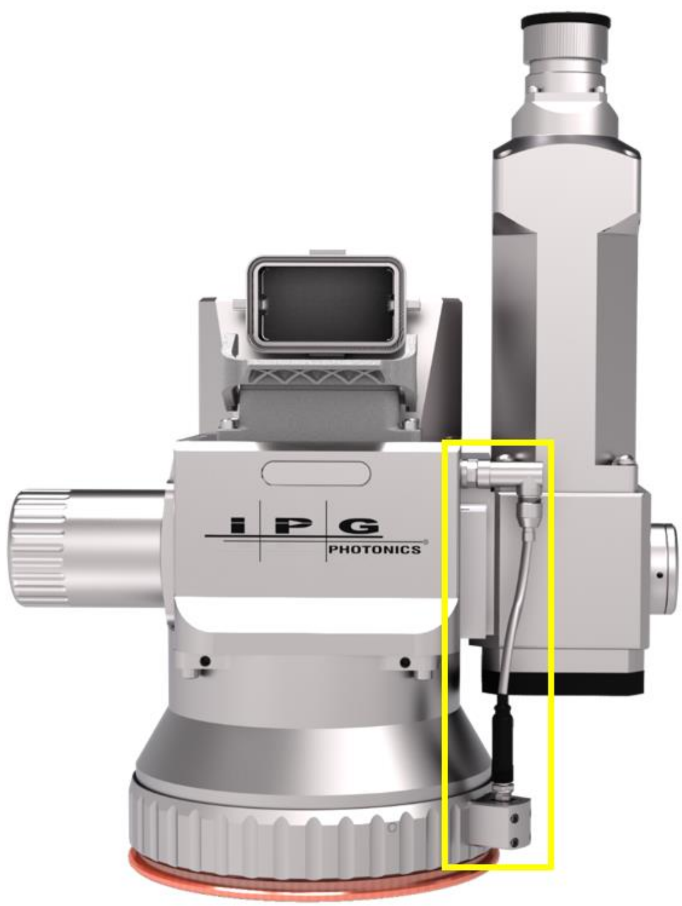

3. Remote API Command "SystemResetAllAlarms"
    a. Via the Remote API users can send the "SystemResetAllAlarms" command. See section "Remote API" for additional details.

# 8 Dirty Window Sensor

## 8.1 Overview

IPG Scan heads designed with the Dirty Window Sensor (DWS) detect scattered light caused by the presence of contamination on the protective window assembly (see Figure 8-1). This hardware and software implementation allows the User to set thresholds for actions taken depending on how much scattered light is detected. The first threshold being exceeded will trigger an error message, but not halt operation. A second higher threshold can also be set up to terminate the job immediately. This tiered approach allows the user to get a message in IPGScan when the window begins to become contaminated, and only abort the job at a much higher level. It is up to the user to determine these thresholds and which action (or both) the system should take when the thresholds are exceeded.

*Figure 8-1 2D High-Power Scanner with Dirty Window Sensor*




All sensor readings are in dBuA, and the max/min range is approximately +45dBuA to -65dBuA. This is a logarithmic scale (i.e. for every change of 20dBuA, the scattered light changes by a factor of 10). See examples below:

40dBuA = 100uA
20dBuA = 10uA
0dBuA = 1uA

…
-60dBuA = 1nA

## 8.2 Sensor Status Window

The following sections detail how to open the DWS Status Window and the statuses contained within the window.

### 8.2.1 Opening the Dirty Window Sensor Status Window

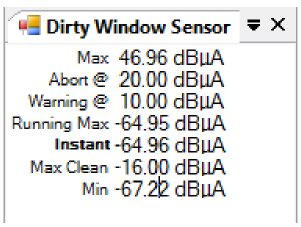
To open the DWS Status Window in IPGScan, perform the following steps:

1. Click “View.”
2. Cursor down to “System Status.”
3. Click “Dirty Window Sensor.”

### 8.2.2 Dirty Window Sensor Statuses

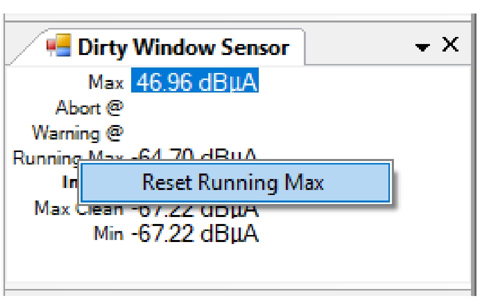
Figure 8-2 outlines the statues contained within the DWS Status Window. A description for each status is contained below.

*Figure 8-2 Dirty Window Sensor Status Window*




1. **Max** – The maximum value the DWS is capable of producing.
2. **Abort @** - The level that will trigger an immediate abort of the running IPGScan job. This value is defined by the user.
3. **Warning @** - The level that will trigger a message error (and optionally a pop-up window) notifying the user that the Warning Threshold has been exceeded. This value is defined by the user.
4. **Running Max** – The highest reading the DWS measured since the last time this value was cleared (by the user). To reset the Running Max field, right click the “Running Max” text in the DWS Status Window and then click “Reset Running Max.” See Figure 8-3. Alternatively, the Remote API has a command that can be used to reset the Running Max value using TCP/IP communication.

*Figure 8-3 Reseting the Running Max Value*




5. **Instant** – The last reading taken. This value is updated approximately every 250ms.

6. **Max Clean** – The maximum value obtained when the User Measures a clean window. This value is input by the user and should be based on the parameters that will be used during a production process.
7. **Min** – The minimum value the DWS is capable of producing.

## 8.3 Determining Max Clean, Warning Threshold, and Abort Threshold Values

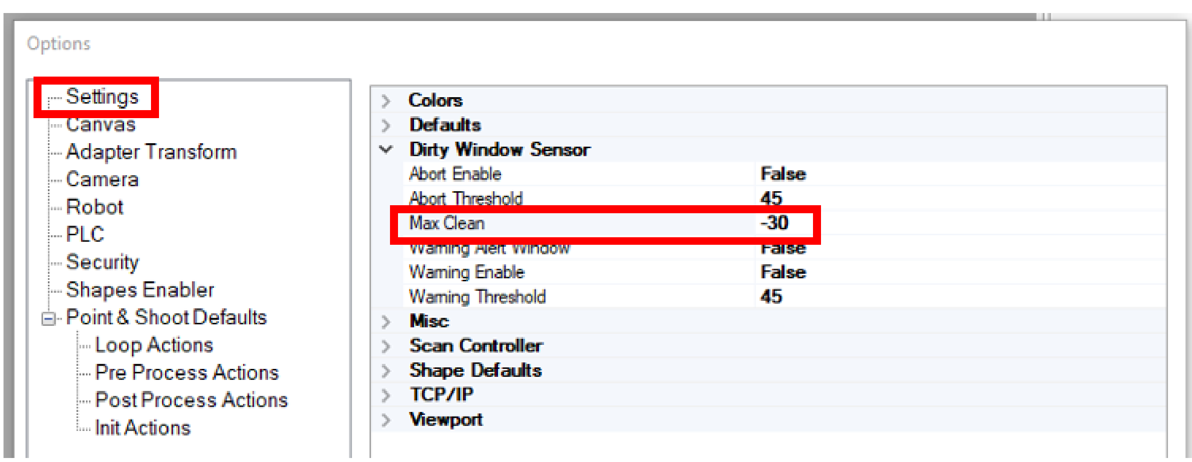
Given that the DWS measures scattered light and that every user application is different, it is imperative that the DWS be setup specifically for each application. Furthermore, the DWS should be setup when process development is complete, given that many different process parameters can have an effect on scattered light (i.e. material type, process speeds, laser power, fixturing, plume suppression, etc.). Once process development is complete, users can begin the process of determining Max Clean, Warning Threshold, and Abort Threshold values for DWS setup. The following procedure outlines how users can determine each of these values.

1. Ensure that process development is complete and that all process parameters are as they will be during the production process.
    a. Example of process parameters includes laser power, material type, process speeds, focus position, processing positions, plume suppression, etc.
2. Insert a new Protective Window into the Window Assembly.
    a. Please refer to the Scanner Series User Manual for details on how to replace a Protective Window.
3. Reset the "Running Max" value in IPGScan.
    a. Please refer to section 8.2.2 for information on how to reset the Running Max value.
4. Cycle the system one time as it is expected to run in production. The system must run with the laser enabled in order for the DWS to capture scattered light as it will during production.

**IMPORTANT** **Ensure that all laser safety practices are followed when firing the laser.**

5. Once the process is complete, record the value listed for "Running Max" in the DWS Status Window.
6. Open the IPGScan "Options" window.
7. In the Dirty Window Sensor settings, enter the value recorded in step 5 into the "Max Clean" field. See Figure 8-4 as an example.

*Figure 8-4 DWS Max Clean Value*

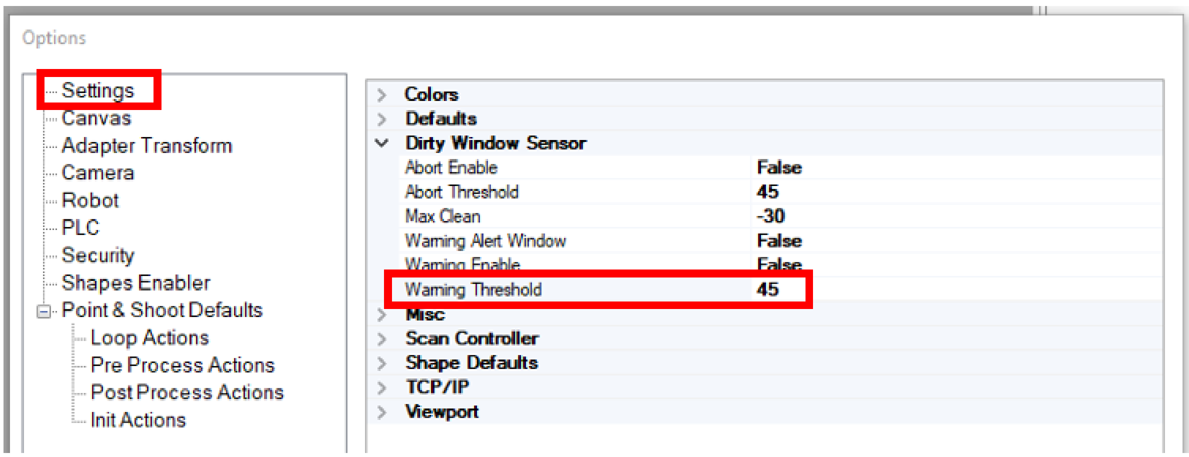



8. Next, run the system as it is expected to run in production. The process/part quality should be monitored and the Protective Window should be periodically examined for damage. Eventually,

it is likely that the Protective Window will become damaged to the point that the process/part quality begins to degrade. When this is noticed, record the "Running Max" value that is currently listed in the DWS Status Window in IPGScan.

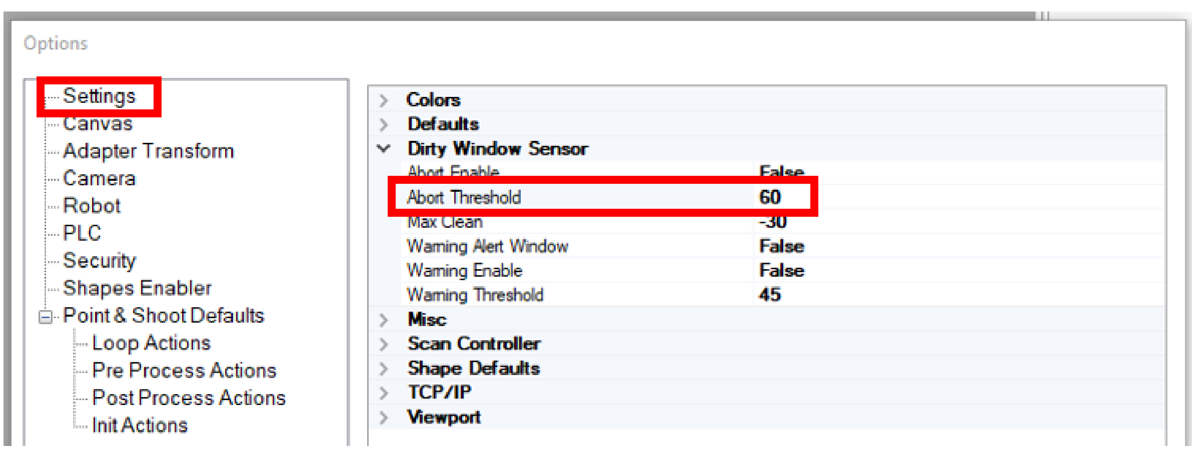
9. Open the IPGScan "Options" window.
10. For the "Warning Threshold" value, enter a value that is less than the "Running Max" value that was recorded in step 8. For instance, users could start by subtracting 15 dBuA from the "Running Max" value that was recorded. See Figure 8-5 as an example.

*Figure 8-5 DWS Warning Threshold*



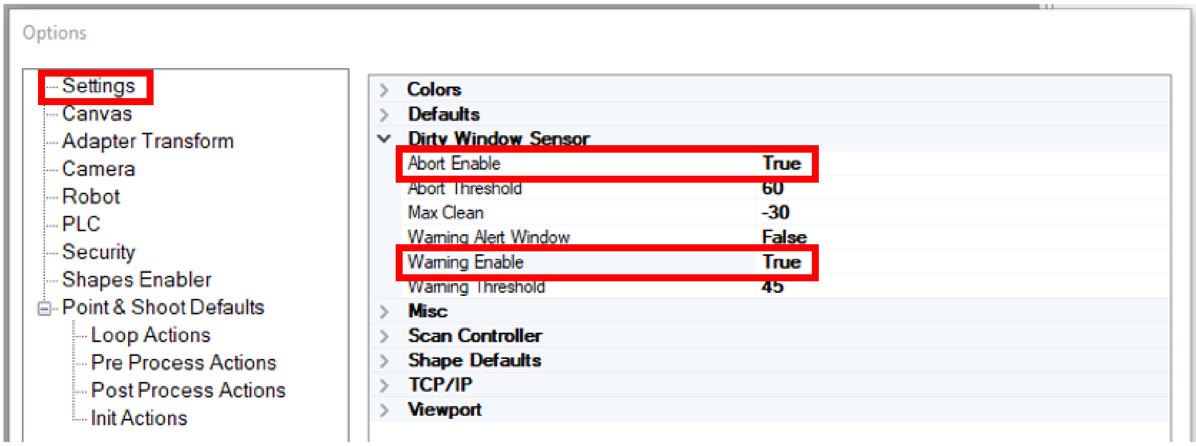

11. For the "Abort Threshold" value, enter the value recorded in step 8 for the "Running Max" value or a value that is slightly higher. See Figure 8-6 as an example.

*Figure 8-6 DWS Abort Threshold*




12. Enable the DWS Abort and Warning functions as desired. Figure 8-7 outlines both the Abort and Warning functions being enabled. The following outlines the expected behavior of IPGScan for each function when it is enabled.
    a. Abort – When enabled and if the "Abort Threshold" is exceeded, IPGScan will abort processing.
    b. Warning – When enabled and if the "Warning Threshold" is exceeded, IPGScan will not abort the process but will generate a warning message.

*Figure 8-7 Enabling the Abort and Warning DWS Functions*




13. Click "Ok" to close the IPGScan "Options" window.
14. With the Warning and Abort Thresholds defined and each function enabled, users can then run their process as desired in production. As the functions are triggered, users should examine their process/part quality as well as the coverslide contamination. If the alarms seem to be too early (there's little coverslide damage and/or part quality is ok) or too late (major coverslide damage or poor part quality) then adjustments can be made to the threshold values to further fine tune the process.

**IMPORTANT** **Keep in mind that the DWS is not intended to be a process quality or monitoring solution. For process quality monitoring, LDD should be utilized.**

### 8.3.1 Warning Alert Window

Users have the ability to enable a Warning Alert Window for when the Abort Threshold or Warning Threshold values are exceeded. When enabled, this will generate a pop-up window within IPGScan that provides the reading of the DWS. Users can enable this functionality by performing the following steps.

1. Open the IPGScan "Options" window.
2. Select "Settings."
3. Navigate to the Dirty Window Sensor settings.
4. Set "Warning Alert Window" to "True." See Figure 8-8.

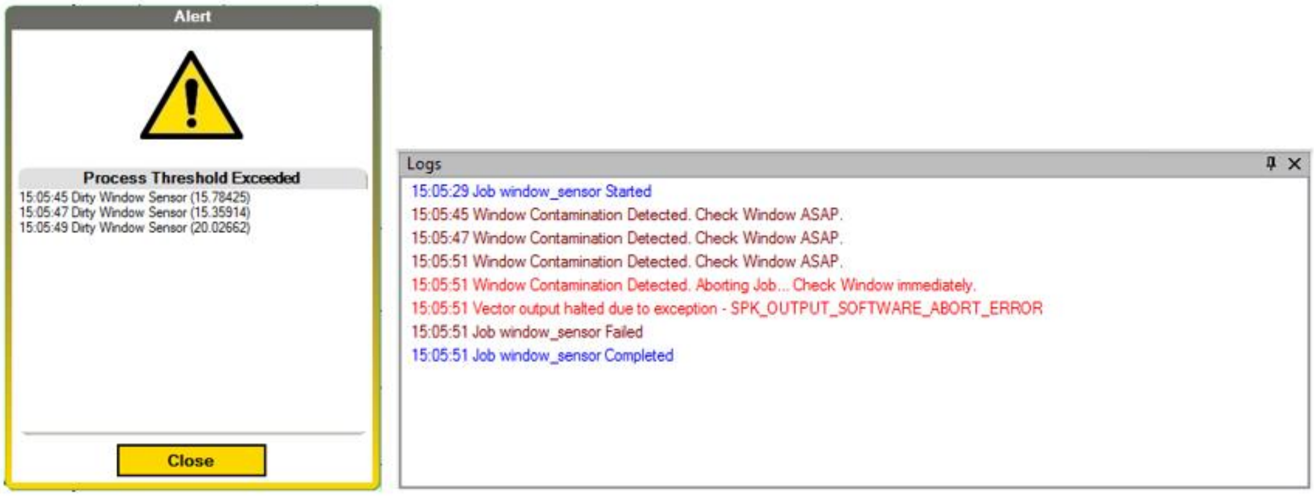
*Figure 8-8 DWS Warning Alert Window Option*


When the Warning Threshold is exceeded and the Warning Alert Window is enabled, uses will be presented with a pop-up and the Logs window will also provide a status message. See Figure 8-9. If the Warning Alert Window is disabled and the Warning Threshold is exceeded, users will only be presented with a message in the Logs window.

*Figure 8-9 Monitoring Alert and Logs Windows - Warning Threshold Exceeded*


When the Abort Threshold is exceeded and the Warning Alert Window is enabled, uses will be presented with a pop-up and the Logs window will also provide a status message. See Figure 8-10. If the Warning Alert Window is disabled and the Abort Threshold is exceeded, users will only be presented with a message in the Logs window.

*Figure 8-10 Monitoring Alert and Logs Windows - Abort Threshold Exceeded*



## 8.4 Configuration for Use in an Automated System

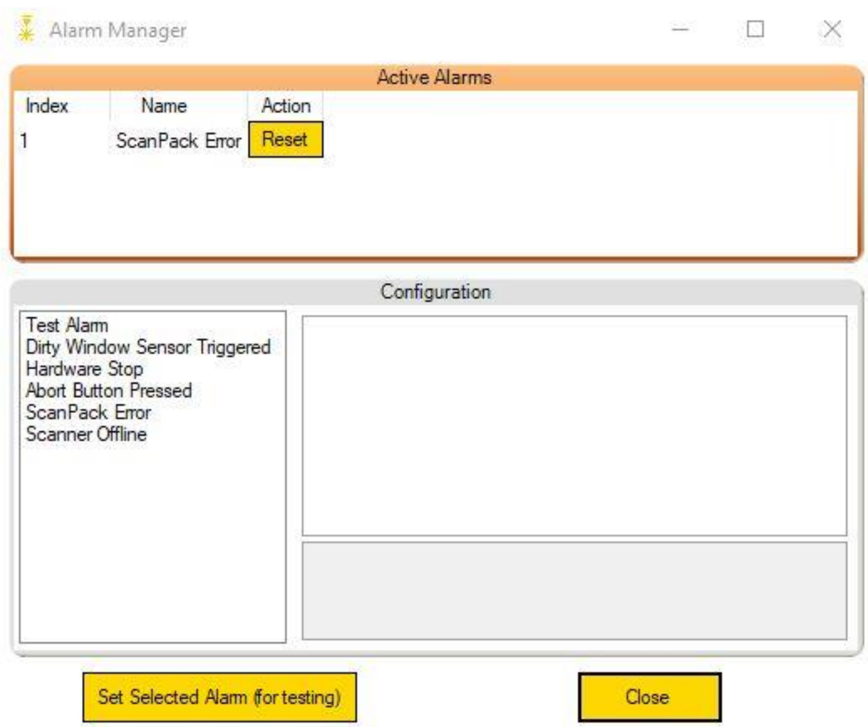
There are multiple ways in which a users automated system (PLC or Robot Controller) can detect that a DWS alarm has been triggered. The following sections outline the various methods for which this can be done.

### 8.4.1 IPGScan Alarm Manager Implementation

The following procedure outlines how a user can configure a digital bit using the Alarm Manager in IPGScan to reflect the status of the Dirty Window Sensor "Warning" or "Abort" alarms.

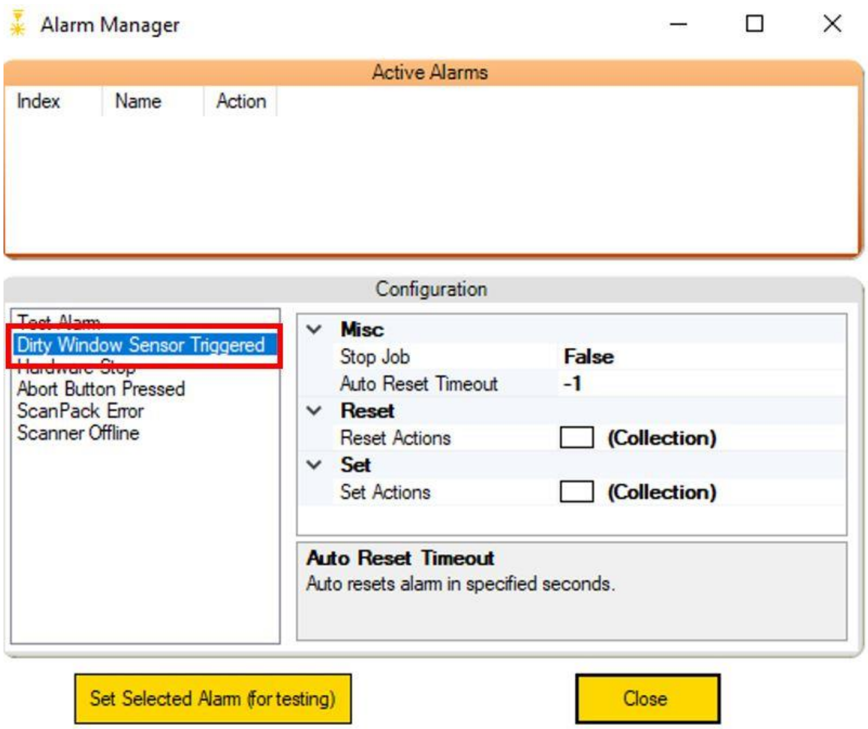
1. In IPGScan, click "View" in the tool bar.
2. Click "Alarm Manager." This will open the Alarm Manager window. See Figure 8-11.

*Figure 8-11 Alarm Manager Window*



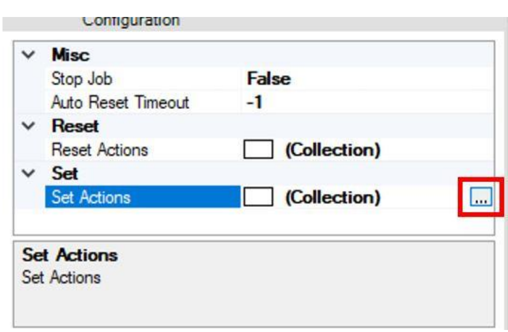

3. Select "Dirty Window Sensor Triggered" in the leftmost configuration list. See Figure 8-12.

*Figure 8-12 Dirty Window Sensor Alarm*



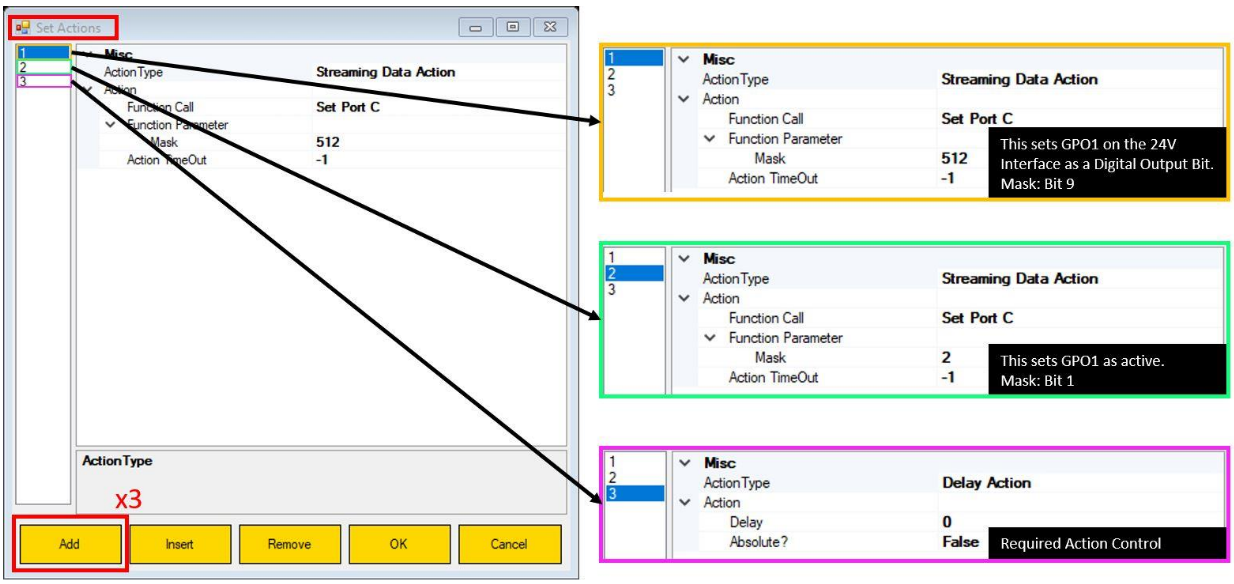

4. Specify the proper configuration settings for "Stop Job."
    a. True: Both "Warning" and "Abort" alarms will cause the running IPGScan job to abort (whether currently processing or not).
    b. False: The "Warning" alarm will not cause the running IPGScan job to abort but the "Abort" alarm will still cause the running IPGScan job to abort.
5. Specify a desired "Auto Reset Timeout" value.
    a. -1: Auto reset will not occur after a given period of time. The user would be required to open the Alarm Manager in IPGScan and reset the active error by clicking on "Reset."
    b. n > 0: The alarm will be reset after the users specified period of time has elapsed.
6. Open the "Set Actions" collection box. See Figure 8-13.

*Figure 8-13 Opening the Set Actions Collection Box*



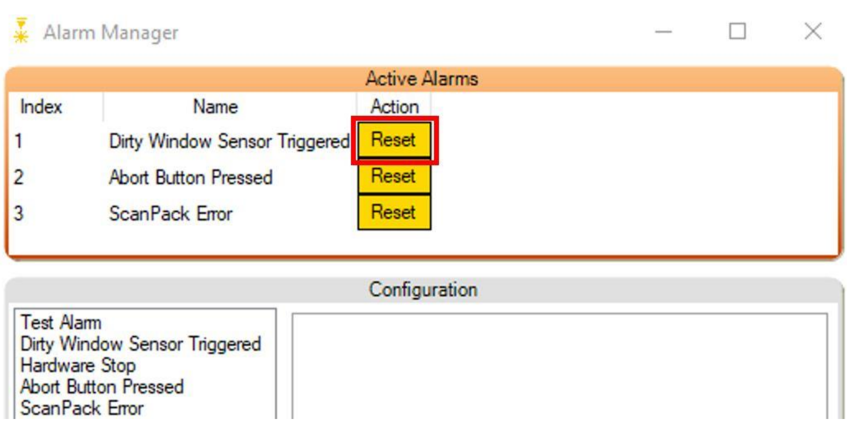
7. Add any desired Action Controls to the collection. These Action Controls will be executed in sequential order when the alarm condition is raised (i.e. the “Warning” or “Abort” alarms occur).
    a. See Figure 8-14 for a list of Action Controls that are used to turn the GPO1 bit active on the 24V Interface when the alarm condition occurs.

*Figure 8-14 Configuring Set Actions*



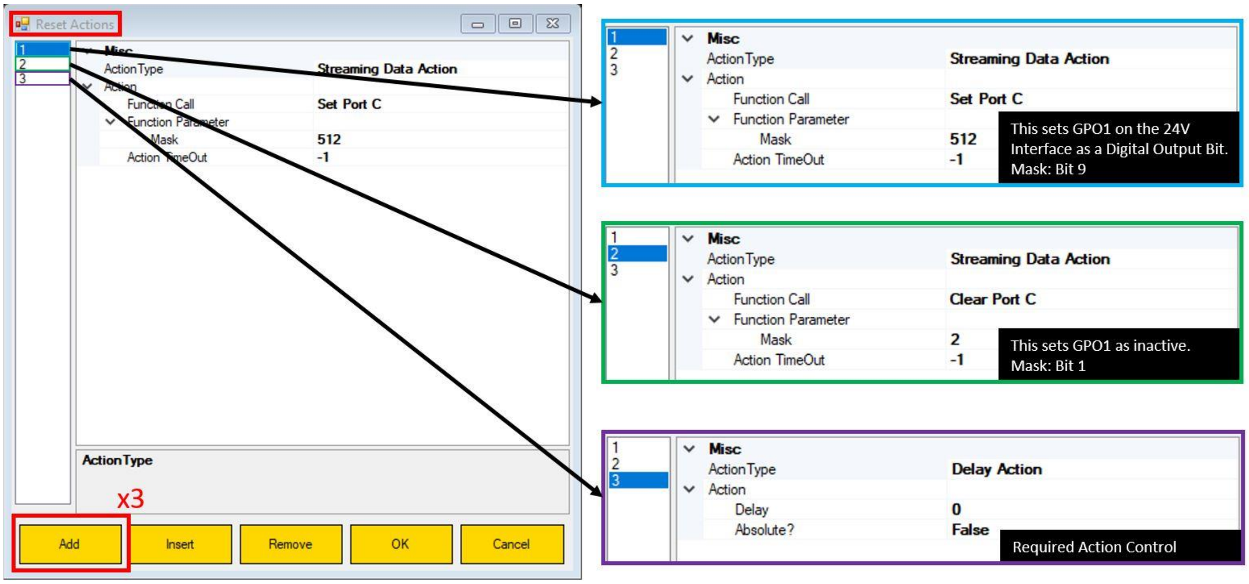

8. Click “Ok” once all desired Action Controls have been added to the sequence.
9. Open the “Reset Actions” collection box.
10. Add any desired Action Controls to the collection. These Action Controls will be executed in sequential order when the alarm condition is reset.
    a. The user can reset an alarm condition using either of the following methods:
        i. By specifying an “Auto Reset Timeout” value greater than 0. See step 5.
        ii. By clicking “Reset” on an error in the Alarm Manager window. See Figure 8-15.

*Figure 8-15 Resetting an Alarm in the Alarm Manager Window*



b. See Figure 8-16 for a list of Action Controls that are used to turn the GPO1 bit inactive on the 24V Interface when the alarm condition is reset.

*Figure 8-16 Configuring Reset Actions*




11. Click "Ok" once all desired Action Controls have been added to the sequence.
12. Click "Close" to close the Alarm Manager window.
13. The Alarm Manager should now be configured to behave in the desired manner based on a specific alarm condition.
    a. If the "Set Actions" and "Reset Actions" were configured as outlined in this example, the user should notice the following behavior.
        i. When a "Warning" or "Abort" alarm is encountered on the DWS, the GPO1 bit on the 24V Interface will go active.
        ii. When the alarm is reset, the GPO1 bit on the 24V Interface will go inactive.

### 8.4.2 Remote API Implementation

Users have the option of utilizing the Remote API to monitor the DWS statuses. When utilizing the Remote API for monitoring the DWS, users can optionally enable the Abort and Warning functions in IPGScan. If users choose to not enable the Abort and Warning functions, it still is possible to get DWS status values. The following Remote API commands exist for the DWS.

DWSResetRunningMax

**Parameters**: none

**Returns**: "DWSRunningMaxReset"

**Description**: Resets Dirty Window Sensor's running Max value.

**Error**: none

**Troubleshooting**: none

**Example**:
S: DWSResetRunningMax<CR><LF>
I: Running Max value is reset in the IPGScan DWS Status

Window.
R: DWSRunningMaxReset<CR><LF>

DWSGetRunningMax

**Parameters**: none

**Returns**: string containing current Running Max value in IPGScan DWS Status Window.
Example: "-67.22169"

**Description**: The Running Max Value displayed in the IPGScan DWS Status Window.

**Error**: none

**Troubleshooting**: none

**Example**:
S: DWSGetRunningMax<CR><LF>.
R: *RunningMaxValue*<CR><LF>

DWSGetInstantValue

**Parameters**: none

**Returns**: string containing current Instant value in IPGScan DWS Status Window.
Example: "-30.56"

**Description**: The Instant Value displayed in the IPGScan DWS Status Window.

**Error**: none

**Troubleshooting**: none

**Example**:
S: DWSGetInstantValue<CR><LF>
R: *InstantValue*<CR><LF>

Through the use of the above commands, users can monitor (typically using a PLC) the Instant value and the Running Max value. In doing so, users can create PLC code that dictates how the cell should behave based on the readings that are gathered. For instance, when a new Protective Window is installed in the head, the PLC would send the "DWSResetRunningMax" command. While production is running, the PLC could poll the "DWSGetRunningMax" and/or "DWSGetInstantValue" at a desired rate. If the determined contamination level is then exceeded, the PLC could require that someone enter the cell and replace the Protective Window prior to the system cycling again. Ultimately, the DWS Remote API commands allow users to tailor the system behavior according to their desires.

# 9 Remote API

## 9.1 Overview and Configuration

This section describes the TCP Remote Application Programming Interface (API) commands for controlling IPGScan externally. The commands are strings that are sent through a TCP/IP connection to IPGScan, so the software can respond accordingly. The strings are encoded based on the encoding setting set within in the IPGScan Options.

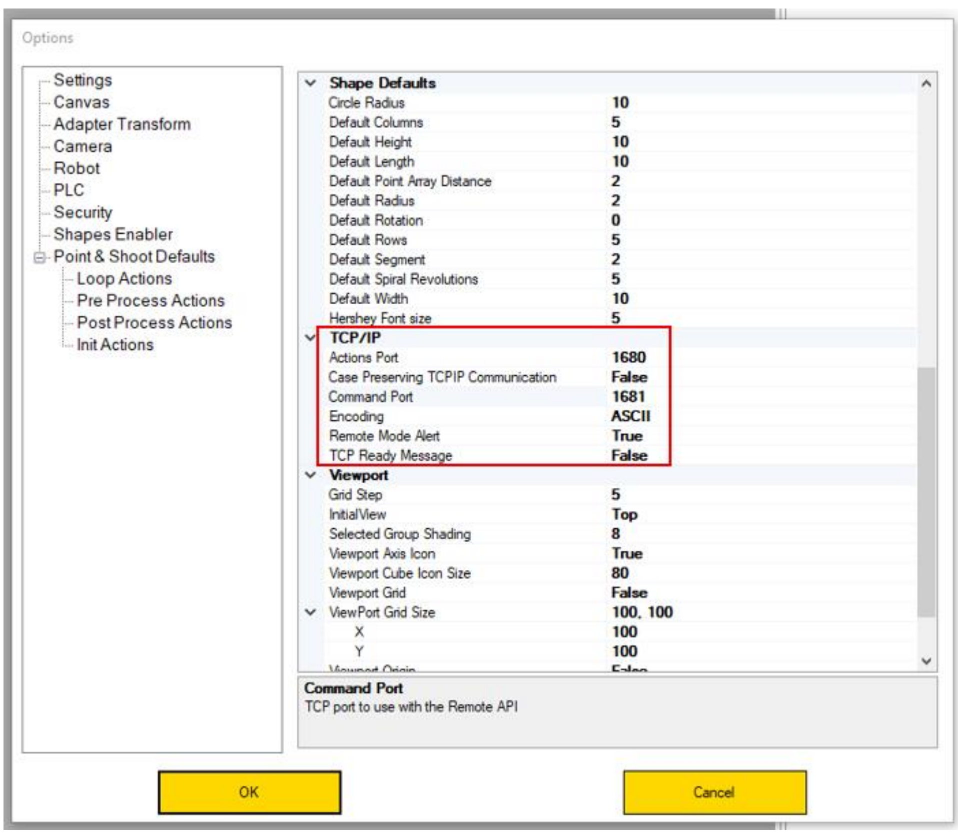
All commands should be followed by a carriage return (ASCII #13) and a line feed (ASCII #10). For example: JobOpen *jobname*<CR><LF>.

Prior to sending any commands, a TCP connection between the computer running IPGScan and the device trying to control it must exist. In this case, IPGScan will behave as a Server while the external device will be the Client requesting a connection to IPGScan.

Prior to utilizing the Remote API, users must configure the appropriate TCP/IP settings in IPGScan. To define the TCP/IP, refer to the following steps.

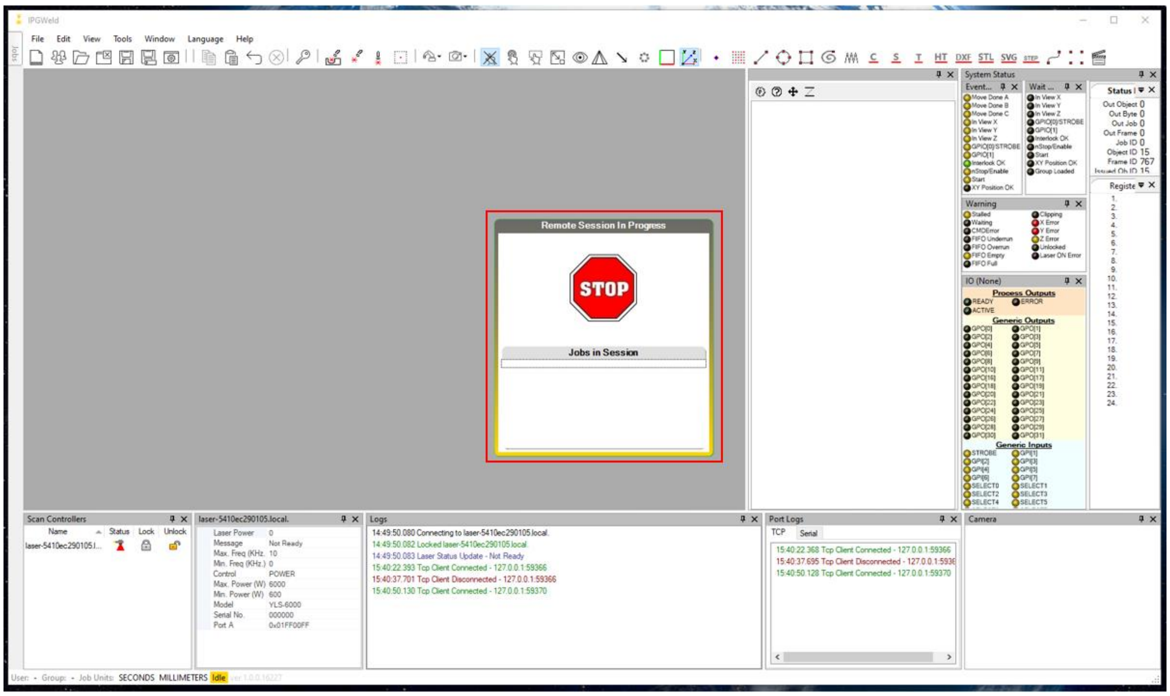
1. Open IPGScan.
2. Click on “View” -> “Options.”
3. Select “Settings” and scroll down to the TCP/IP settings. See Figure 9-1.

*Figure 9-1 TCP/IP Settings*



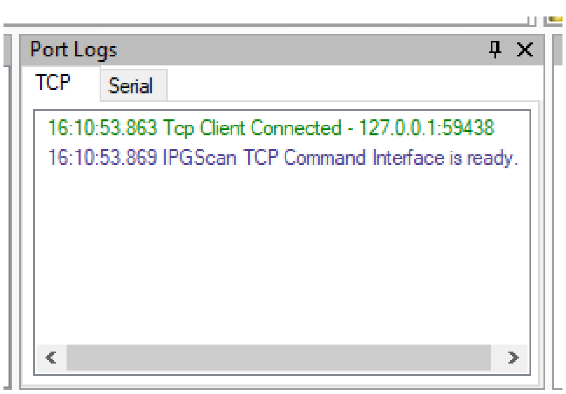

4. Set the following settings as desired.

a. Case Preserving TCPIP Communication – Enables/disables case preserving.
b. Command Port – The port number for which IPGScan will send/receive commands for the Remote API.
c. Encoding – The desired string encoding.
d. Remote Mode Alert – When set “True,” this enables a “Remote Session In Progress” pop-up window that prevents users from manually performing GUI actions within IPGScan while a connection exists with the Remote API.




e. TCP Ready Message – When set “True,” IPGScan will send the message, “IPGScan TCP Command Interface is ready.” when a connection is established with the Remote API. See Figure 9-2.

*Figure 9-2 TCP Ready Message*



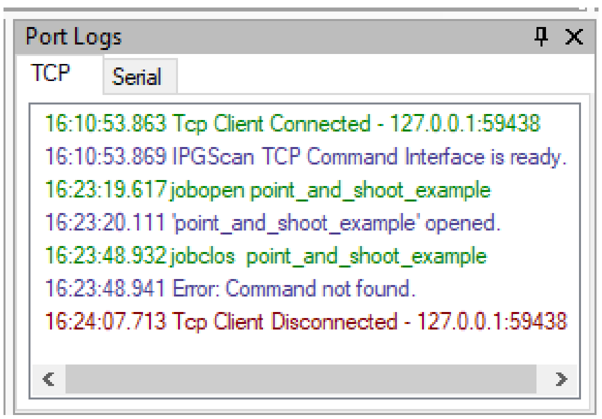

5. Close and reopen IPGScan for the changes to take effect.

**The "Actions Port" setting does not pertain to the Remote API in the TCP/IP settings.**

IMPORTANT **Some versions of IPGScan allowed users to specify an IP Address within the TCP/IP settings. This was to specify a specific IP Address that IPGScan would only send/receive commands from for the Remote API, but was removed because the Command Port number serves a similar purpose.**

With the appropriate settings specified, users can then connect to the IPGScan Remote API for use. Within IPGScan, users can view the Port Logs window to observe any communication via the Remote API. Figure 9-3 provides an example of how users can view the commands and responses for the Remote API in IPGScan.

*Figure 9-3 Port Logs Window*




- Green Text – Incoming commands to IPGScan.
- Blue Text – IPGScan response.
- Brown Text – TCP Client disconnected from the Remote API.

## 9.2 Available Commands

### Help

**Parameters**: none

**Returns**: List of available commands

**Description**: Returns list of available commands.

**Error**: none

**Troubleshooting**: none

**Example**:
S: Help<CR><LF>
R: Send 'Help [commandName]' for command specific help

*List of all available commands*
End Of Command List<CR><LF>

### Help (command)

**Parameters**: *command*

**Returns**: command specific help

**Description**: Returns help string for specified command

**Error**: none

**Troubleshooting**: none

**Example**:
S: help jobopen<CR><LF>
R: JobOpen [jobName]
Opens a job
jobName is Job name without extension<CR><LF>

### JobOpen

**Parameters**: *Filename*

**Returns**: "'*Filename'* opened."

**Description**: Opens a job file in the IPGScan Jobs folder. Filename should not include the ".wjb" extension.

**Error**: "Error: *'Filename'* not found"

**Troubleshooting**: Job doesn't exist or can't be opened. Check that spelling and case is correct. Non-ASCII characters in job names may also cause this error.

**Example**:
S: JobOpen weld-job<CR><LF>
I: Job weld-job is opened
R: 'weld-job' opened.<CR><LF>

JobStart

**Parameters**: *Filename*

**-***guide* (enables the guide [optional])

-*dryrun (enables dryrun* [optional])

**-***savefile* (enables saving binary file [optional])

-*signalmonitoroff (turns off signal/Laser On Monitor* [optional])

-*signalmonitoron (turns on signal/Laser On Monitor* [optional])

**-***group[name]* (runs the group with the name "name" (no brackets)[optional])

**Returns**: "*'Filename' is starting now…*"

**Description**: Starts processing *Filename* job. Filename should not include the ".wjb" extension.

In newer releases of IPGScan, the JobStart command will also open a job if the job is not already open.

**Error**: "Error: ScanController not connected. 'focus_run' cannot be started right now."
"Error: Processing is in progress. '*CurrentJob*' cannot be started right now."
"Error: '*Filename*' is not opened"

**Troubleshooting**:
Check that a scanner is connected by using the ScannerGetStatus command.
If another job is in progress it must be stopped before the specified job can be started.
Check if the job *Filename* is open by using the JobOpenedList command.

**Example**:
S: JobStart weld-job –groupG1<CR><LF>
I: Group "G1" is ran from the job weld-job.
R: 'weld-job' is starting now…

JobStop / JobAbort

**Parameters**: none

**Returns**: "Job Stopped" or "Job Aborted"

**Description**: JobStop and JobAbort replicate pressing the Stop and Abort buttons, respectively.

**Error**: "Error: No Running Job found."

**Troubleshooting**: No job is currently running.

**Example**:
S: JobStop<CR><LF>
I: Currently running job is stopped.
R: Job Stopped<CR><LF>

JobClose

**Parameters**: *Filename*

**Returns**: "'*Filename*' closed."

**Description**: *Filename* is closed.

**Error**: "Error: *Filename* not closed"

**Troubleshooting**: A job cannot be closed if any job is running or if the specified job is not open.

**Example**:
S: JobClose weld-job
I: weld-job is closed.
R: 'weld-job' closed.

### JobList

**Parameters**: none

**Returns**: List of filenames inside IPGScan's Jobs folder. Carriage return and line feed ("\r\n") are appended at the end of every job in the list. The last line will be "End Of Job List\r\n".

**Description**: List all available jobs in IPGScan's Jobs folder.

**Error**: "Error: IPGScan directory not found"

**Troubleshooting**: "C:\IPGP\IPGScan\Jobs" folder cannot be found. Check that this folder exists on the file system.

**Example**:
S: JobList<CR><LF>
R: *Job1*<CR><LF>
…
*JobN*<CR><LF>
End Of Job List<CR><LF>

### JobOpenedList

**Parameters**: none

**Returns**: List of currently opened jobs. Carriage return and line feed ("\r\n") are appended at the end of every job in the list. The last line will be "End Of Job List\r\n".

**Description**: Lists the currently opened jobs.

**Error**: none

**Troubleshooting**: none

**Example**:
S: JobList<CR><LF>
R: *OpenedJob1*<CR><LF>
…
*OpenedJobN*<CR><LF>
End Of Job List<CR><LF>

### JobGetStatus

**Parameters**: none

**Returns**: "Idle" or "Busy"

**Description**: Returns the status of IPGScan. Returns busy if job is running.

**Error**: none

**Troubleshooting**: none

**Example**:
S: JobGetStatus<CR><LF>
R: Busy<CR><LF>

### JobGetStatus2

**Parameters**: none

**Returns**: JobGetStatus2 Group: '*groupName*', Object Name: '*objectName*'

**Description**: Returns the name of the group and object currently being added to the buffer or run by IPGScan (in the case of an Action Control.)

**Error**: none

**Troubleshooting**: none

**Example**:
S: JobGetStatus2<CR><LF>
R: JobGetStatus2 Group: 'G1', Object Name: 'Circle1'<CR><LF>

### JobLastRunSuccessful

**Parameters**: none

**Returns**: "True" or "False"

**Description**: Returns the status of the last run job; will return "False" if the job encountered any errors.

**Error**: none

**Troubleshooting**: none

**Example**:
S: JobLastRunSuccessful<CR><LF>

R: True<CR><LF>

JobExport

**Parameters**: none

**Returns**: Job Process Object parameter data

**Description**: Exports the currently opened jobs Process Object parameter data to xml.

**Error**: none

**Troubleshooting**: none

**Example**:
S: JobExport<CR><LF>
R: All parameter data

ScannerGetStatus

**Parameters**: none

**Returns**: Name of the scanner that IPGScan is currently connected.

**Description**: Used to inform if IPGScan is connected to a specific scanner

**Error**: "Error: Not Connected"

**Troubleshooting**: IPGScan is not connected to a scan controller.

**Example**:
S: ScannerGetStatus<CR><LF>

R: laser-scanner.local. <CR><LF>

GetEncoding

**Parameters**: none

**Returns**: Text Encoding Scheme

**Description**: Returns text encoding scheme set in Options.

**Error**: none

**Troubleshooting**: none

**Example**:
S: GetEncoding<CR><LF>
R: UTF8<CR><LF>

ScannerGetStartBit

**Parameters**: none

**Returns**: "True" or "False"

**Description**: Returns the hardware value of the start signal. "True" if active; "False" if inactive.

**Error**: Error: ScanController not connected.

**Troubleshooting**: Be sure that the desired scan controller is connected.

**Example**:
S: ScannerGetStartBit<CR><LF>
R: True<CR><LF>

ConnectionGetStatus

**Parameters**: none

**Returns**: Name of the computer on which IPGScan is running.

**Description**: Returns the name of the computer on which IPGScan is running. Can be used to check the connection with IPGScan.

**Error**: none

**Troubleshooting**: Check IP Address and Port in IPGScan Options and the connection target of the TCP client.

**Example**:
S: ConnectionGetStatus<CR><LF>
R: LAPTOP-144F4I9E<CR><LF>

### ScannerGetEnableBit

**Parameters**: none

**Returns**: “True” or “False”

**Description**: Returns the hardware value of the enable signal. “True” if active; “False” if inactive.

**Error**: Error: ScanController not connected.

**Troubleshooting**: Be sure that the desired scan controller is connected.

**Example**:
S: ScannerGetEnableBit<CR><LF>
R: True<CR><LF>

### ScannerGetPortA

**Parameters**: none

**Returns**: Port A value in hexadecimal.

**Description**: Port A value in hexadecimal.

**Error**: Error: ScanController not connected.

**Troubleshooting**: Be sure that the desired scan controller is connected.

**Example**:
S: ScannerGetPortA<CR><LF>
R: 0x01FF0EFF<CR><LF>

### ScannerLock

**Parameters**: *scannerName*

**Returns**: ‘*scannerName*’ is locked.

**Description**: Locks a scanner.

**Error**:
Error: ‘*scannerName*’ could not be locked. *scannerName* is already locked.
Error: ‘*scannerName*’ could not be locked. Other Scanner is already locked.
Error: ‘*scannerName*’ could not be locked.

**Troubleshooting**:
Check that IPGScan is not already connected to a different scanner.
Check that the desired scanner is available to be locked.

**Example**:
S: ScannerLock laser-scanner.local.<CR><LF>
I: IPGScan locks laser-scanner.local
R: ‘laser-scanner.local’ is locked.<CR><LF>

### ScannerUnlock

**Parameters**: *scannerName [optional]*

**Returns**: ‘*scannerName’* is unlocked.

**Description**: Unlocks a scanner. If no *scannerName* is provided, the currently locked scanner will be unlocked.

**Error**:
Error: ‘*scannerName*’ could not be unlocked.
Error: ‘*scannerName*’ could not be unlocked. ‘*CurrentScanner*’ is locked currently.
Error: ‘*scannerName*’ is not locked.

**Troubleshooting**: Check that IPGScan is connected to the scan controller trying to be unlocked.

**Example**:
<ScannerLock>
S: ScannerUnlock<CR><LF>
I: IPGScan unlocks the current scanner.
R: ‘laser-scanner.local.’ is unlocked.<CR><LF>

## ScannerInit

**Parameters**: none

**Returns**: Currently locked scanner is initialized.

**Description**: Initializes currently locked scanner.

**Error**:

Error: Failed initializing currently locked scanner.

Error: ‘No Scanner is locked.

**Troubleshooting**: Be sure that a scanner is locked before initializing a scanner.

**Example**:

S: ScannerInit<CR><LF>
I: IPGScan initializes the current scanner.
R: Currently locked scanner is initialized.

## ScannerParkAt

**Parameters**: Desired galvo position in millimeters.

**Returns**: ParkAt done.

**Description**: Parks galvos at specified position. Position is specified by 3 numbers separated by one space. Format is “X Y Z.”

**Error**:

Error: ScannerParkAt failed due to wrong parameters.

Error: ScanController not connected. ParkAt cannot be performed right now.

**Troubleshooting**:

Verify that the position is specified by 3 numbers separated by one space.
Verify that a scan controller is connected.

**Example**:

S: ScannerParkAt 1 2 3.5<CR><LF>
I: Current scanner is parked at position (1, 2, 3.5)
R: ParkAt done.<CR><LF>

## ScannerGetWorkspacePosition

**Parameters**: none

**Returns**: Galvo Position: *X Y Z*

**Description**: Gets the current position of the galvos in millimeters.

**Error**:

Error: ScanController not connected. ScannerGetWorkspacePosition cannot be performed right now.

**Troubleshooting**: Verify that IPGScan is connected to a scan controller.

**Example**:

S: ScannerGetWorkspacePosition<CR><LF>
R: Galvo Position: 0.245 2 0<CR><LF>

ScannerGetList

**Parameters**: none

**Returns**: *scannerName1*
*scannerName2*
…
End Of Scanner List

**Description**: Returns the list of scanners currently visible to IPGScan. Each scanner is separated by a carriage return and a new line, “\r\n”.

**Error**: none

**Troubleshooting**: none

**Example**:
S: ScannerGetList<CR><LF>
R: laser-alpha.local.<CR><LF>
laser-beta.local.<CR><LF>
End Of Scanner List<CR><LF>

ScannerGetStatusList

**Parameters**: none

**Returns**: *scannerName1, available*
*scannerName2, busy*
…
End Of Scanner Status List

**Description**: Returns the list of scanners currently visible to IPGScan and its status. Each status is separated from its scanner by a comma and a space, “, ”. Each scanner is separated by a carriage return and a new line, “\r\n”.

**Error**: none

**Troubleshooting**: none

**Example**:
S: ScannerGetStatusList<CR><LF>
R: laser-alpha.local., available<CR><LF>
laser-beta.local., busy<CR><LF>
End Of Scanner Status List<CR><LF>

ScannerGetConnectionStatus

**Parameters**: *scannerName*

**Returns**: the connection status of the specified scanner

**Description**: Returns connection status of a scanner, either “Available” or “Busy”.

**Error**: Error: *scannerName* not found

**Troubleshooting**: Verify that IPGScan is connected to a scan controller.

**Example**:
S: ScannerGetConnectionStatus laser-scanner.local.<CR><LF>
R: Scanner ‘laser-scanner.local.’ is Available<CR><LF>

ScannerGuideOff

**Parameters**: none

**Returns**: Turned off Guide beam

**Description**: Turns off the guide beam of currently locked scanner if possible.

**Error**:

Error: Processing is in progress. Cannot turn Guide beam off.

Error: ScanController not connected. Cannot turn Guide beam off.

**Troubleshooting**:

Verify that IPGScan is not currently processing a job.

Verify that IPGScan is connected to a scan controller.

**Example**:

S: ScannerGuideOff<CR><LF>
I: Guide beam of the connected scanner is turned off.
R: Turned off Guide beam<CR><LF>

SystemSetVariable

**Parameters**:

variableNumber – register number to set

value – value to set the register to

**Returns**: SystemSetVariable Done.

**Description**: Sets a register to a given value.

**Error**: Error: SetVariable failed. Variable index out of range.

**Troubleshooting**: Make sure that the variable number trying to be set is greater than 0 and less than or equal to the greatest register.

**Example**:

S: SystemSetVariable 5 IPG<CR><LF>
I: Variable5 now contains the value IPG
R: SystemSetVariable Done.<CR><LF>

SystemGetVariable

**Parameters**: *variableNumber*

**Returns**: SystemGetVariable 'Variable *variableNumber*' value is '*value*'

**Description**: Gets the value of a register.

**Error**:

Error: SetVariable failed. Invalid format.

Error: SetVariable failed. Variable index out of range.

**Troubleshooting**: Verify that the variable specified is a valid register number.

**Example**:

S: SystemGetVariable 5<CR><LF>
R: SystemGetVariable 'Variable 5' value is 'IPG'<CR><LF>

ScannerGetMessageStatus

**Parameters**: none

**Returns**: Scanner Message: *scanner/laser message*

**Description**: Gets current scanner/laser message

**Error**: ScanController not connected.

**Troubleshooting**: Verify that IPGScan is connected to a scan controller.

**Example**:

S: ScannerGetMessageStatus<CR><LF>
R: Scanner Message: Power Supply Off<CR><LF>

SystemGetVersion

**Parameters**: none

**Returns**: IPGScan Version: *IPGScan version*

**Description**: Gets the IPGScan version.

**Error**: none

**Troubleshooting**: none

**Example**:
S: SystemGetVersion<CR><LF>
R: IPGScan Version: 1.0.0.9800<CR><LF>

SystemResetAllAlarms

**Parameters**: none

**Returns**: "Reset all alarms complete"

**Description**: Resets all active alarms in IPGScan.

**Error**: none

**Troubleshooting**: none

**Example**:
S: SystemResetAllAlarms<CR><LF>
R: Reset all alarms complete<CR><LF>

DWSResetRunningMax

**Parameters**: none

**Returns**: "DWSRunningMaxReset"

**Description**: Resets Dirty Window Sensor's running Max value.

**Error**: none

**Troubleshooting**: none

**Example**:
S: DWSResetRunningMax<CR><LF>
I: Running Max value is reset in the IPGScan DWS Status Window.
R: DWSRunningMaxReset<CR><LF>

DWSGetRunningMax

**Parameters**: none

**Returns**: string containing current Running Max value in IPGScan DWS Status Window.
Example: "-67.22169"

**Description**: The Running Max Value displayed in the IPGScan DWS Status Window.

**Error**: none

**Troubleshooting**: none

**Example**:
S: DWSGetRunningMax<CR><LF>.
R: *RunningMaxValue*<CR><LF>

DWSGetInstantValue

**Parameters**: none

**Returns**: string containing current Instant value in IPGScan DWS Status Window.
Example: "-30.56"

**Description**: The Instant Value displayed in the IPGScan DWS Status Window.

**Error**: none

**Troubleshooting**: none

**Example**:
S: DWSGetInstantValue<CR><LF>
R: *InstantValue*<CR><LF>

LaserGetStatusMessage

**Parameters**: none

**Returns**: Current Laser Status Message from laser status window.

**Description**: Returns the current status that is reported in the Message field in the laser status window.

**Error**: none

**Troubleshooting**: none

**Example**:
S: LaserGetStatusMessage<CR><LF>
R: Not Ready<CR><LF>

LaserGetStatusCode

**Parameters**: none

**Returns**: Current laser status code.

**Description**: Returns the current laser status code. Users should reference corresponding laser documentation for code details.

**Error**: none

**Troubleshooting**: none

**Example**:
S: LaserGetStatusCode<CR><LF>
R: LaserGetStatusCode<CR><LF>

# 10 External Devices

## 10.1 Sentech Ethernet Camera

Sentech Ethernet cameras can be utilized to provide a live video within IPGScan. This can be useful for aligning, previewing, and inspecting the work surface. This section seeks to provide users with an understanding of how to connect to an Ethernet camera within IPGScan, setup crosshairs for target alignment, as well as provide information on other existing camera functionality.

For details concerning the hardware mounting and setup of a camera and camera arm, please refer to the Scanner Series User Guide. Please also refer to any accompanying camera documentation.

### 10.1.1 Sentech Ethernet Camera Connection

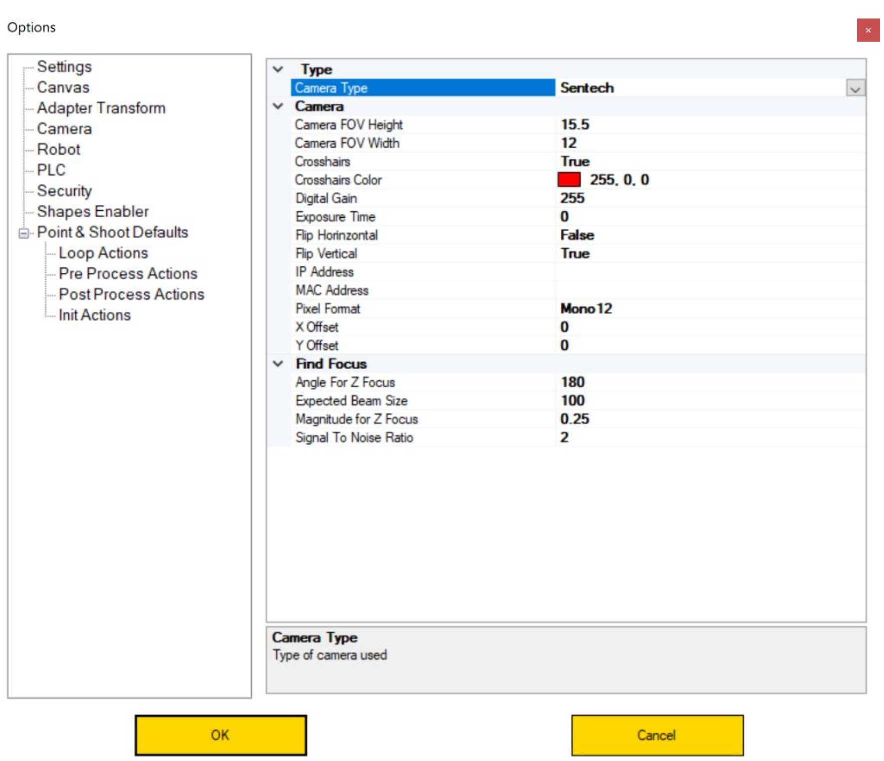
The following steps outline how users can connect to the Ethernet camera in IPGScan.

1. Connect the camera to a computer using an Ethernet cable.

| | |
|---|---|
| **IMPORTANT** | **The camera should be directly connected to the computer. While it is ok to use a USB to Ethernet adapter for this connection, the camera should not be put on a network. High network traffic can interrupt the connection with the camera.** |

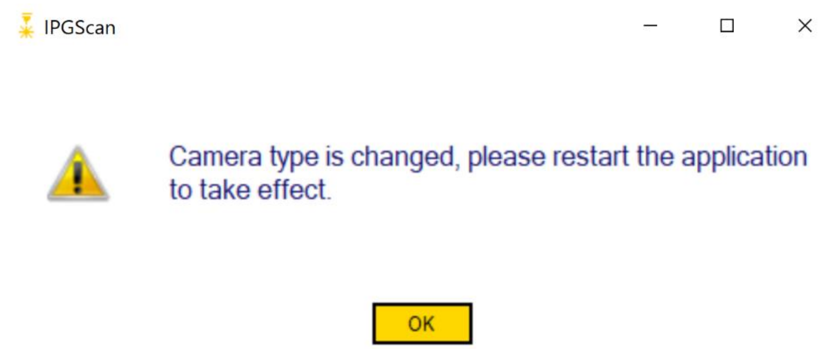
2. Power the camera on using the provided power supply.
3. Launch IPGScan.
4. Open the “Options” menu and navigate to “Camera.”
5. For “Camera Type,” select “Sentech.” See Figure 10-1.

*Figure 10-1 Selecting Camera Type*



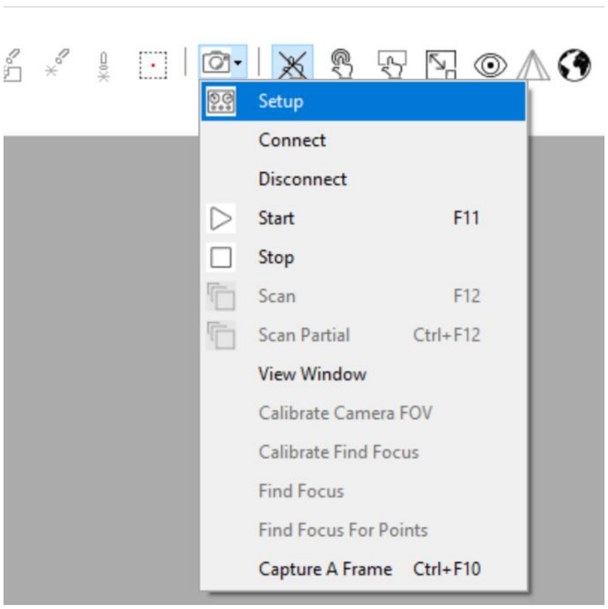
6. Click “OK” to close the IPGScan Options window. Users will then be prompted to restart IPGScan for the changes to take effect. Click “OK.” See Figure 10-2.

*Figure 10-2 Prompt to Restart IPGScan*



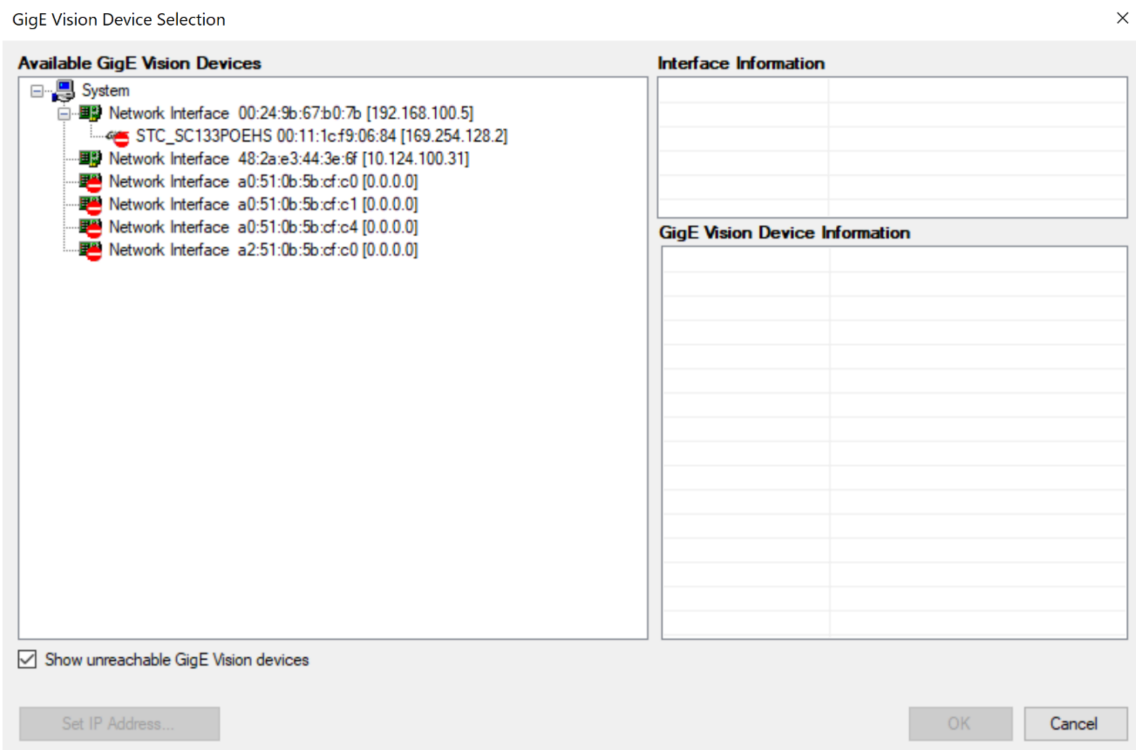

7. Close IPGScan.
8. Open IPGScan.
9. Users should now have a camera icon in the IPGScan Tool Bar. Click the camera icon and click “Setup.” See Figure 10-3.

*Figure 10-3 Camera Tool Bar Menu*



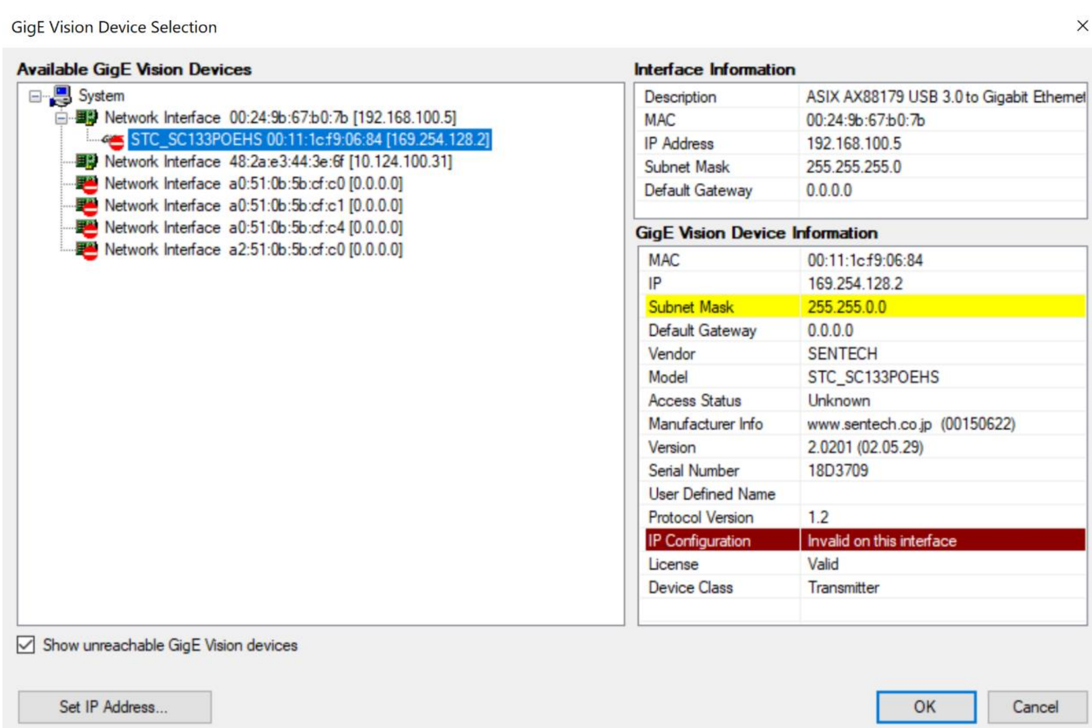

10. In the setup window, users will find various network interfaces and any cameras that might be available on those network interfaces. If a camera is connected to the computer but is not appearing under one of the specified networks, click the “Show unreachable GigE Vision devices” check box to find the camera. See Figure 10-4.

*Figure 10-4 Finding Cameras in the Setup Menu*




11. Select the desired camera from the list. See Figure 10-5.

*Figure 10-5 Selecting a Camera*

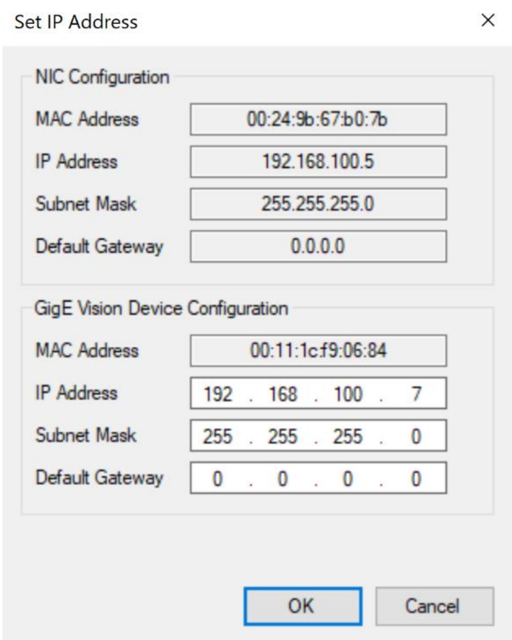



12. If an IP address or Subnet Mask conflict exists, users will need to set a new IP address on the

camera. Click the “Set IP Address” button to open the IP address settings window.

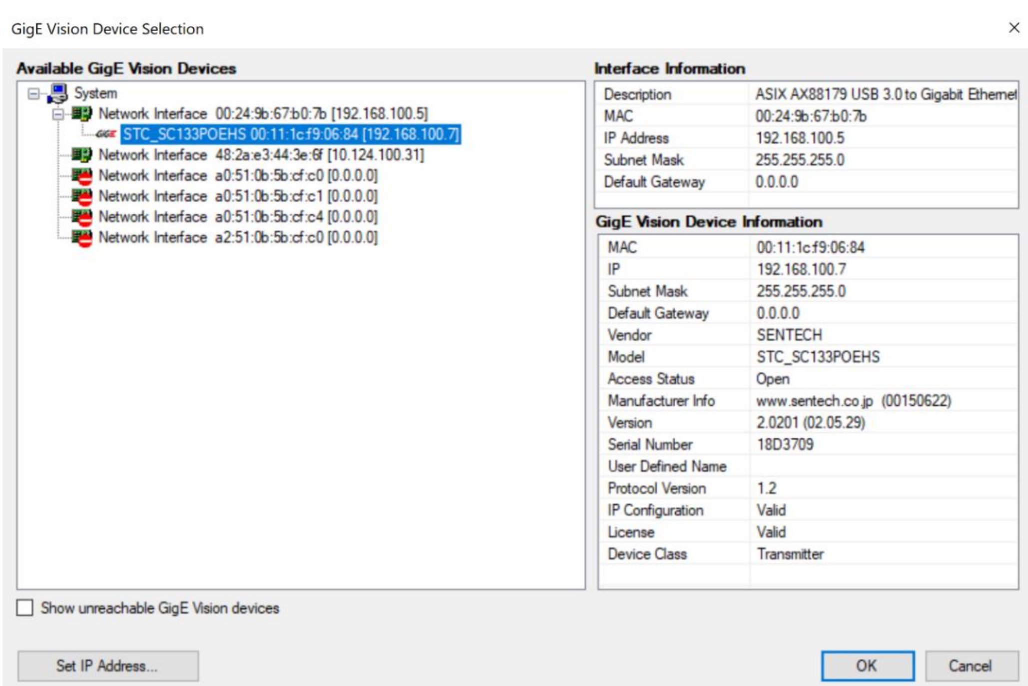
13. In the “Set IP Address” window, specify a IP address and subnet mask for the camera. See Figure 10-6.

*Figure 10-6 Set IP Address Window*




14. Click “OK” once the changes have been made.
15. Users should now be able to select the camera and no adapter setting conflicts should exist. Click “OK” to close the Camera Setup window. See Figure 10-7.

*Figure 10-7 No Network Adapter Conflicts Exist*



16. In the IPGScan Tool Bar, click on the camera icon and click “Start” in order to start streaming the camera image (see Figure 10-8). The camera image will appear in the IPGScan Camera window (see Figure 10-9).
    a. Please note, clicking start causes IPGScan to automatically connect to the camera that was selected in the Setup menu.
    b. If users have closed the Camera window in IPGScan prior to setting up the camera, click the “View Window” or reset the IPGScan layout in the View menu.

*Figure 10-8 Starting Camera Streaming in IPGScan*

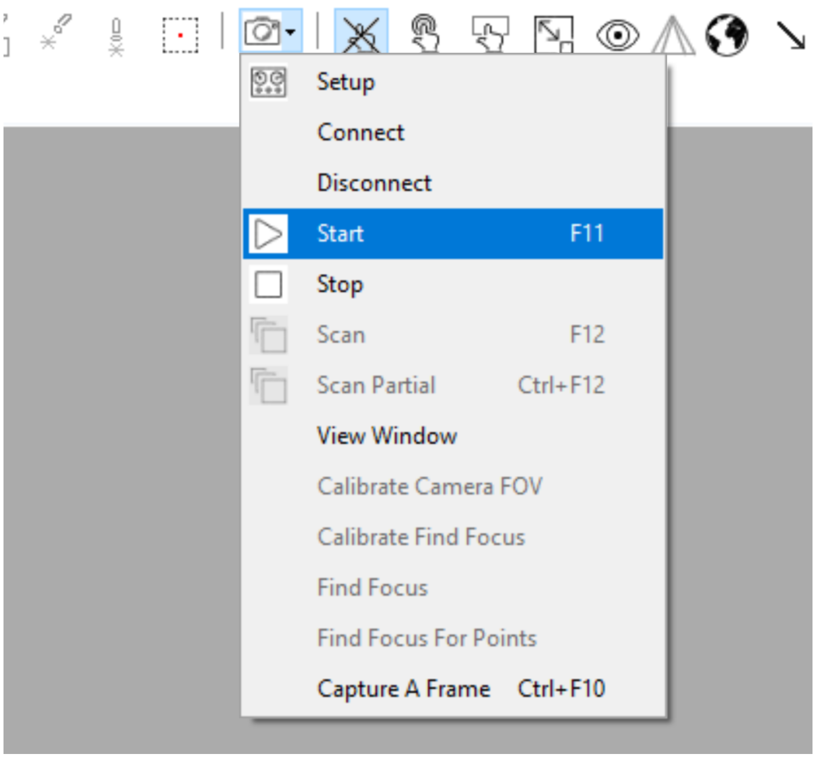
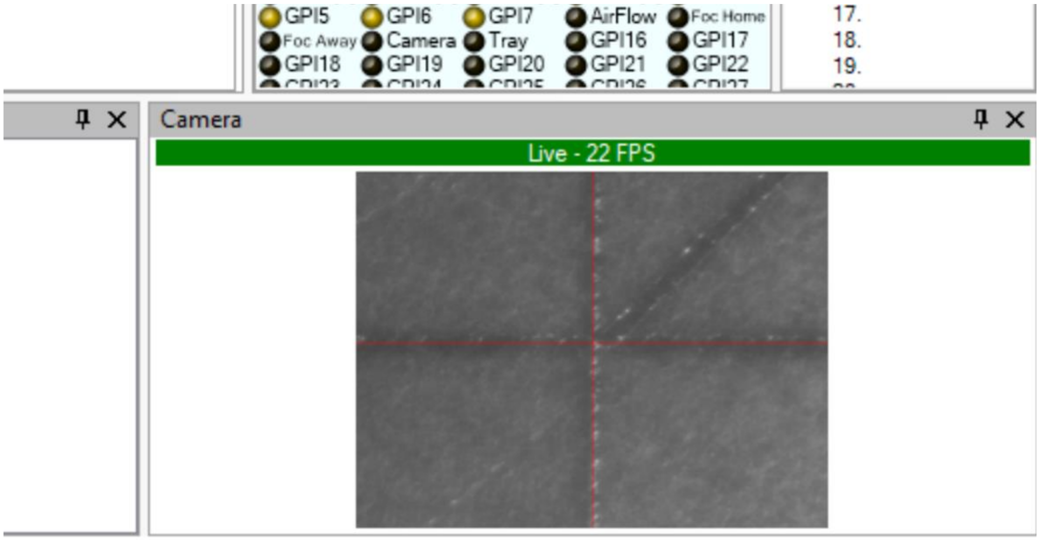


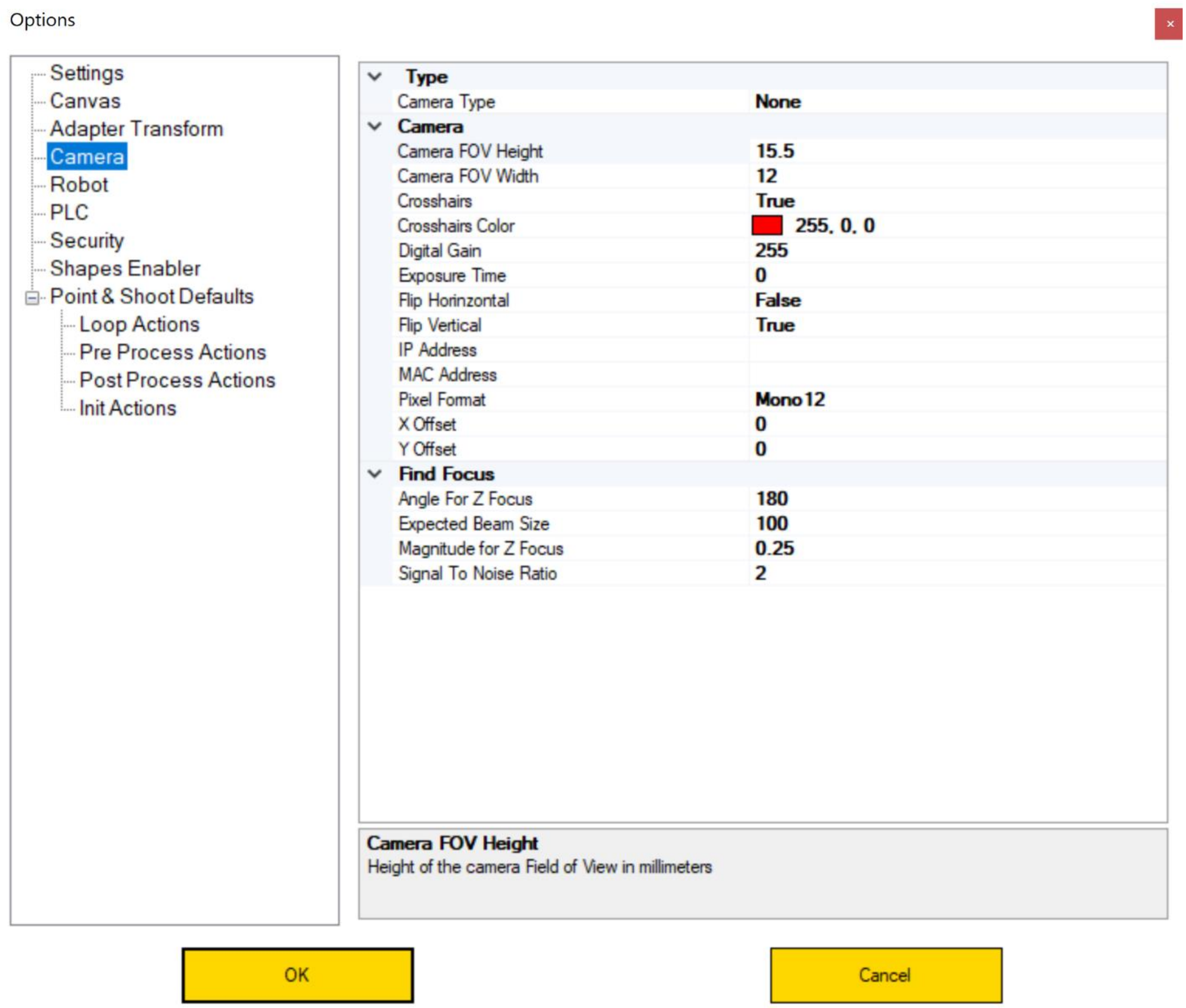
*Figure 10-9 Camera View in IPGScan*



### 10.1.2 Ethernet Camera Settings

The following details outline some of the available camera settings within IPGScan for the Ethernet cameras. Figure 10-10 displays the available camera settings in the IPGScan Options.

*Figure 10-10 IPGScan Ethernet Camera Settings*

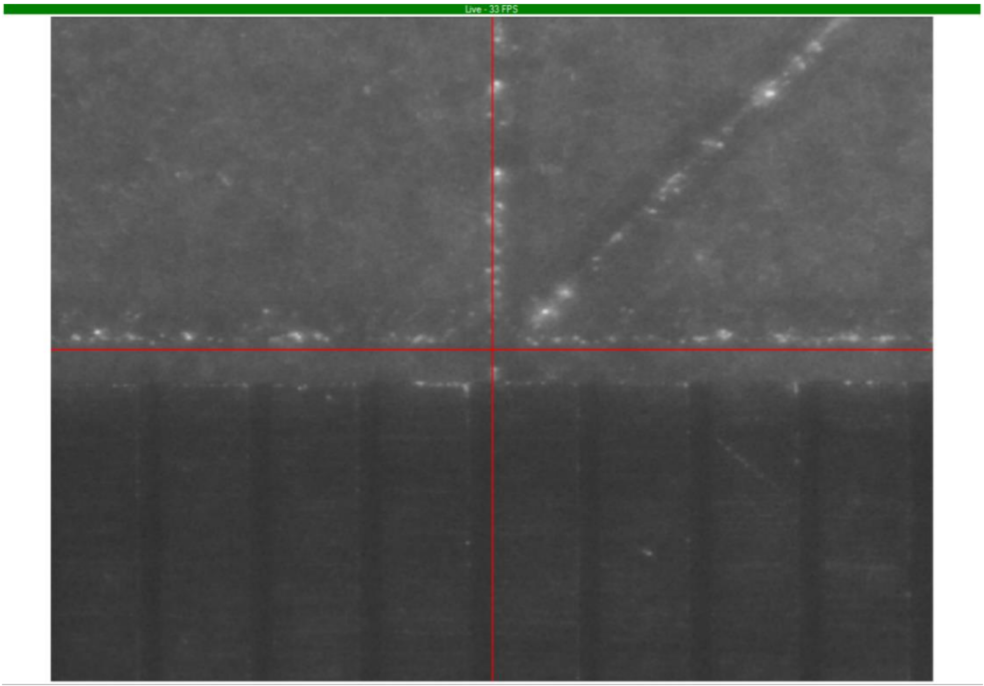



The following settings are available for user adjustment:

- Camera Type
    - This allows users to select the camera they plan on using. The Ethernet cameras provided by IPG are Sentech cameras.
- Camera FOV Height and Camera FOV Width
    - Inputs that specify the measure of the camera viewing area. These measures are required for the “Scan” and “Scan Partial” features which are found in the Camera Icon menu.

Users should get these inputs by placing a ruler under the cameras FOV and getting a measure. See Figure 10-11.

*Figure 10-11 Measuring the Camera FOV*




- Crosshairs and Crosshairs Color
    - These settings allow users to enable or disable the crosshairs in the camera view as well as change the color of the camera crosshairs displayed.
- X Offset and Y Offset
    - Users can change the crosshair position in the camera FOV by adjusting these values.
- Digital Gain and Exposure Time
    - Users can adjust the camera image using these values.
- Flip Horizontal and Flip Vertical
    - Users can flip the camera image display using these settings.
- Pixel Format
    - Allows users to specify the pixel format (number of bits of data per pixel).
- IP Address and MAC Address
    - When connected to a camera, the IP Address and MAC Address of the camera will be displayed here.
- Find Focus
    - This feature is currently unavailable with 2D High Power Scanners.

# 11 Point and Shoot Processing

## 11.1 Overview

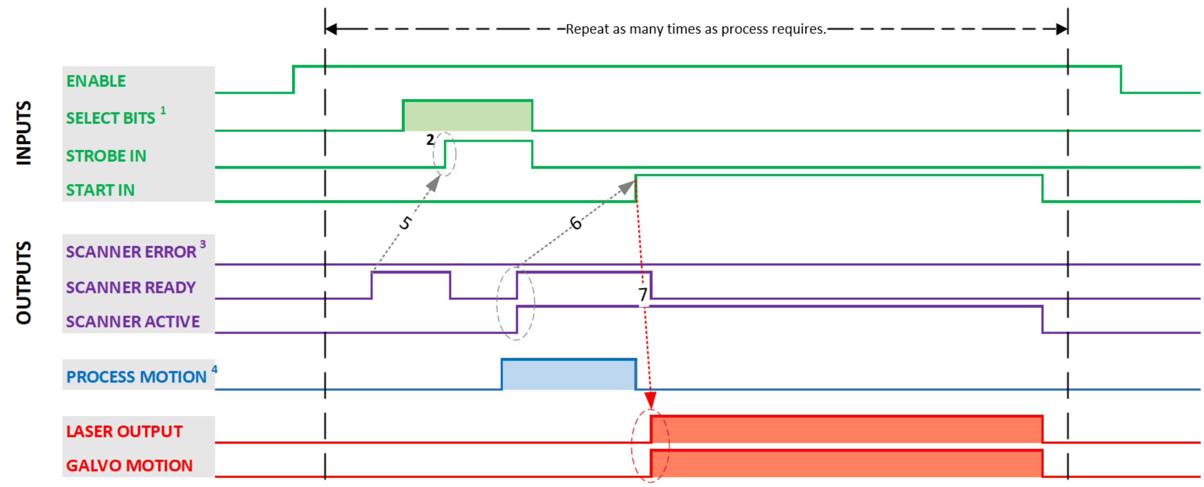
Point and Shoot processing is a scanner application where the scan controller and an external device (e.g., PLC and/or a robot) work together to integrate the scanner into a larger automation system. The external device can choose what processing is done by the scanner based upon other components in the system (e.g., turn table, fixturing, or robot motion). The processing configuration and parameters are configured in IPGScan and the external device sequences when each object should be output.

A couple of use cases are:

- Using a robot to weld a part which is larger than the scan head's field of view
- Using a PLC to mark a barcode on a part that comes off a production line
- Using a PLC to process a component based on which part is presented to the scanner

## 11.2 Signal Sequencing

The external device coordinates with the scanner system through a series of signals. Figure 11-1 is a timing diagram which shows the sequencing of these signals. The signals work in the following order:




1 - This represents all 9 select bits. The select bits represent a binary value that is converted into a decimal value for job selection.
2 – Select bits must be set to the desired state prior to setting strobe active.
3 - ERROR only goes active if processing is interrupted (such as ENABLE going inactive during processing) or the scanners connection is lost during processing. ENABLE must be active when processing is started.
4 - Process motion entails any process procedures that may be required prior to processing with the scanner. Examples include presenting the part to the scanner or moving a robot/positioner that the scanner is mounted to into the desired processing position. This does not represent signal input/output.
5 – Control must wait Ready output HIGH to assert Strobe
6 – Control must wait Ready **and** Active HIGH to assert Start (besides Process Motion)
7 – Scanner Motion and Laser Output

*Figure 11-1 - Point and Shoot Timing Diagram*

1. The external device should select the group ID (see section "Groups and Group IDs") of the processing objects the scanner should next execute. This selection is done with the SELECT[8:0] bits. After all of the bits have been set, the external device should set the STROBE bit to active.
    a. If the desired output is always the same processing group (e.g. a company logo and a serial number or a series of the same welds), the group selection can be skipped. The sequence can function with only the STROBE or START bits.

    b. If a different method of group selection than hardwire signals is desired (e.g. Serial or Ethernet), steps 1 and 2 would be replaced with the desired method.
2. The scanner first waits for the STROBE signal from the external device. While the scanner is waiting: the READY signal will be active, the ACTIVE signal will be inactive, the ERROR signal will be inactive.
3. After receiving the STROBE signal, the scanner system will deactivate the READY signal. When IPGScan has processed the data into the scan controller, the READY and ACTIVE signals will be active.
4. When the external device would like to tell the scan controller to begin outputting the processing objects previously loaded, it should set the START signal to active. The STROBE signal can be set to inactive.
5. While the scan controller is outputting the processing objects, the READY signal will be inactive and the ACTIVE signal will remain active.
6. When all of the chosen processing objects have been output by the scanner, the ACTIVE signal will be set to inactive and the process will return to step 1.

**IMPORTANT** **The above sequencing is dependent upon the IPGScan job structure. Please refer to the following sections for additional detail.**

## 11.3 IPGScan Point and Shoot Job Setup

A Point and Shoot job in IPGScan uses the same processing objects as a Default job but uses additional Action Controls and job structure to achieve the Point and Shoot capabilities previously described. A Point and Shoot job can be created from a Default type IPGScan job and a Point and Shoot type IPGScan job. Both are described below using a hardwiring implementation. A Point and Shoot job can be configured to use an interface other than a hardwire interface as well.

For a Point and Shoot job, the external device selects what the scanner should execute by selecting groups based upon the group ID. A group must start with an Action Control to block execution until the START signal. Following this Action Control, a group can contain any number of processing objects and Action Controls. The whole job is managed by the "Loop Group." The Loop Group gets the Group ID of the next group to run from the external device. Then the Loop Group moves execution to the selected group.

**IMPORTANT** **In a Default job, an additional GoToGroup Action Control is required after each processing group to return execution to the Loop Group.**

### 11.3.1 Point and Shoot Job Type

The following sections outline the setup and use of an IPGScan "Point & Shoot" job type.

#### *11.3.1.1 Default Point & Shoot Job Settings*

The user can configure the default "Point & Shoot" job type settings by going to View → Options, in IPGScan. Extending the "Point & Shoot Defaults" drop down, and clicking on "Loop Actions," "Pre Process Actions," "Post Process Actions," or "Init Actions" allows the user to configure the default settings for each collection. Figure 11-2 illustrates the "Pre Process Actions" default settings.

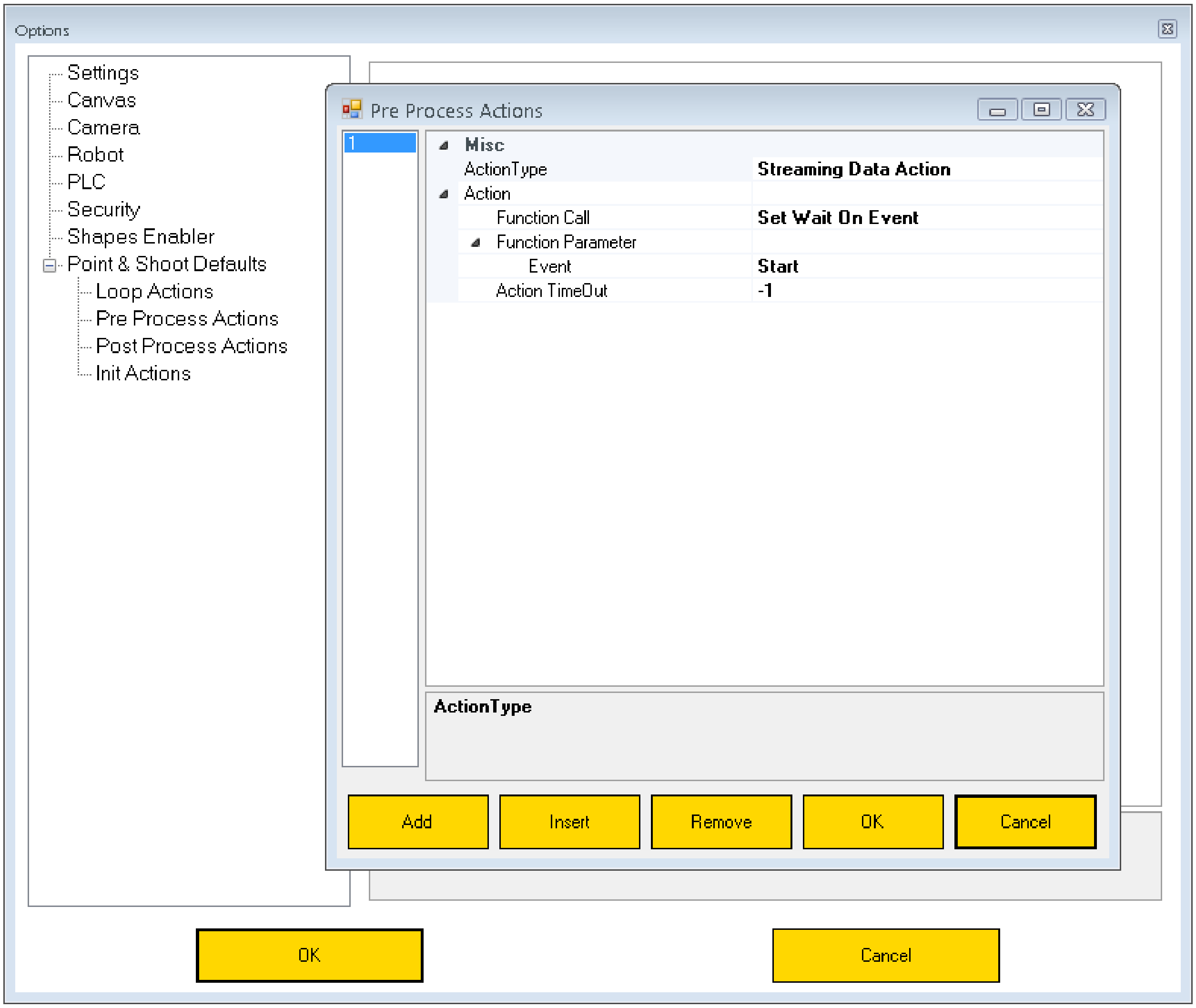



*Figure 11-2 - Pre Process Actions*

Using the “Add” or “Insert” buttons, the user can add additional program steps to the Loop, Pre Process, Post Process, and Init collection settings. Clicking “Add” will insert a step below all the other steps while clicking “Insert” adds a step above the step that was selected. Clicking “Remove” will delete the selected step.

By default, IPGScan is setup so that the “Loop Actions,” “Pre Process Actions,” “Post Process Actions,” and “Init Actions” are configured with Action Controls that are typically used in a default type Point and Shoot job. The purpose behind this is to make it so users only need to create the desired number of groups, confirm Group IDs, and insert process objects in order to create a job.

#### *11.3.1.2 Job Setting*

Any job can be set as a “Point & Shoot” type job by selecting the job name and then select “Point & Shoot” from the “Type” drop down menu. See Figure 11-3.

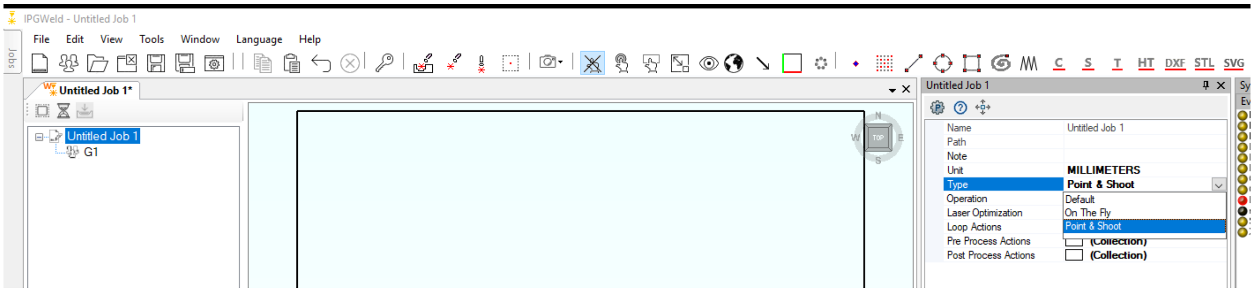



*Figure 11-3 - Selecting "Point & Shoot" Job Type*

Next, the user simply needs to create the desired number of process groups and add process objects to each group. Figure 11-4 outlines an example IPGScan Point and Shoot job structure that consists of two Process Groups and an Exit Group to stop processing. In a Point and Shoot type job, the Action Controls are hidden within collection boxes (accessed by clicking on the job name) whereas in a Default type job, users must add the Action Controls directly within the Job Tree.


*Figure 11-4 - Point & Shoot Job Type Example Program Structure*

### *11.3.1.3 Understanding Process Sequencing*

Although the user no longer is required to create the Action Controls in each process group that are normally required for a Default type point and shoot job, the same Action Controls are still being utilized. These Action Controls are simply created and stored within the "Loop Actions," "Pre Process Actions," "Post Process Actions," and "Init Actions" collection boxes. When the job is started, processing immediately jumps into the "Init Actions" collection. This collection is to initialize any registers required for the IPGScan job. The "Init Actions" are only run once at the beginning of the job. Processing then goes into the "Loop Actions" collection box. Once all actions are performed within the "Loop Actions" collection, processing then proceeds into the "Pre Process Actions" collection box. After the "Pre Process Actions" are performed, processing then proceeds to the Process Group which the user selected via the Load Register Action Control in the "Loop Actions." Finally, once all objects and actions are performed in the Process Group, anything setup in the "Post Process Actions" will execute and processing will then return to the "Loop Actions" collection.

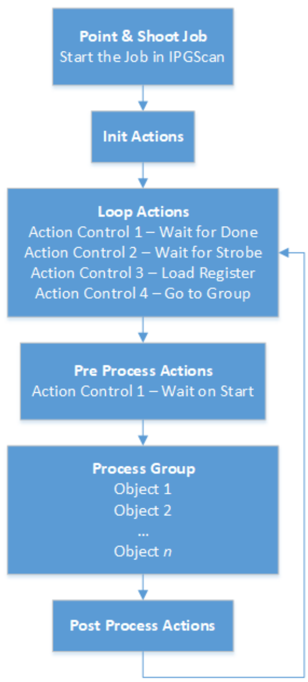



*Figure 11-5 - Point and Shoot Job Example Sequence*

**Note:** Processing takes place through the job tree sequentially.

**Note:** The user has the ability to add, modify, or remove any of the Action Controls outlined in Figure 11-5. Figure 11-5 outlines the "Loop Actions," "Pre Process Actions," "Post Process Actions," and "Init Actions" as they should be upon first installing IPGScan. All objects and actions in the "Process Group" are determined by the user.

#### *11.3.1.4 Modifying "Loop Actions," "Pre Process Actions," "Post Process Actions," and "Init Actions"*

The user has the ability to modify the "Loop Actions," "Pre Process Actions," "Post Process Actions," and "Init Actions" specific to each individual IPGScan "Point & Shoot" type job. By clicking on the job title in the Job Tree, the user can access the collection box for each of the process stages in the Parameter Tree (see Figure 11-6). Within each of the collection boxes, users can add, modify, and remove Action Controls and objects. Please note that changes made to the collection boxes within the job are specific to only that job.


*Figure 11-6 - Modifying Process Stages Specific to an IPGScan Job*

### 11.3.2 Default Job Type

The following list is an example Default Point and Shoot style job. This is copied from the job "point_and_shoot_example.wjb" which can be found in the Jobs folder of an IPGScan installation.

- Loop Group (Group ID = 1)
    - Wait - WaitForDone
        - Description
            - The purpose of this Action Control is to block the execution of the job until all of the previously chosen processing objects have been output.
        - Properties
            - Action Type: Wait
            - WaitForMethodTypes: Wait For Done
            - Wait For Method: <blank>
            - Action TimeOut: -1
    - Wait – PortABit_True
        - Description
            - The purpose of this Action Control is to block the execution of the job until the STROBE signal from the external device is active as part of the handshaking sequence (see Section 11.2).
        - Properties
            - Action Type: Wait
            - WaitForMethodTypes: Port A Bit Action
            - Wait For Method
                - Bit: 0
                - Wait For: True
            - Action TimeOut: -1
    - LoadRegister – PortA
        - Description
            - The purpose of this Action Control is to read the selected group to execute from the external device. The selected group is stored in the "Destination Register," in this case Variable 1.
        - Properties
            - Action Type: Load Register
            - Load Register Using Type: Port A
            - Load Register Method
                - Shift: 0x010
                - Width: 9
                - Destination Register: Variable 1
    - GoToGroup – Register
        - Description
            - The purpose of this Action Control is to direct the execution of the job to the chosen group based upon the Group ID.
        - Properties
            - Action Type: Go To Group
            - GoToGroupMethodType: Register

            - Go To Group Method
                - Go To Group At: Variable 1
                    - **Note:** the variable in the Go To Group Action Control must match he Destination Register in the previous Load Register Action Control.

- Process1 (Group ID = 2)
    - StreamingDataAction – WaitForStart
        - Description
            - The purpose of this Action Control is to block the output of the processing objects from the Scan Head until the external device is ready and sets the START signal to be active.
        - Properties
            - Action Type: Streaming Data Action
            - Function Call: Set Wait On Event
            - Function Parameter
                - Event: Start
            - Action TimeOut: -1
    - Processing Objects
    - GoToGroup – Constant
        - Description
            - The purpose of this Action Control is to direct the execution of the job back to the Loop Group. The Go To parameter should match the Group ID of the Loop Group.
        - Properties
            - Action Type: Go To Group
            - GoToGroupMethodType: Constant
            - Go To Group Method
                - Go To: 1
- Process2 (GroupID = 3)
    - See Process1
- Exit Group (GroupID = 4)
    - Exit Action
        - Description
            - This Action will stop the running IPGScan job.

## 11.4 External Device

Any external device can be used with the Scan Controller. An external interface board must be used (see External Interface Manual). The most common devices are a PLC or an Industrial Robot.

### 11.4.1 Industrial Robot

When using an industrial robot as the external device with a scanner Point and Shoot job, the same methods of communication are available as with a PLC: Digital I/O, EtherNet/IP, Profinet, and EtherCAT.

To assist in the programming of the robot, helper functions have been written for several robot brands: KUKA, ABB, Yaskawa Motoman, and FANUC. These helper functions are listed in Table 11-1. An example program called "IPG_EXAMPLE_WELD" is also available.

*Table 11-1 - Industrial Robot Helper Functions*

| Helper Function | Description |
|---|---|
| IPG_EXECUTE_WELD | This function executes the pre-selected IPGScan group. If this function waits for longer than a specified timeout for the scanner to be ready to execute, the program will stop execution of the robot program. See IPG_PREP_WELD for more information regarding setting and disabling the timeout. |
| IPG_LASER_DIS | This function configures any I/O for disabling the laser or system. It is called IPG_SHUTDOWN_WELD. This function is empty by default. |
| IPG_LASER_EN | This function configures any I/O for enabling the laser or system. It is called IPG_PREP_WELD and SETUP_WELD. This function is empty by default. |
| IPG_PREP_WELD/SETUP_WELD | This function sets up all I/O for use in these helper functions. This function only needs to be called once at the beginning of the main program.<br><br>This function expects a timeout in milliseconds as a parameter. In IPG_EXECUTE_WELD, if the robot waits longer than a specified timeout for the scan controller to be ready, the main program will stop execution. An argument of -1 will disable the timeout.<br><br>IPG_PREP_WELD and SETUP_WELD are the same function. New versions are called IPG_PREP_WELD. |
| IPG_SELECT_WELD | This function selects the desired processing group from IPGScan. This function needs to be called before each call of IPG_EXECUTE_WELD in the default setup of a Point and Shoot job in IPGScan.<br><br>In order to save time, this function should be called before moving the robot and before calling IPG_EXECUTE_WELD. The time that the scanner will spend processing the new data will overlap the time that the robot spends moving. |

| | |
|---|---|
| | This function expects the desired processing group as a parameter. This should be a number between 0 and 511, inclusive. |
| IPG_SELECT_WELD_WAIT | This function is the same as IPG_SELECT_WELD, but incorporates the Ready signal indicating that the Scanner system is ready for the Strobe signal. |
| IPG_SHUTDOWN_WELD | This function turns off all I/O used in these helper functions. This function only needs to be called once at the end of the main robot program.<br><br>IPG_SHUTDOWN_WELD and CLOSE_WELD are the same job. New versions are called IPG_SHUTDOWN_WELD. |

#### *11.4.1.1 Special Cases*

##### 11.4.1.1.1 KUKA

KUKA requires two additional files: IPG_EXAMPLE_WELD.dat and IPG_SCAN_VARIABLES.dat. IPG_EXAMPLE_WELD.dat contains position information used by move commands in IPG_EXAMPLE_WELD.src. IPG_SCAN_VARIABLES.dat contains variable definitions for variables used in multiple helper subprograms.

##### 11.4.1.1.2 ABB

The functions for ABB are provided as a single module file with multiple procedures.

##### 11.4.1.1.3 Yaskawa Motoman

The timeout argument in IPG_PREP_WELD is in clock ticks, not milliseconds. This value is stored in a global integer variable. The default variable is I000.

##### 11.4.1.1.4 FANUC

An additional numeric register is required for the job IPG_SELECT_WELD. By default, register 2 is used. Two user alarms are used as well. Their severity is setup in IPG_SETUP_WELD. The user alarm messages must be setup by the user in the FANUC menu. By default, alarm 1 is used in case of a timeout in IPG_EXECUTE_WELD and alarm 2 is used in case of an invalid group ID in IPG_SELECT_WELD.

#### *11.4.1.2 Pseudocode Example*

The following is a pseudocode example of a Point and Shoot program on an industrial robot.

```
// IPG_PREP_WELD weld is called once at the beginning.
// The argument 1000 will create a timeout of 1 second.

IPG_PREP_WELD(1000)

// IPG_SELECT_WELD is used to select IPGScan Group ID 2

IPG_SELECT_WELD(2)
```

```
// The robot is moved after calling IPG_SELECT_WELD so that IPGScan
// can process the data in Group ID 2 while the robot is moving.
Move_J(P1)
// IPG_EXECUTE_WELD will wait until the scanner system is ready to
// execute.
IPG_EXECUTE_WELD()
// The sequence of IPG_SELECT_WELD, Movement, IPG_EXECUTE_WELD is all
// that is required after calling IPG_PREP_WELD at the beginning of
// the program.
IPG_SELECT_WELD(3)
Move_J(P2)
IPG_EXECUTE_WELD()
IPG_SELECT_WELD(4)
// Multiple moves can be used.
Move_J(P3)
Move_L(P4)
IPG_EXECUTE_WELD()
IPG_SELECT_WELD(5)
// Commands other than moves can be used.
Move_J(P5)
Wait(di[44] == True)
IPG_EXECUTE_WELD()
// IPG_SHUTDOWN_WELD finishes the program and only has to be called at
// the end
IPG_SHUTDOWN_WELD
```

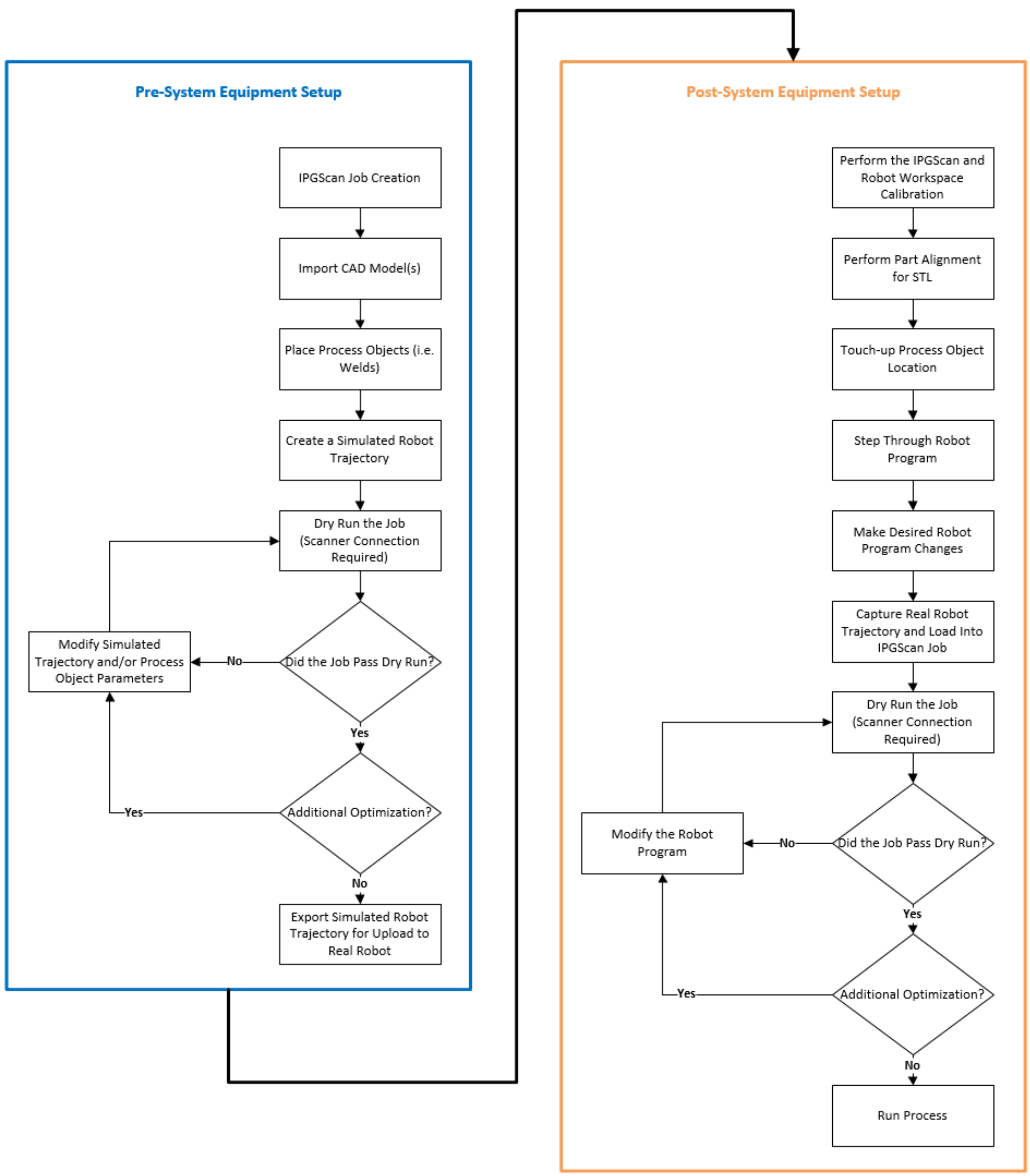
# 12 Robotic On-The-Fly Processing

## 12.1 Overview

Robotic On-The-Fly (OTF) processing enables users to create a process where the scanner is in motion during processing. The purpose of such a process is to help reduce process cycle time and increase throughput. Applications where there are many Process Objects (i.e. welds) which are spaced out over a large area often greatly benefit from the increased throughput that OTF processing can provide.

Prior to starting the development of an OTF process, it is important to understand the overall process development flow. Figure 12-1 outlines the general steps for creating a robotic OTF process when utilizing an IPG scanner.

*Figure 12-1 Robotic OTF Process Overview Diagram*



The following sections detail the features and actions that are both required and optional for the creation of a robotic OTF process with an IPG scanner.

## 12.2 Robot Requirements and Setup

The following sections detail robot requirements and the method to configure IPGScan to communication with a given industrial robot brand.

**IMPORTANT** **Point and Shoot style processing does not require the same communication setup that Robotic OTF processing requires in order to capture robot trajectory data.**

### 12.2.1 Requirements

Each robot manufacturer requires additional options for robot to PC communications and coordination. The additional options for each robot manufacturer are shown in Table 12-1.

*Table 12-1 Robot Required Options*

| **Manufacturer** | **Required Options** |
|---|---|
| KUKA | KUKA Robot Sensor Interface: KUKA.RSI |
| FANUC | SIT PC Interface (PCIF): RTL-PCIN |
| | Robot Server: PC RTL-RSR |
| FANUC KAREL | SIT PC Interface (PCIF): RTL-PCIN |
| | KAREL: RTL-R632 |
| | User Socket Messaging: RTL-R648 |
| Yaskawa Motoman | MotoPlus Robot Controller Option |
| ABB | Externally Guided Motion (EGM): 689-1 |
| | UDPUC Driver |
| | PC Interface: 616-1 |

**IMPORTANT** **Robotic OTF Processing requires a consistent trigger for process repeatability. For this reason, Robot digital discrete IO modules should be wired directly to the START bit on an appropriate External Interface via hardwire. The START bit should not be passed through a PLC via a Fieldbus interface prior to being wired to an appropriate External Interface.**

**IMPORTANT** **With FANUC robots, either the Robot Server interface can be used or the KAREL interface can be used. The KAREL interface results in better results and better performance. The Robot Server interface should only be used if a higher number of coordination flags is required (beyond 5 flags).**

The robot controllers which have been tested with Robotic On-The-Fly are listed in Table 12-2.

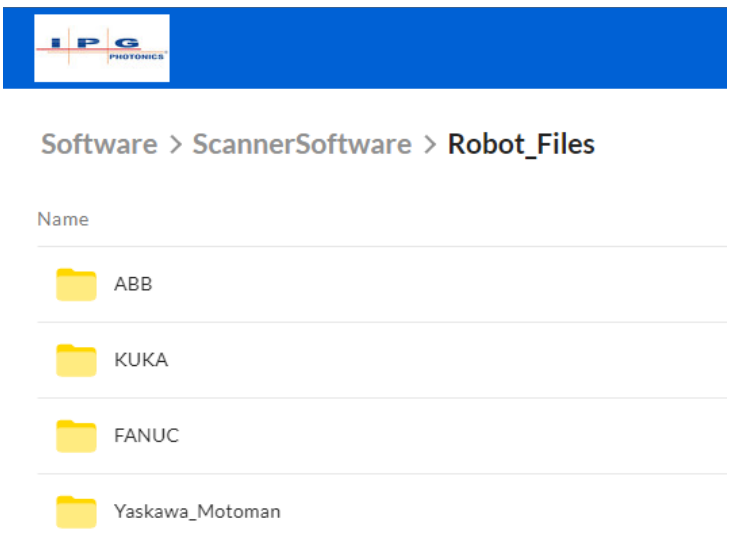
*Table 12-2 Robotic On-The-Fly Tested Controllers*

| **Manufacturer** | **Tested Controllers** |
|---|---|
| KUKA | KRC4 |
| FANUC | R-30iB+ |
| | R-30iB |
| Yaskawa Motoman | DX200 |
| ABB | IRC5, RW 6.08 |

### 12.2.2 Robot Configuration Files

Configuration files for all robots can be found on the IPG software website (software.ipgphotonics.com) under the ScannerSoftware → Robot_Files location. See Figure 12-2. Users will require the appropriate configuration files for the robot being interfaced to.

*Figure 12-2 Robot Configuration Files on IPG Software Website*



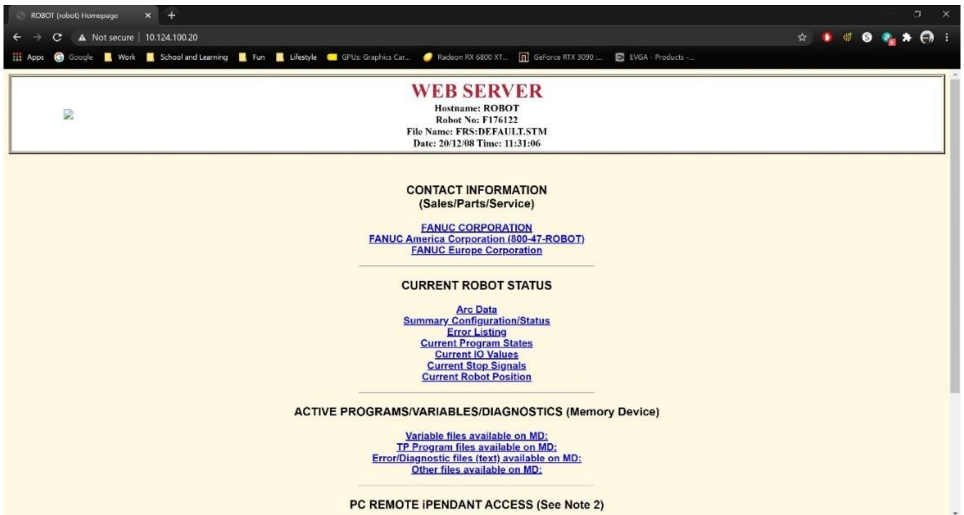
## 12.2.3 FANUC Setup

The following sections detail the configuration of a FANUC robot for use with the Robot Server or KAREL methods of Robotic OTF processing.

### *12.2.3.1 Robot Server*

1. Configure a TCP/IP connection between the robot and the computer.
2. To check, enter the Robot's IP Address into a web browser and the robot's home page should load. See Figure 12-3 as an example.

*Figure 12-3 Robot Web Server Page*

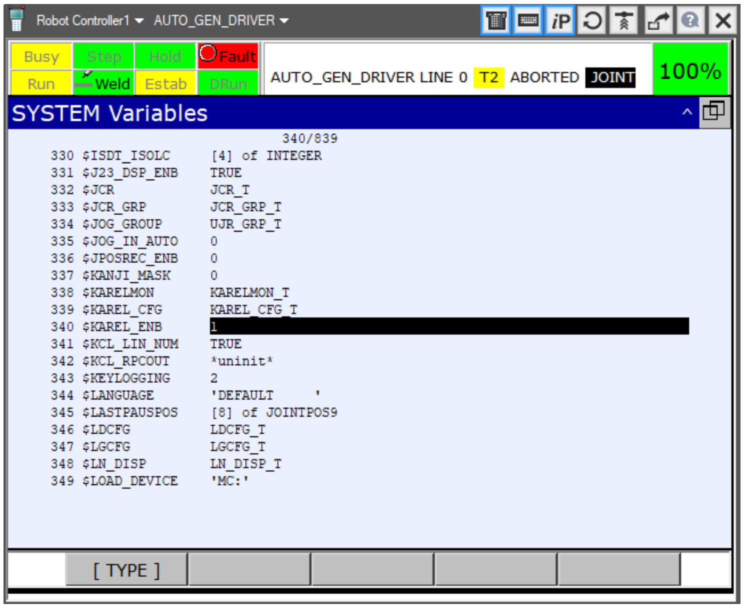



Robot Server is now setup and the user is ready to configure robot settings in IPGScan. See section, "Connecting to a Robot," for additional details.

### *12.2.3.2 KAREL*

1. Configure a TCP/IP connection between the robot and the computer.
2. To check, enter the Robot's IP Address into a web browser and the robot's home page should load. See Figure 12-3 as an example.
3. On the robot teach pendant, navigate to the system variable, $KAREL_ENB. Make sure this is set to 1. If it is set to 0, change it to 1 and cycle power on the controller. See Figure 12-4.

*Figure 12-4 Robot System Variables ($KAREL_ENB)*

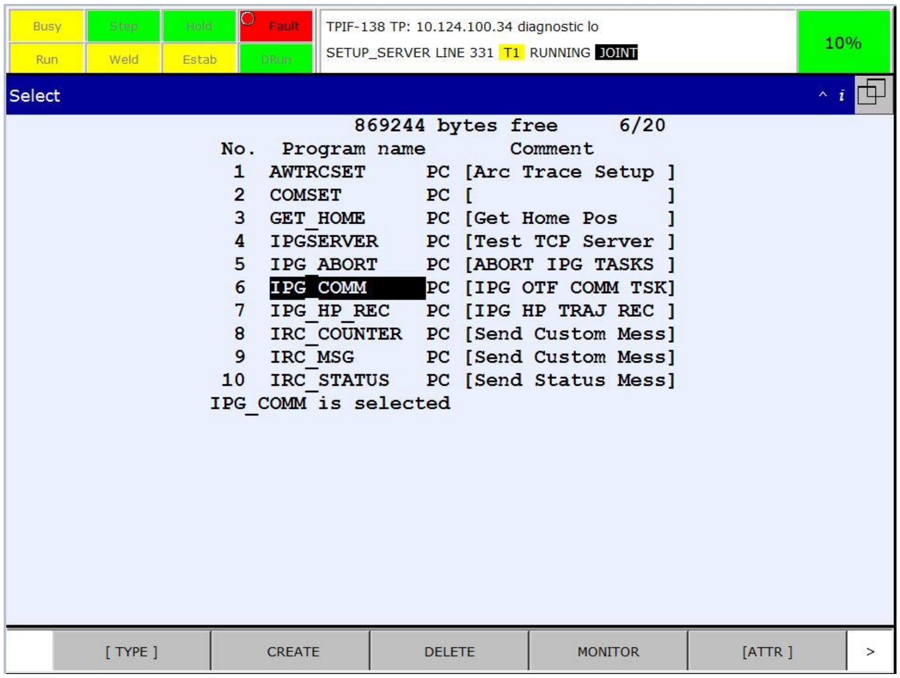



4. Load the following KAREL files (.PC) onto the robot. These files can be obtained by referring to section "Robot Configuration Files."
    a. IPG_ABORT.PC
    b. IPG_COMM.PC
    c. IPC_HP_REC.PC

**IMPORTANT** **It's important that the correct version files are uploaded to the robot based on the software version the robot is running (version 8 or version 9).**

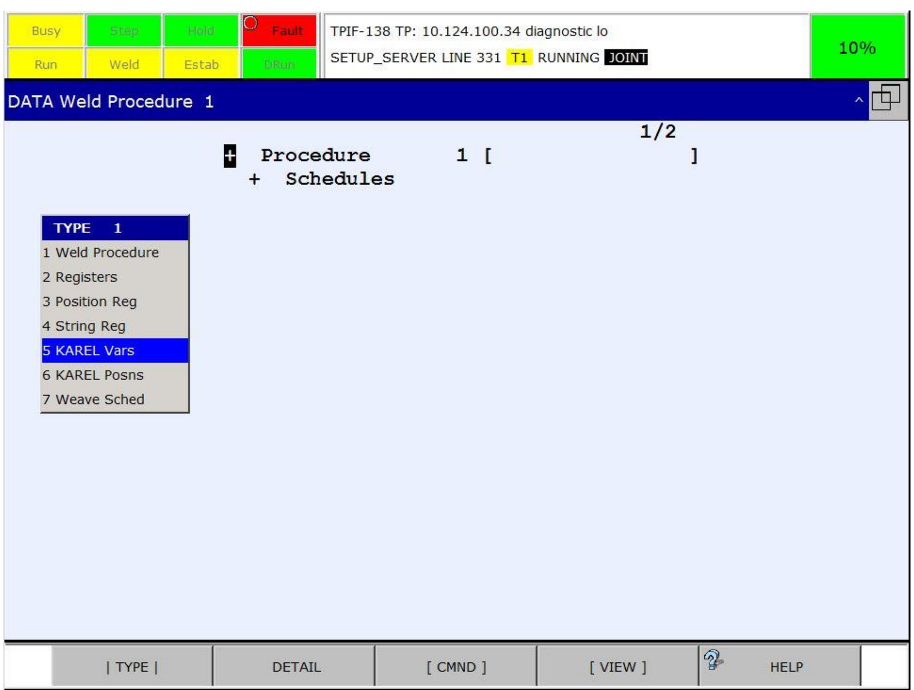
5. Configure the TCP server tag using the robot teach pendant. Instructions for configuration can be found in the KAREL Reference Manual Revision I, Section 11.3.2. (this is a FANUC document).
    a. Be sure to record the port number configured in this step. This number will need to be entered in the IPGScan robot options in order to be able to connect to the robot in IPGScan.
6. Select IPG_COMM.PC in the SELECT menu. See Figure 12-5.

*Figure 12-5 Selecting IPG_COMM*




7. Press the “DATA” button on the robot teach pendant to open the DATA menu.
8. Select “TYPE” and click “KAREL Vars” to open the KAREL variables. See Figure 12-6.
    a. Be sure that the teach pendant is set to display “Single.” The KAREL variables may not appear if the display is set to two or more windows.

*Figure 12-6 Data - KAREL Vars Menu*

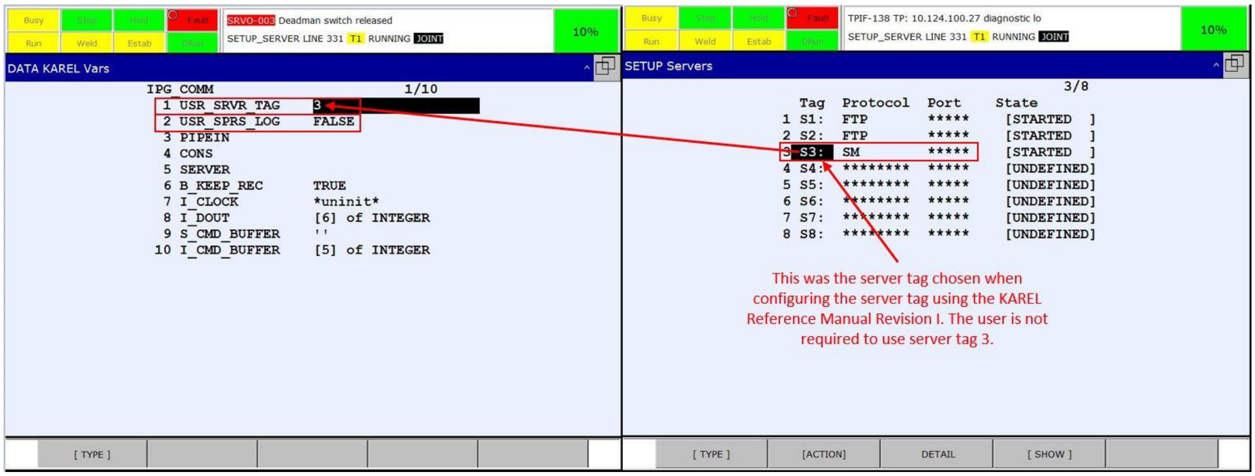



9. Set the appropriate values for “USR_SRVR_TAG” and “USR_SPRS_LOG.” See Figure 12-7.

a. USR_SRVR_TAG – enter the number of the TCP Server Tag. This number represents the server tag number that was selected during step 8 of the “Setting up a Server Tag” in the FANUC KAREL Reference Manual.

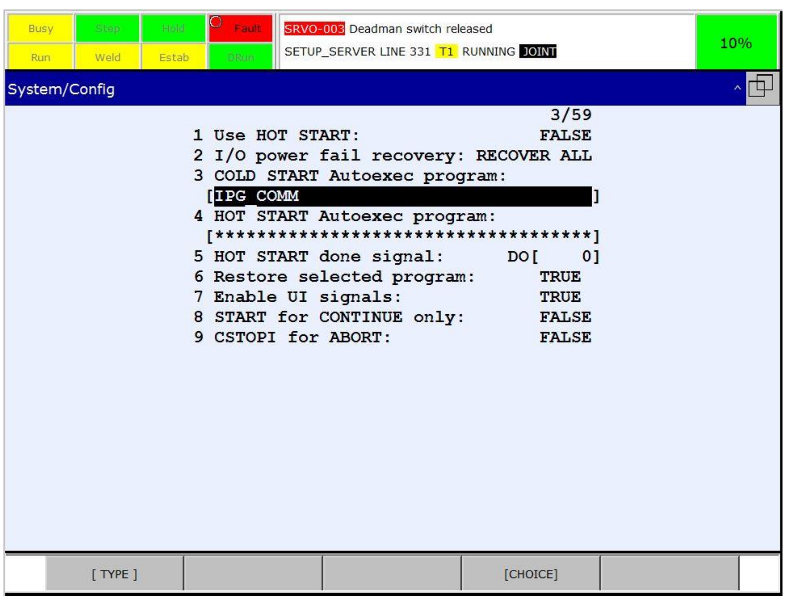
b. USR_SPRS_LOB – enter TRUE or FALSE. With TRUE, logging to “CONSLOG.DG” will be suppressed. If undefined, FALSE is assumed.

*Figure 12-7 Setting DATA KAREL Vars*




10. Add IPG_COMM to the “COLD START Autoexec program” setting in the FANUC Config Screen. See Figure 12-8.

*Figure 12-8 Setting the COLD START Autoexec Program*




KAREL is now setup and the user is ready to configure robot settings in IPGScan. See section, “Connecting to a Robot,” for additional details.

##### 12.2.3.2.1 KAREL Notes

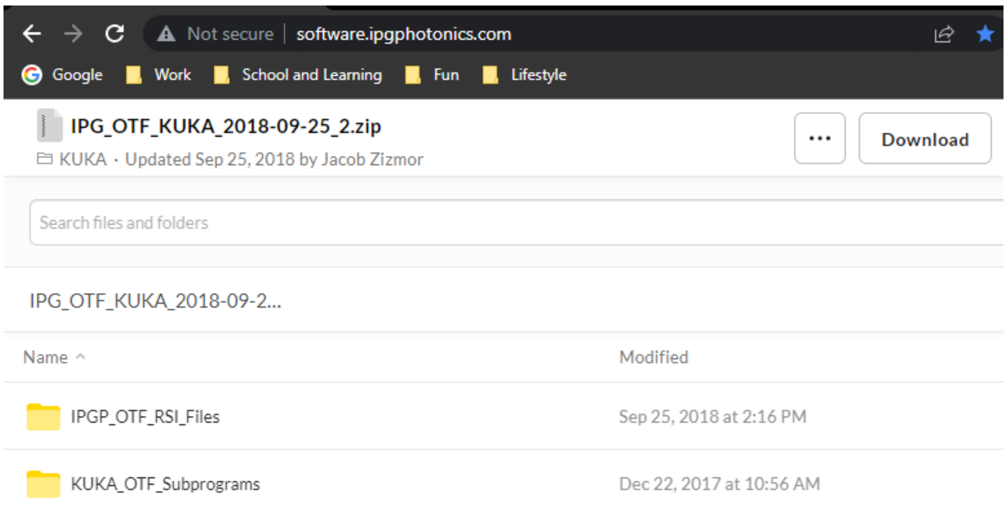
- IPGScan communication with the robot will not work without IPG_COMM running. If a fatal error occurs, if aborted, or if not configured to automatically start with the robot, IPG_COMM can be started from the SELECT Screen.
- IPG_COMM will not change either the “Busy” status or the Active light on the robot controller.
- IPG_COMM or IPG_HP_REC will not respond to the abort button. Run IPG_ABORT to terminate.
- Error messages are located on the “User” Screen of the robot teach pendant. Log messages are recorded in the log “CONSLOG.DG” (if not suppressed).
- If the active tool frame has changed, run IPG_ABORT and then IPG_COMM to reinitialize.

### 12.2.4 KUKA Setup

The following steps detail the configuration of a KUKA robot for robotic OTF processing.

1. Setup RSI on the robot controller. Please refer to the appropriate KUKA documentation.
2. Download the KUKA files from the IPG Software website.
    a. The appropriate OTF download folder will contain a folder named “IPGP_OTF_RSI_Files” and “KUKA_OTF_Subprograms.” See Figure 12-9.

*Figure 12-9 KUKA OTF Files for Download*

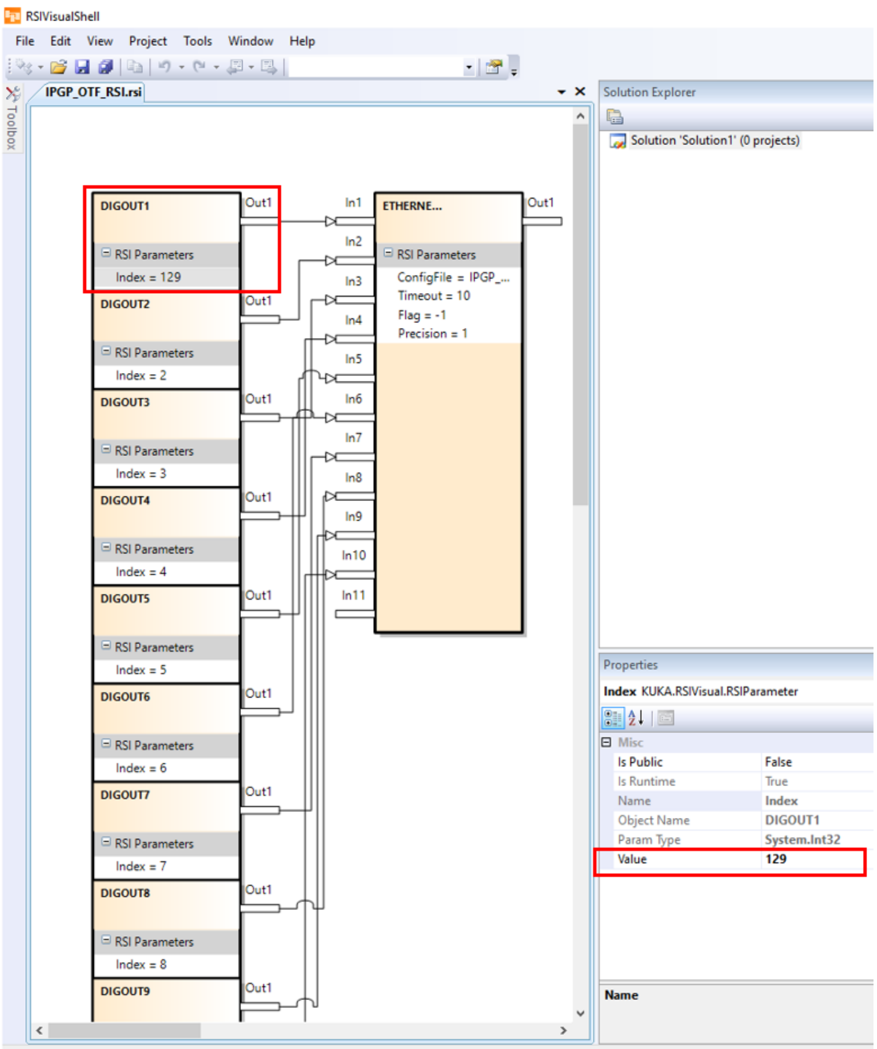



3. Configure the “IPGP_OTF_RSI.rsi” file to setup the START bit and any desired flags.

**IMPORTANT** **RSIVisualShell is required to modify the “IPGP_OTF_RSI.rsi” file. RSIVisualShell should be acquired from KUKA.**

    a. START bit Configuration
        i. Set the appropriate Index value for “DIGOUT1” based on the digital output bit on the robot that corresponds with the scanners START bit. Figure 12-10 provides an example of the “DIGOUT1” bit being configured for digital output bit 129.

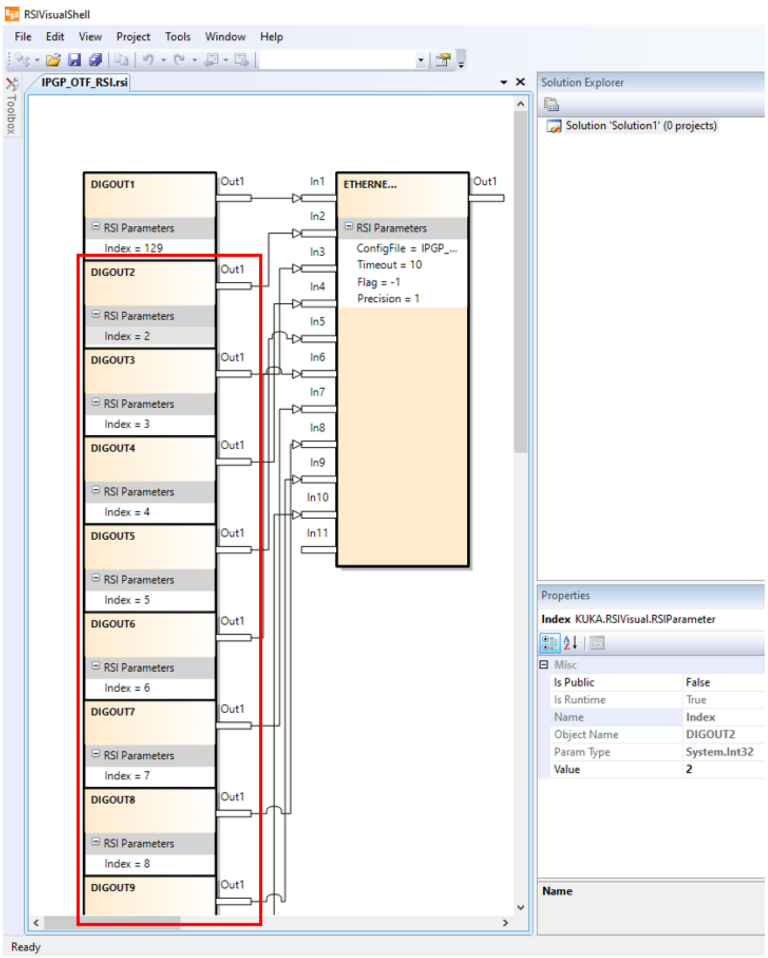
*Figure 12-10 START Bit Configuration in RSIVisualShell*




b. Flag Configuration (optional)

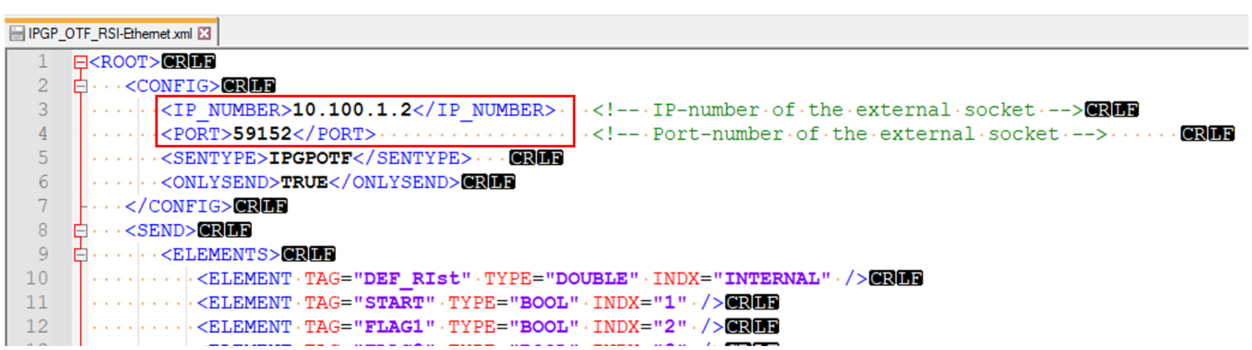
i. Set the desired Index values of "DIGOUT2" through "DIGOUT10" for any desired flags. Please also refer to section, "KUKA." See Figure 12-11.

*Figure 12-11 FLAG Configuration in RSIVisualShell*




4. Open the "IPGP_OTF_RSI-Ethernet.xml" file in a text editor and modify the "IP_Number" and "Port" parameters as required. See Figure 12-12 as an example.
    a. Modify the IP Number parameter to contain the IP Address that will be set on the computers local area adapter connection for connecting with the RSI Interface of the KUKA robot.
    b. Enter a desired Port number value. This must also be entered into the IPGScan Robot Options (see section "Connecting to a Robot").

*Figure 12-12 IPGP_OTF_RSI-Ethernet File*



5. Load the following files onto the KUKA controller to the directory specified by the KUKA RSI manual:
    a. IPGP_OTF_RSI.rsi
    b. IPGP_OTF_RSI.rsi.diagram
    c. IPGP_OTF_RSI.rsi.xml
    d. IPGP_OTF_RSI-Ethernet.xml

KUKA robot configuration is now complete and users can continue with system setup.

To assist users with RSI control, the following modules can be called:

- IPG_OTF_BEGIN_DATA_XFER.src
    a. This will start the transfer of robot positional data to IPGScan.
- IPG_OTF_END_DATA_XFER.src
    a. This will stop the transfer of robot positional data to IPGScan.

The following module can be used for the IPGScan and Robot Workspace calibration as an example:

- IPG_OTF_CALIBRATION.src
- IPG_OTF_CALIBRATION.dat

The robot is now setup and the user is ready to configure robot settings in IPGScan. See section, "Connecting to a Robot," for additional details.

### 12.2.5 Yaskawa Motoman Setup

The following steps provide an overview of the setup required for robotic OTF processing with a Motoman robot.

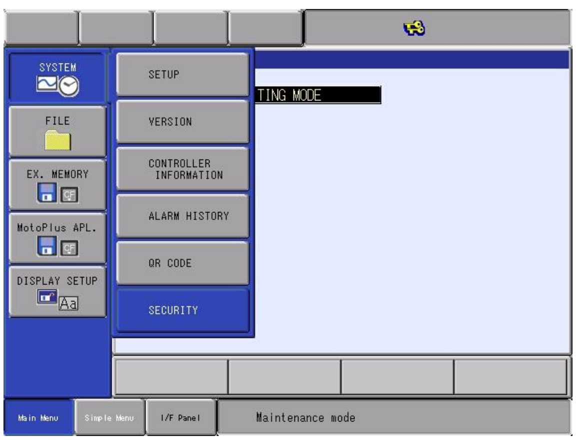
1. Setup MotoPlus on the robot controller.
2. On the robot controller, load “IPG_OTF_DX200.out” (see *MOTOPLUS APPLICATION INSTALLATION INSTRUCTIONS* from Yaskawa Motoman (PN 166687-1CD) for instructions.).
3. Connect the robot and computer using an Ethernet connection. Configure the IPv4 Windows Network Adapter properties according to the network settings on the robot.
    a. The specified port for Motoman is 50245

The following sections provide a detailed setup procedure for the above steps.

#### *12.2.5.1 MotoPlus Setup*

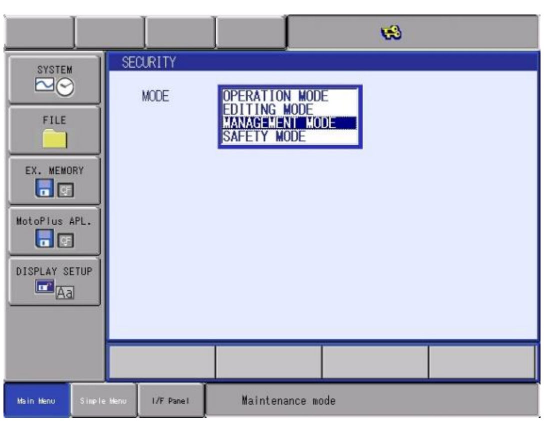
1. Start the robot and controller in Maintenance Mode by holding “Menu” on the teach pendant while powering up the controller.
2. Once the controller has boot-up, click “System” and select “Security.” See Figure 12-13.

*Figure 12-13 Selecting Security*



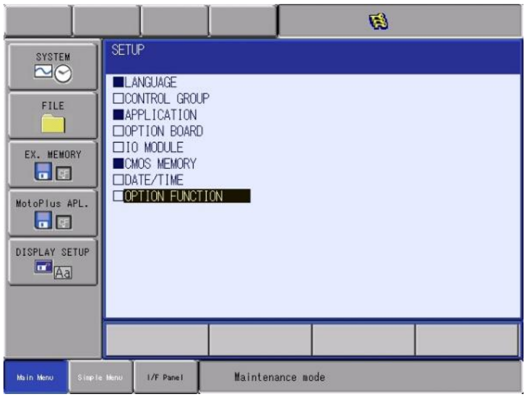

3. Select “Management Mode” and enter the password. See Figure 12-14.
    a. The default password consists of all 9s.

*Figure 12-14 Selecting Management Mode*



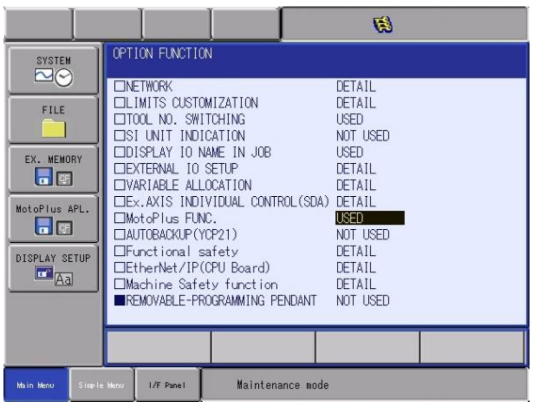
4. Next, select “System” and click “Setup.”
5. Enter “Option Function.” See Figure 12-15.

*Figure 12-15 Option Function Menu*




6. Cursor down to “MotoPlus FUNC.” and set it to “USED.” See Figure 12-16.

*Figure 12-16 MotoPlus FUNC.*




### 12.2.5.2 Loading the “IPG_OTF_DX200.out” File

1. Insert a USB with the “IPGP_OTF_DX200.out” file into the teach pendant.
    a. This file can be obtained by referring to section “Robot Configuration Files.”
2. With the controller still in Maintenance mode, select “System” and click “Load (User Application)”. See Figure 12-17.

*Figure 12-17 Selecting the IPGP_OTF_DX200 File*

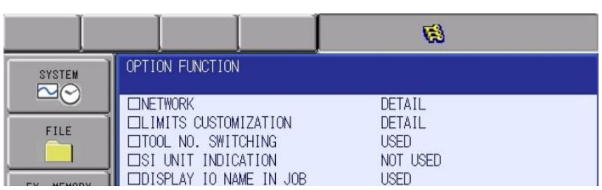

3. Select the “IPGP_OTF_DX200.out” file (a star should appear next to the file name once it is selected).
4. Once selected, click “Enter” and click “YES” in order to load the file. See Figure 12-18.

*Figure 12-18 Loading the IPG_OTF_DX200 File*


## 12.2.5.3 Ethernet Connection Setup

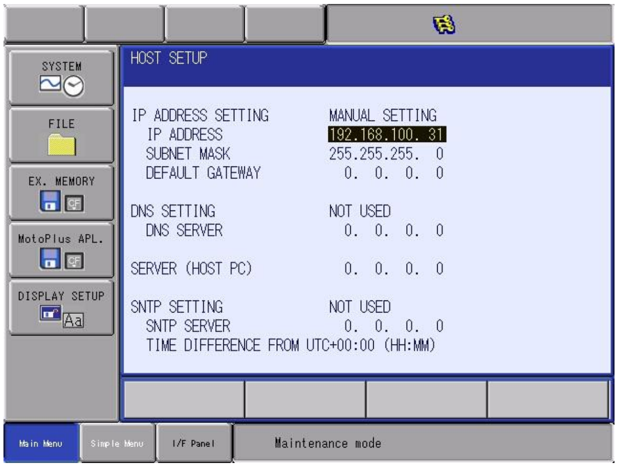
1. Select “System.”
2. Select “Setup.”
3. Select “Option Function.”
4. Select “Network.” See Figure 12-19.

*Figure 12-19 Select Network*




5. Navigate to the “IP Address” Setting. From here, the user can either set a desired IP address or simply take note of the IP address that is already assigned to the robot. See Figure 12-20.
   a. Note that after the IP address is changed in the Motoman, “Enter” must be pressed multiple times on the teach pendant in order to confirm the changes.

*Figure 12-20 The Robots IP Address*




6. Power cycle the robot.
7. Make sure the local area connection on the PC that is used to connect to the robot is set to an appropriate IP address.
    a. To verify this is correct, attempt to ping the robot's IP address from the computer.

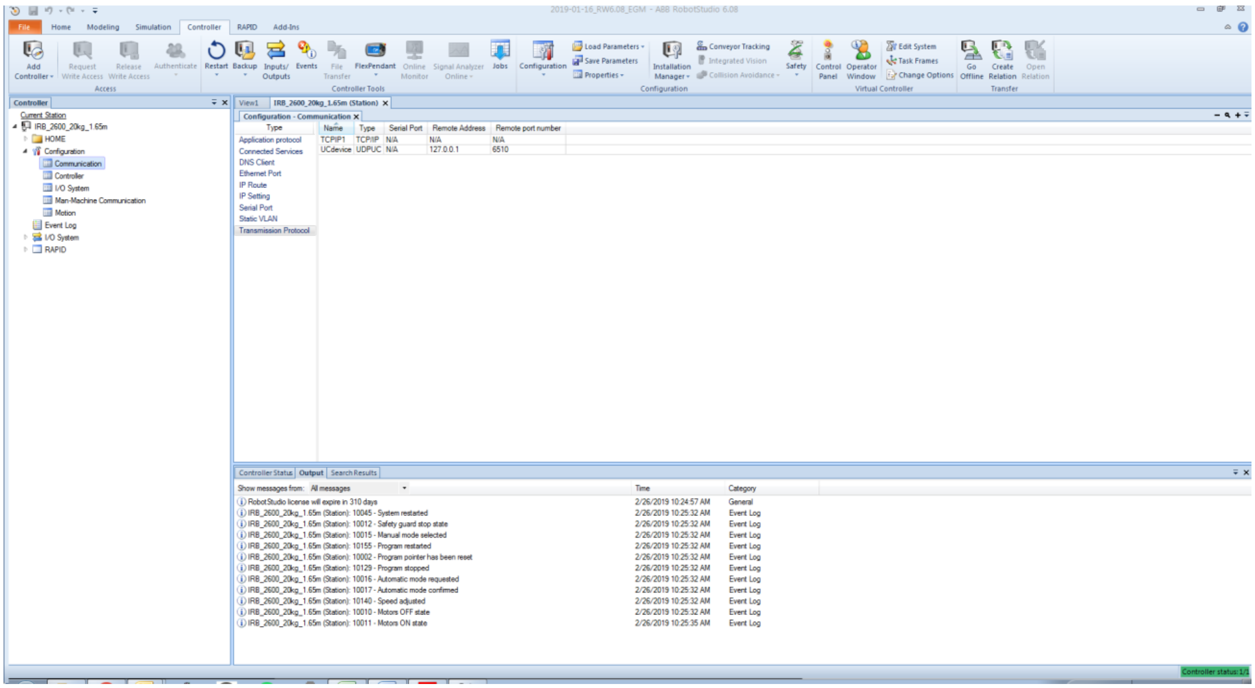
The robot is now setup and the user is ready to configure robot settings in IPGScan. See section, "Connecting to a Robot," for additional details.

### 12.2.6 ABB Setup

The following steps detail the configuration of an ABB robot for Robotic OTF processing.

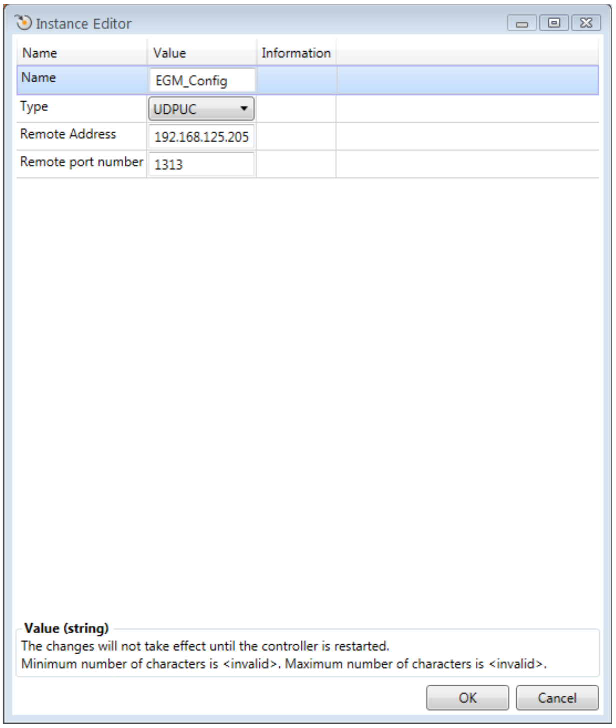
1. Configure Transmission Protocol.
    a. Connect to the controller in RobotStudio.
    b. Go to the Transmission Protocol configuration under "Configuration / Communication / Transmission Protocol." See Figure 12-21.

*Figure 12-21 Transmission Protocol Configuration*




    c. Right click and select "New Transmission Protocol…"
    d. Change the "Type" to "UDPUC."
    e. Change the "Remote Address" to the IP Address of the computer running IPGScan that will be connected to the robot.
    f. The "Name" and "Remote port number" have no additional restrictions or requirements. See Figure 12-22.

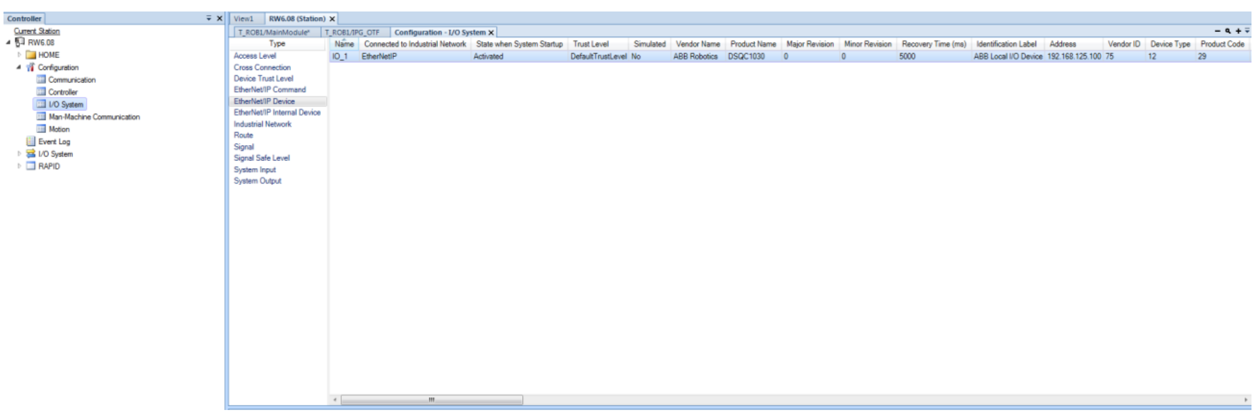
*Figure 12-22 Instance Editor Window*




   g. Press “OK” and restart the controller.

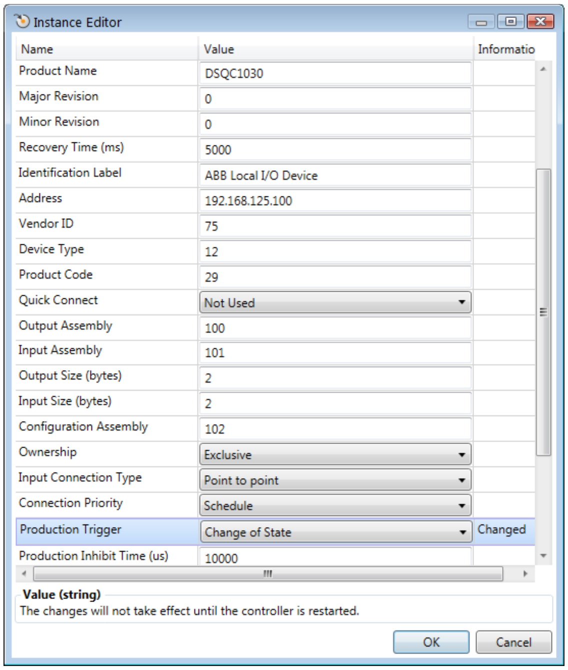
2. If applicable, configure the IO Device for Change of State.
   a. Connect to the controller in RobotStudio.
   b. Go to the Ethernet/IP Device configuration under I/O System configuration. See Figure 12-23.

*Figure 12-23 Ethernet/IP Device Configuration*



   c. Double click on the Ethernet/IP Device which has the “Start” signal
   d. In the Instance Editor, find the “Production Trigger” setting. Make sure that this setting is set to “Change of State.” Figure 12-24.

*Figure 12-24 Change of State Setting*




3. Load the System Module "IPG_OTF.sys" onto the controller.
    a. This file can be obtained by referring to section "Robot Configuration Files."

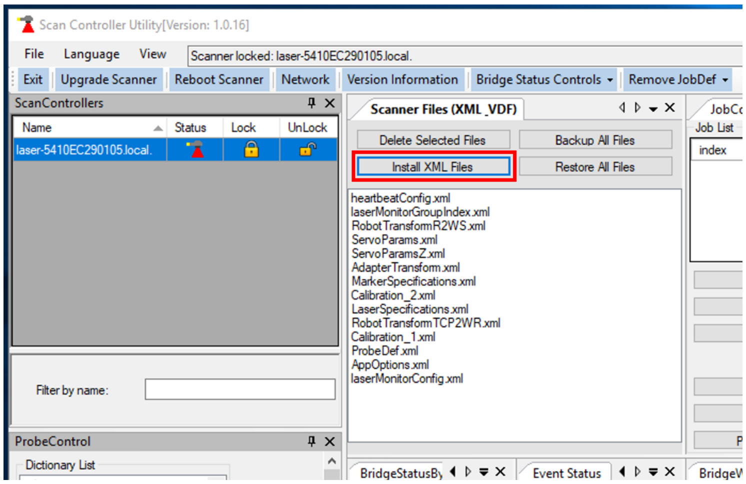
The robot is now setup and the user is ready to configure robot settings in IPGScan. See section, "Connecting to a Robot," for additional details.

## 12.3 Scan Controller Requirements

Robotic OTF processing requires that Scan Controllers have a "CoordinationParams.xml" file uploaded to the controller. While some scanners may ship from IPG production with the file, others may not. The following procedure outlines how users can upload this file to a Scan Controller.

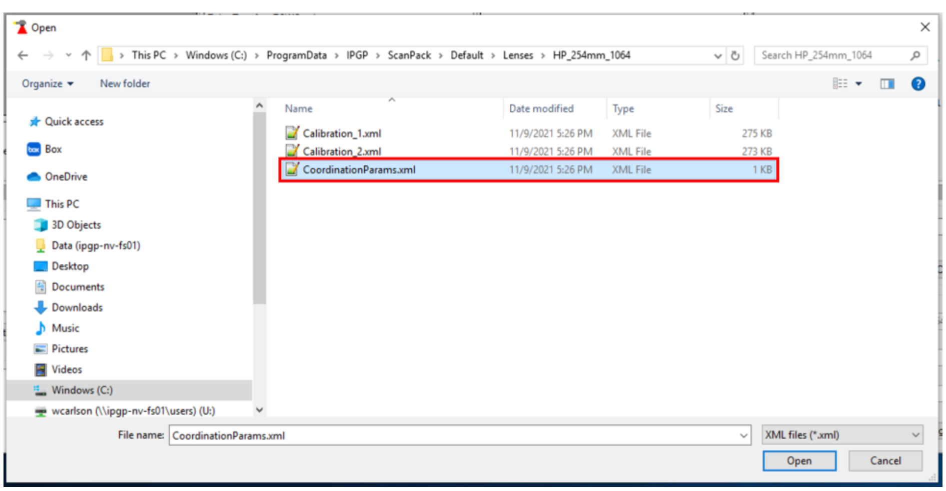
1. Connect to the desired scanner in the Scan Controller Utility.
2. Click the "Install XML Files" button. See Figure 12-25.

*Figure 12-25 Installing an XML File*




3. Open the "Default" folder.
4. Open the "Lenses" folder.
5. Open any of the following folders:
   a. HP_254mm_1064
   b. HP_415mm_1064
   c. HP_510mm_1064
6. Select the "CoordinationParams.xml" file. See Figure 12-26.

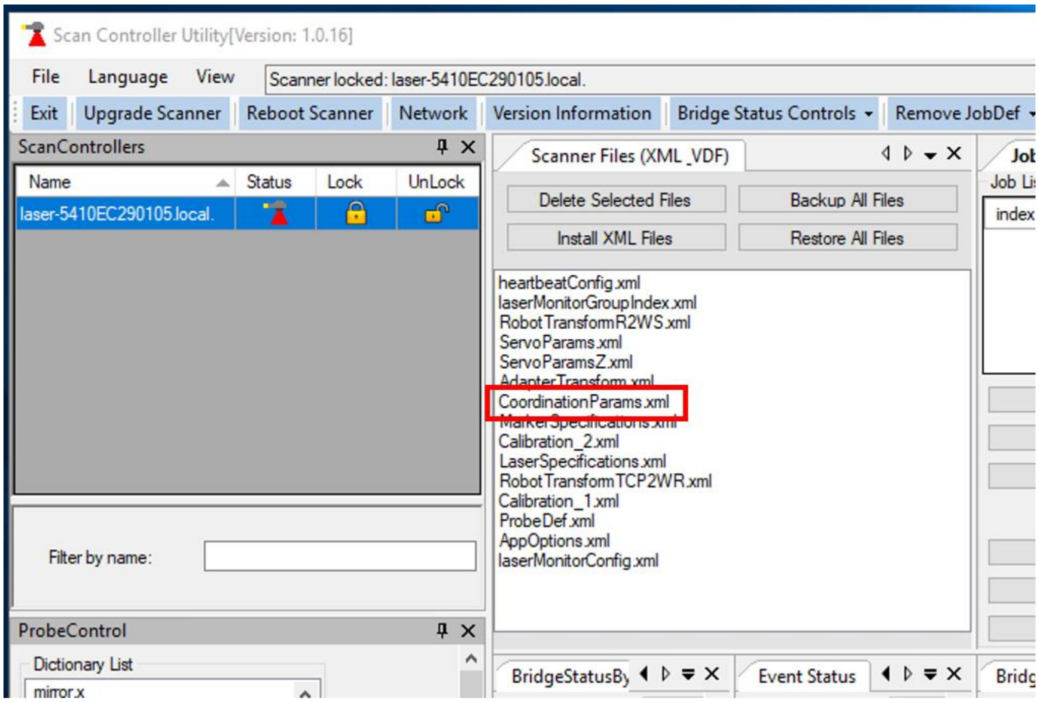
*Figure 12-26 Selecting the CoordinationParams File*




7. Click "Open."

8. Click “OK” to acknowledge the installation.
9. Reconnect to the Scan Controller and verify the file was uploaded. See Figure 12-27.

*Figure 12-27 CoordinationParams.XML*




Users can now disconnect from the Scan Controller in the Scan Controller Utility and proceed with other setup and use.

## 12.4 IPGScan Robotic OTF Programming

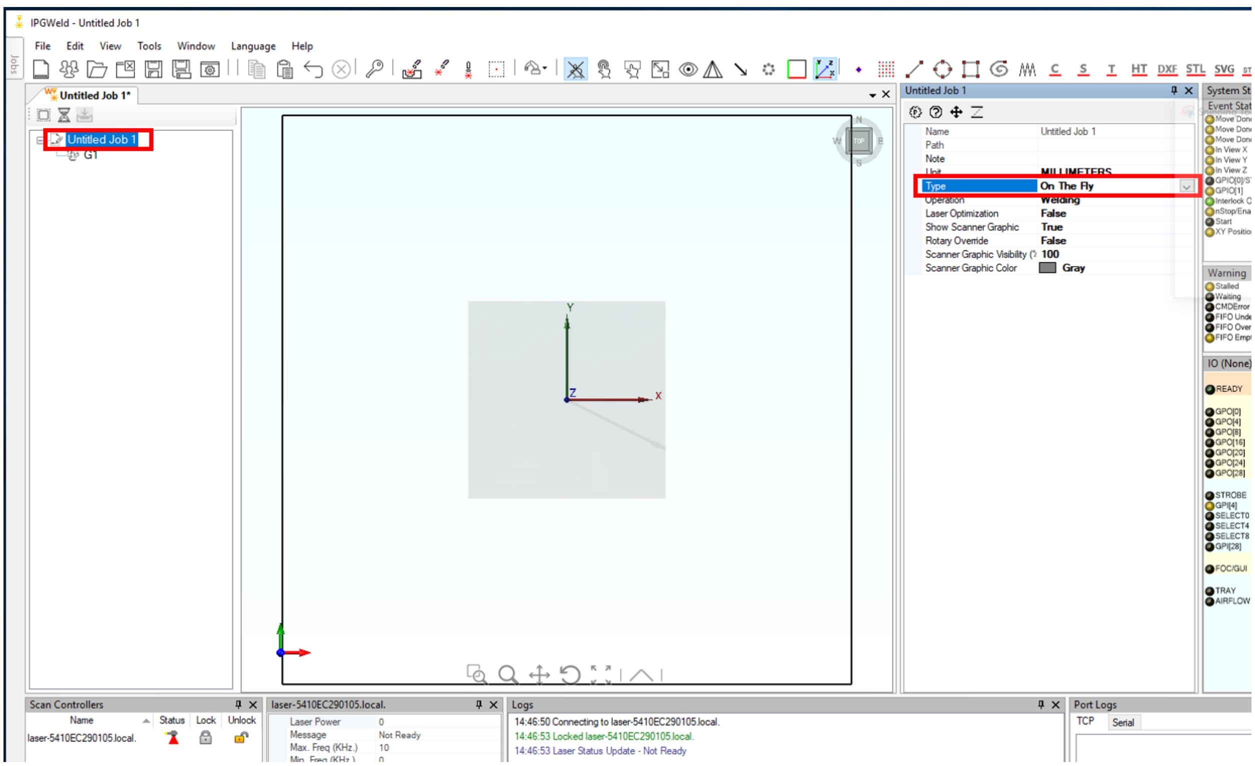
The following sections outline how users can create a Robotic OTF job, necessary setup procedures, and process specific features.

### 12.4.1 Creating a Robotic OTF Job

Users can create a Robotic OTF job by following the steps below:

1. Open IPGScan.
2. Create a new job.
3. Select the job name in the Job Tree.
4. In the Properties window, change the Type parameter to "On The Fly." See Figure 12-28.

*Figure 12-28 Creating an On The Fly Job*




**IMPORTANT** **Type should be "Default" or "Point & Shoot" for any jobs that are not On-The-Fly. When a Robot Type is set to "None" in the IPGScan Robot Options (see section** "IPGScan Robot Options"**), newly created jobs will be created as "Default" type. When Robot Type is set to anything other than "None," newly created jobs will be "On The Fly" type by default.**

Users can now proceed to importing CAD models, placing Process Objects, and additional setup procedures.

### 12.4.2 STL Models and Preliminary Process Object Placement

IPGScan supports the import of STL models into jobs. The purpose behind importing STL models into the job is so that users have a visual aid that assists with the placement of process objects and the creation of the overall process. This enables users to be able to create the majority of the process offline and before equipment is setup on the production floor. Furthermore, the use of STL models for programming can reduce programming efforts later in the process when transitioning to the real world equipment. This is accomplished through the use of the "Part Alignment for STL" functionality, which is covered in a later section.

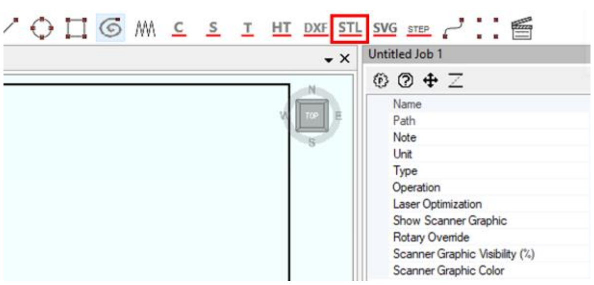
The following sections outline how users can import STL models and also provides an example for placing Process Objects according to the STL models.

#### *12.4.2.1 Importing STL Models*

Users can import STL models using the following procedure.

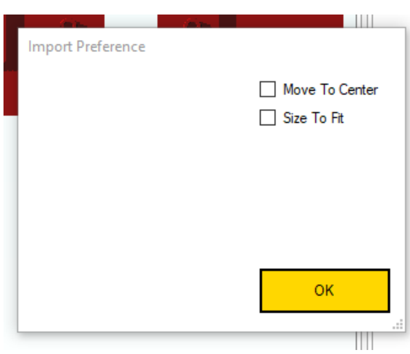
1. Open IPGScan and create an "On The Fly" type job.
2. Click the "STL" button in the Tool Bar. See Figure 12-29.

*Figure 12-29 STL Import Button*




3. Navigate to the desired STL model for import and select the file.
4. Click "Open."
5. Select the desired Import Preference options. See Figure 12-30.
    a. Move To Center – This will center the STL model visual at the 0, 0, 0 location of the IPGScan canvas.
    b. Size To Fit – This will scale the STL model to fit completely within default scanner FOV (not typically used).

*Figure 12-30 Import Preference Options*




6. Click "OK."

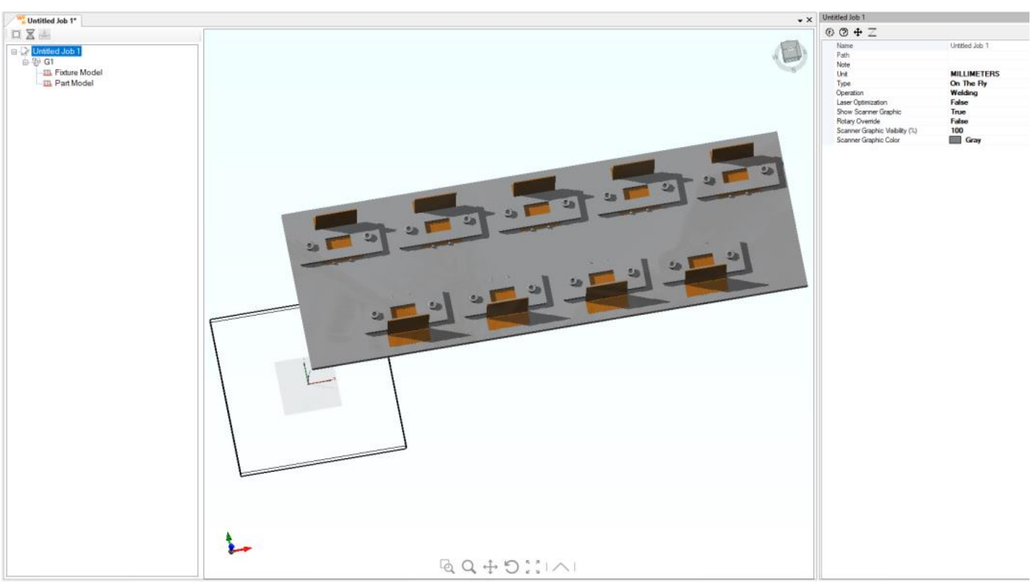
**IMPORTANT** **Depending on the file size of the STL model, it can take some time for the model to import.**

With the STL model imported, users can now select the STL model in the Job Tree and modify parameters as desired. Figure 12-31 provides an example of an imported model.

*Figure 12-31 Example Imported STL Model*


Users can import multiple STL models if desired. By importing multiple STL models and assigning each model with a different color, users can create a visual in IPGScan that makes it easier to distinguish a part from a fixture, which may assist with Process Object placement. Figure 12-32 demonstrates how two STL models can be imported into a single job. One model represents a fixture while the other model is for the parts in the fixture.

*Figure 12-32 Multiple STL Models Imported in One Job*

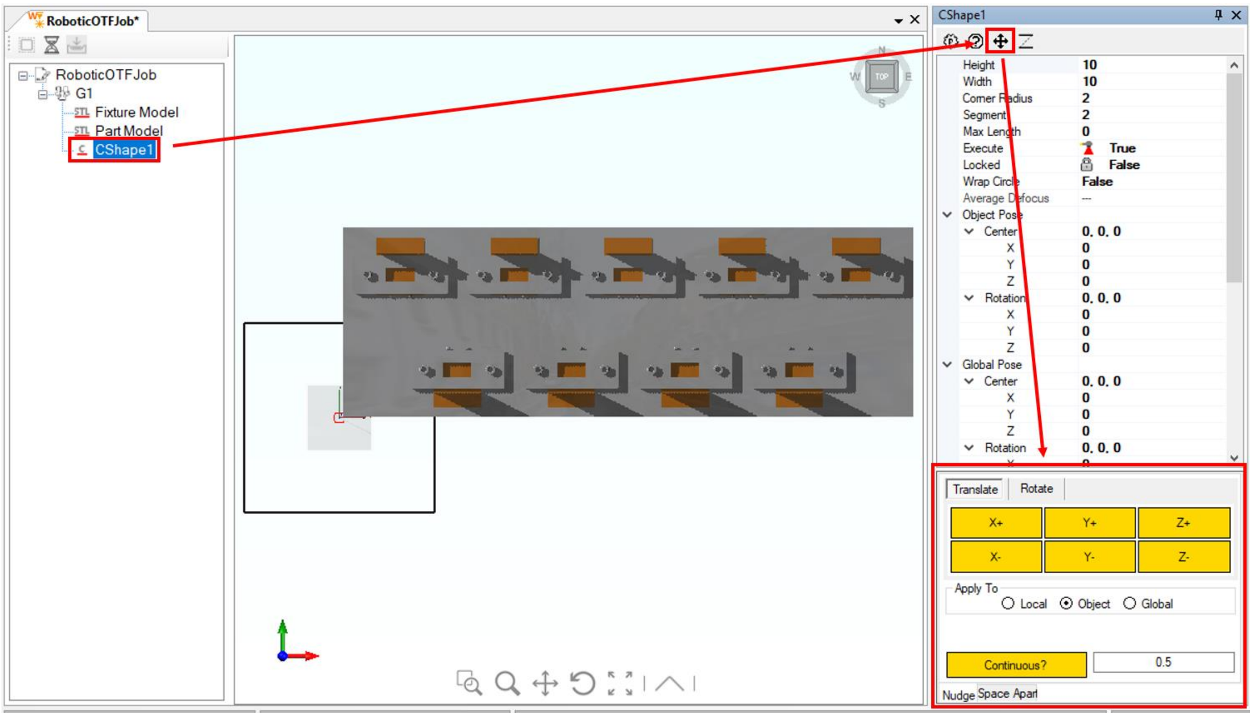


### 12.4.2.2 Preliminary Process Object Placement

Once users have imported any desired STL models, Process Objects can then be placed accordingly to the STL models. The following steps detail how users can utilize the Nudge Tool to place Process Objects relative to the STL models.

1. Create any desired Process Object using the Tool Bar.
2. Select the Process Object in the Job Tree and click the Nudge Tool button to open the Nudge Tool window. See Figure 12-33.

*Figure 12-33 Opening the Nudge Window*



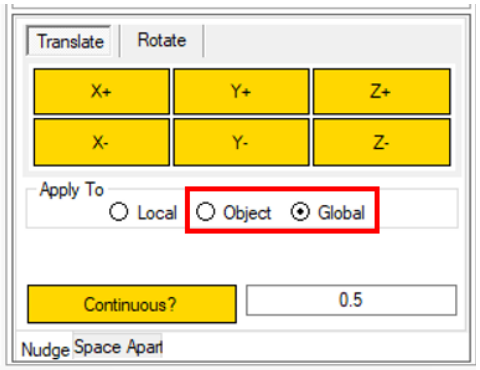

3. Change the "Apply To" setting from "Object" to "Global." See Figure 12-34.

**IMPORTANT**

**For Robotic OTF applications, it is highly recommended that users only change Global Pose position (center and rotation) information. Any changes in Object Pose to a Process Object will offset the preview image of the guide laser when utilizing Process Alignment functionality for Process Object placement/touch-up.**

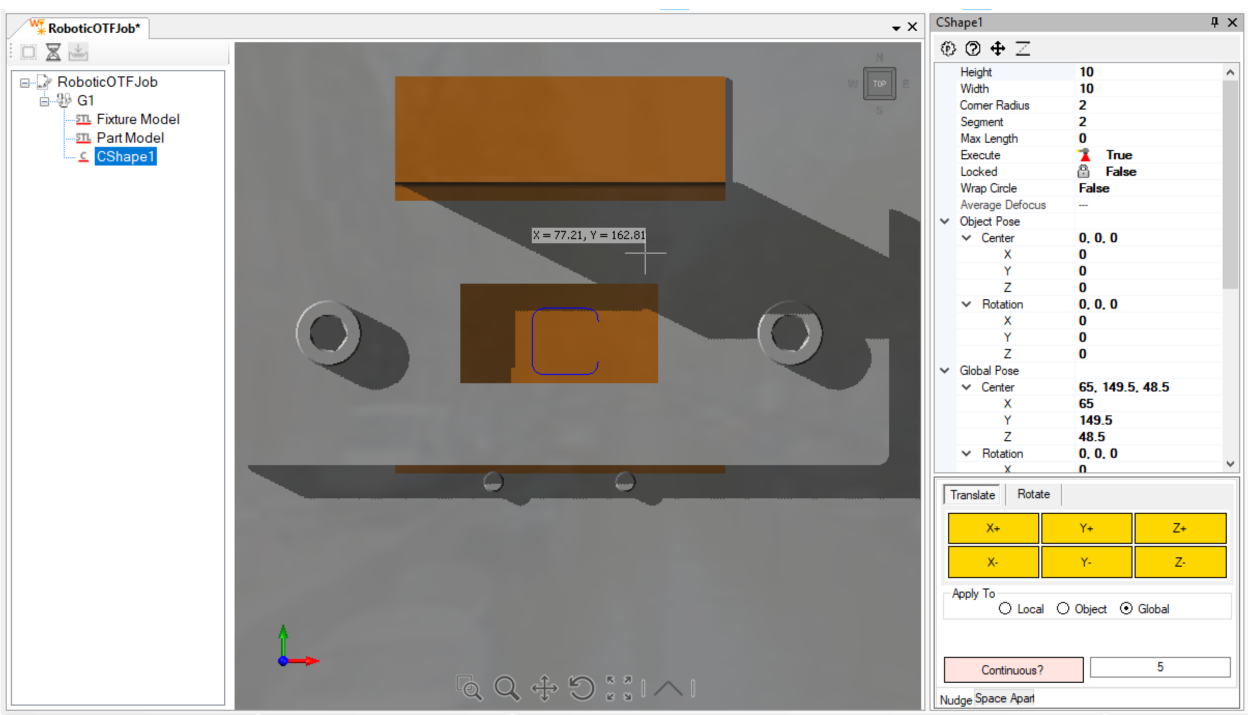
**IPGScan Position Information Order of Operations: Object Scale + Object Rotation + Object Translation + Group Scale + Group Rotation + Group Translation + Global Rotation + Global Translation**

*Figure 12-34 Nudge about Global Pose*



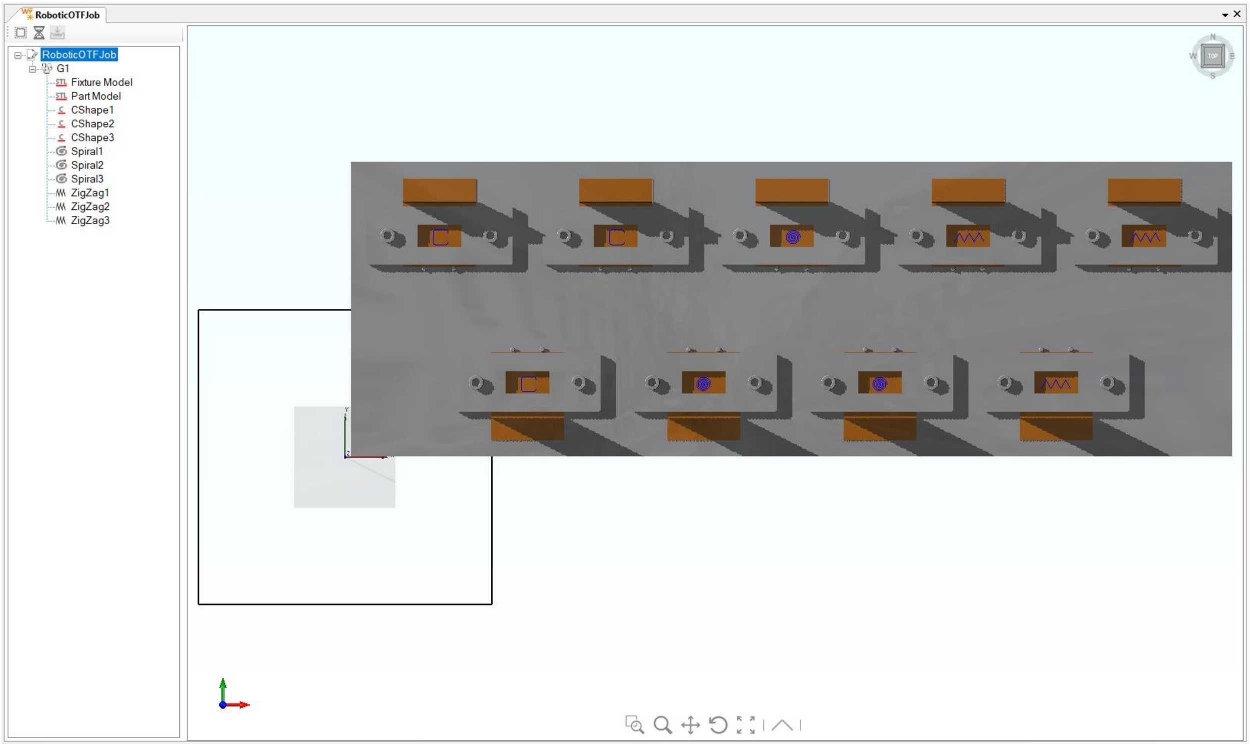

4. Users can now nudge the process object into the appropriate location relative to the CAD model by using the X, Y, and Z buttons in the Nudge Tool window. See Figure 12-35.
    a. Continuous Button – When enabled, users can click and hold the X, Y, or Z Nudge buttons for continuous repositioning. When disabled, each mouse click will cause the Process Object to move one time (based on the defined increment amount).
    b. Increment Amount – Users can change the increment amount value as desired.

*Figure 12-35 Nudging a Process Object into Position*




5. Users can now copy, paste, or create new Process Objects and position them relative to the STL model(s) as desired using the Nudge Tool. See Figure 12-36, which details an example weld fixture with Process Objects placed in the appropriate weld locations.

*Figure 12-36 Example Weld Fixture with Process Objects*




With all desired Process Objects placed relative to the STL model(s), users can transition into creating a Simulated Robot Trajectory in order to start visualizing the process and determining cycle times.

### 12.4.3 Simulated Trajectory and Projection Volume (Head Preview)

The combination of the Simulated Trajectory and Projection Volume functionality allows users to visualize the scanning process without ever having setup any equipment in the real world. Simulated Trajectory enables users to create an expected robot trajectory that will emulate what the real robot trajectory will resemble, while Projection Volume allows users to view the Scan Head and laser beam as it executes Process Objects. Through the combination of these two features, users can visualize and develop the process prior to ever stepping on the production floor.

The following sections detail the functionality of Projection Volume and Simulated Trajectory.

#### *12.4.3.1 Projection Volume (Head Preview)*

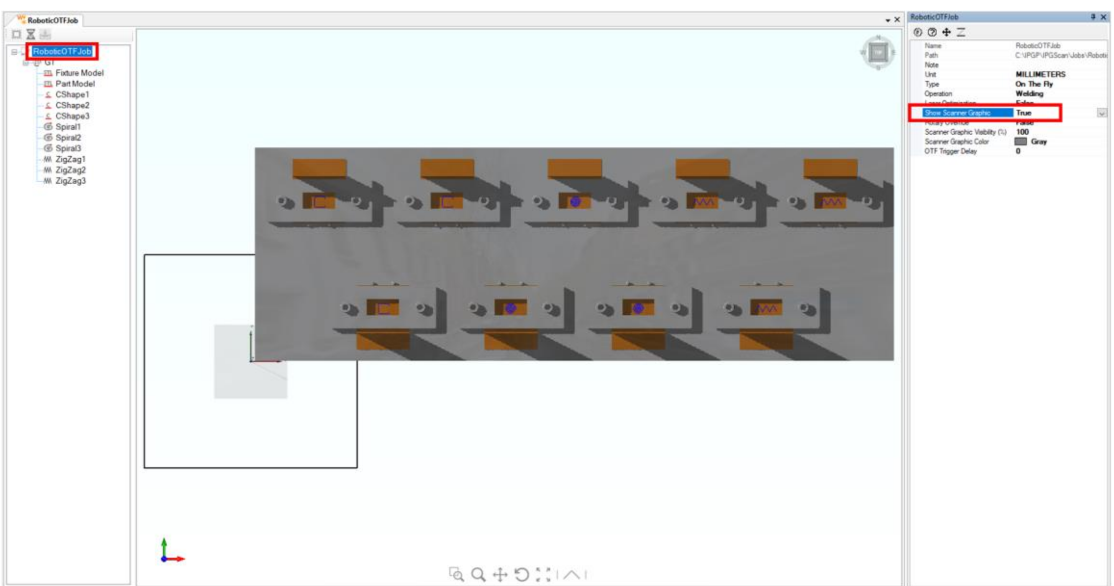
The Projection Volume feature allows users to display a visual of the scan head as well as the beam trajectory in the IPGScan canvas. This feature can be useful for visualizing the process and better understanding of whether or not obstructions may exist in the path of the laser beam when attempting to process a part.

| | |
|---|---|
| **IMPORTANT** | **The scanners calibration files require specific parameters for this functionality. If this feature is enabled and the head and beam trajectory are not displayed, it is likely that the calibration files do not have the required parameters. Please contact the appropriate IPG Beam Delivery support personnel concerning the update of the calibration files.** |

The following procedure outlines how users can enable the Projection Volume feature.

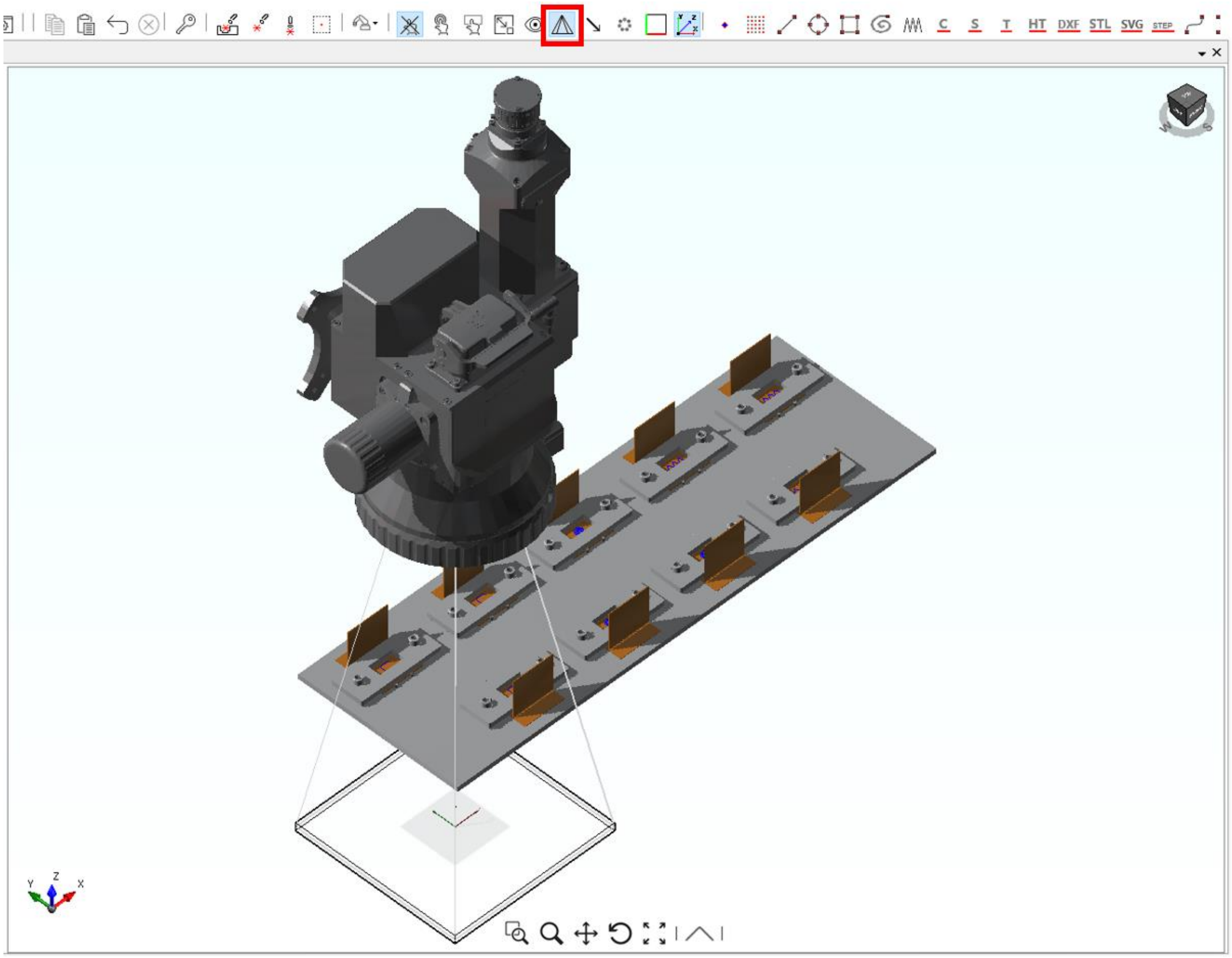
1. Open IPGScan and connect to the desired scanner.
2. Create a new IPGScan job or open an existing job.
3. Click on the job's name in the Job Tree.
4. In the Parameter Window, set "Show Scanner Graphic" to "True." See Figure 12-37.
    a. Here is where users can adjust the scanner graphic color and transparency.

*Figure 12-37 Enabling Show Scanner Graphic*



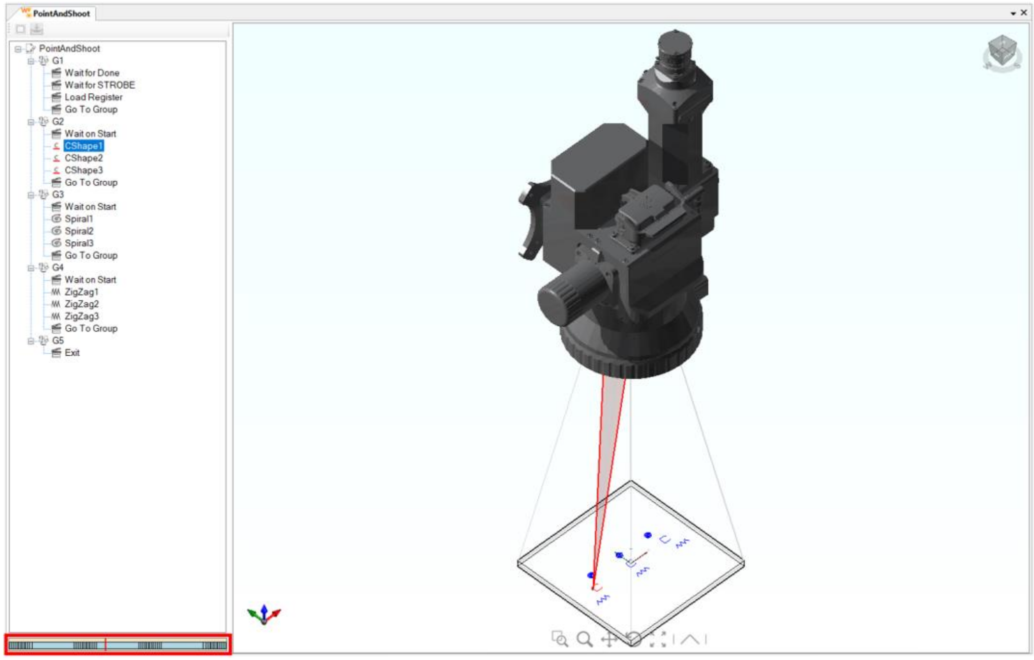
5. Click the “Show Projection Volume” button in the Tool bar. This will cause the head to appear in the IPGScan canvas. See Figure 12-38.

*Figure 12-38 Projection Volume Display*



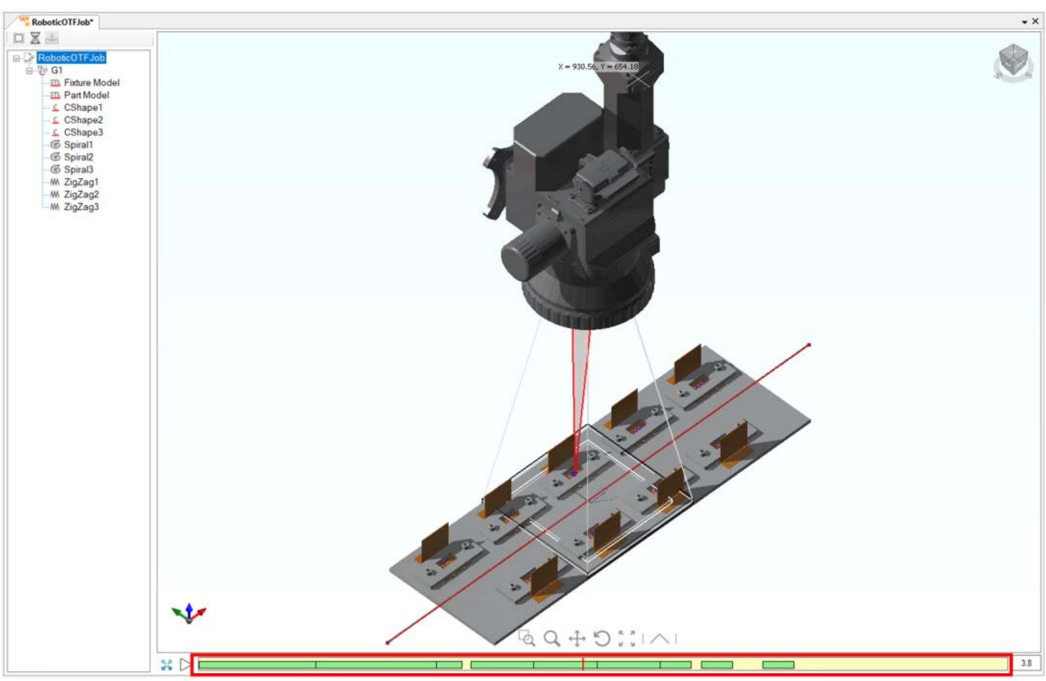

6. With “Show Projection Volume” enabled, users can then preview the beam trajectory by scrubbing along the appropriate timelines for the given job types.
    a. Default and Point & Shoot Job Types – Select a Process Object in the job and scrub along the timeline that appears at the bottom of the Job Tree. See Figure 12-39.

*Figure 12-39 Projection Volume in a Point and Shoot Job*




b. Robotic OTF Job Type – A simulated robot trajectory or a real robot trajectory must be loaded into the job in order to preview the beam path. When either of these types of trajectories are loaded, users can scrub along the basic timeline to view the beam trajectory. Please refer to section "Simulated Trajectory" for details on creating a simulated trajectory. See Figure 12-40.

*Figure 12-40 Projection Volume in an On-The-Fly Job*



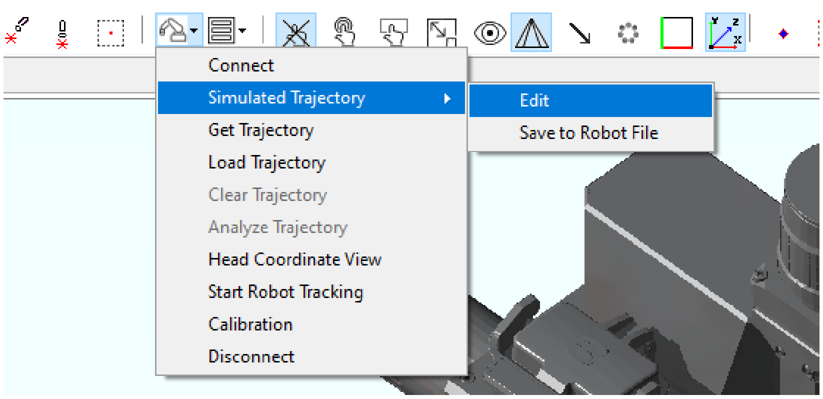
### 12.4.3.2 Simulated Trajectory

Simulated Trajectory allows users to create a desired/preliminary robot trajectory that can be utilized to assist with process development and visualization. By defining various robot positions and speeds, users can begin to visualize how the scan head will traverse along the parts and how the process will take place. Additionally, by utilizing Projection Volume along with Simulated Trajectory, users can better understand the approach of the beam during processing and whether or not obstructions from fixturing may exist and how laser on/off timings may need to be adjusted.

The following steps detail how users can create a simulated robot trajectory.

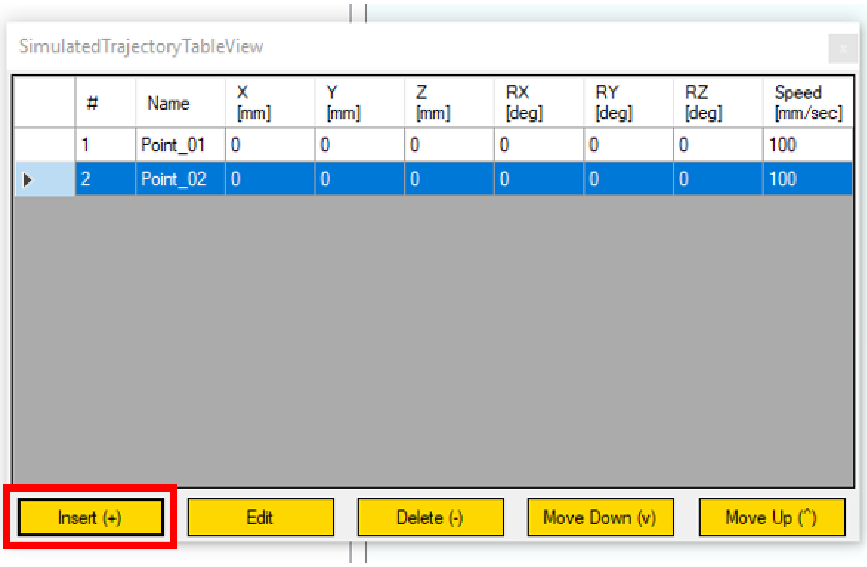
1. Open IPGScan and connect to the desired scanner.
2. Create a new IPGScan job that is an "On The Fly" type or open a previously created job.
3. (Optional) Enable the display of the scan head in the IPGScan canvas by turning on "Show Projection Volume." Please refer to the section "Projection Volume (Head Preview)," for more details concerning this feature.
4. In the Tool bar, click on the robot icon, navigate to "Simulated Trajectory," and click "Edit." See Figure 12-41.
    a. If no robot icon exists in the Tool bar, the user likely needs to configure the robot options in the IPGScan Options menu. A robot type must be specified in order to utilize Simulated Trajectory. Refer to section "IPGScan Robot Options" for details on specifying a robot type.

*Figure 12-41 Opening the Simulated Trajectory Window*



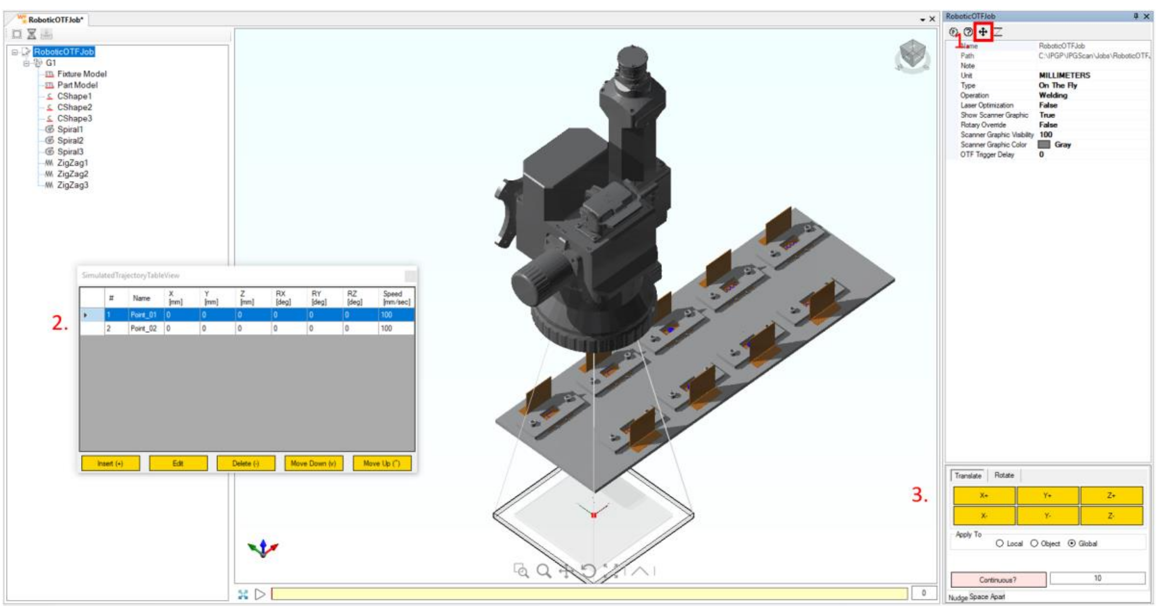

5. In the Simulated Trajectory Table View, users can add and position simulated trajectory points throughout the IPGScan canvas. Users can create simulated points by clicking the "Insert (+)" button. See Figure 12-42.
    a. For this example, two simulated trajectory points are created.

*Figure 12-42 Creating Simulated Trajectory Points*




6. Move the simulated trajectory points into the desired locations. This can be accomplished in two ways:
    a. Direct Entry Method – Users can type in the exact coordinates into the simulated trajectory table.
    b. The Nudge Tool – Users can open the Nudge Tool, select the desired simulated robot point to move, and then reposition the point by clicking the Nudge Tool buttons. See Figure 12-43.

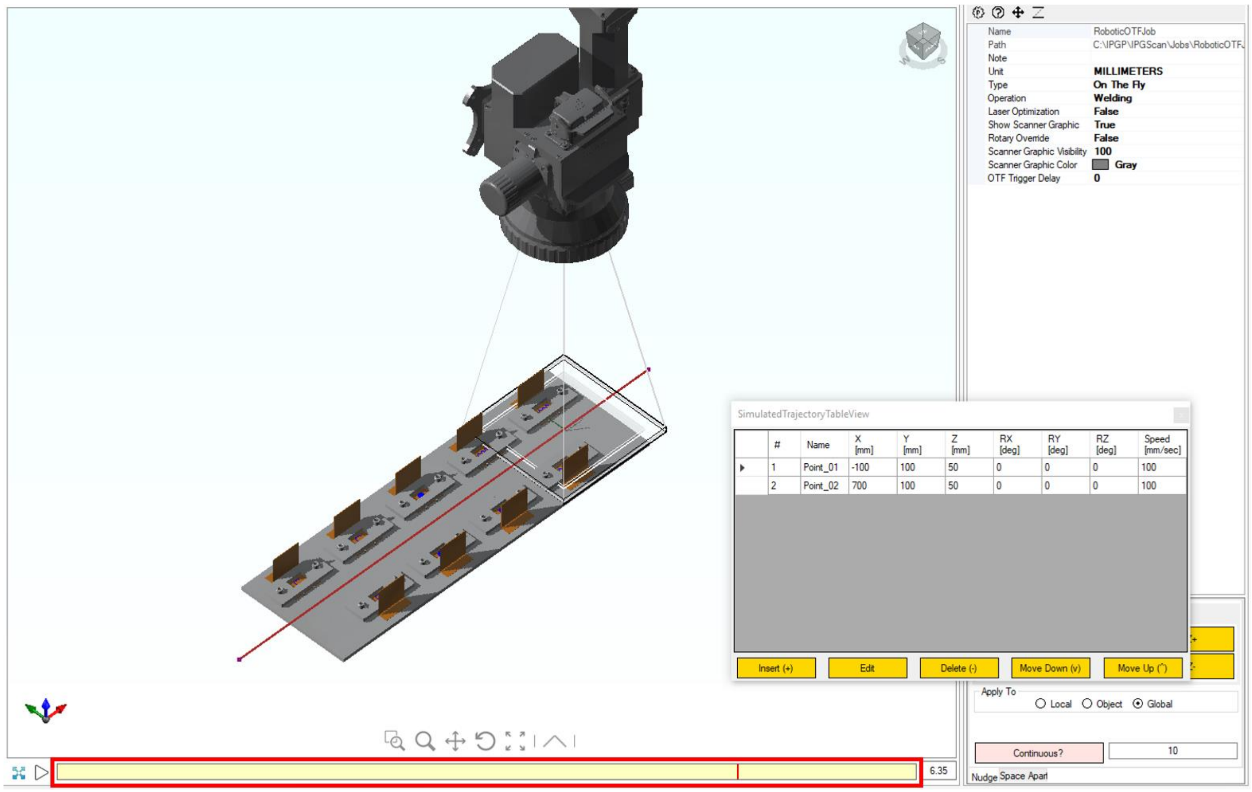
*Figure 12-43 Using the Nudge Tool to Position Simulated Trajectory Points*



7. When moving the simulated trajectory points into position, users can scrub along the timeline (below the IPGScan canvas) to visualize the head orientation along the simulated trajectory path. See Figure 12-44.

**IMPORTANT** **Simulated trajectories are created with fine termination types and instant acceleration.**

*Figure 12-44 Scrubbing Along the IPGScan Canvas Timeline*

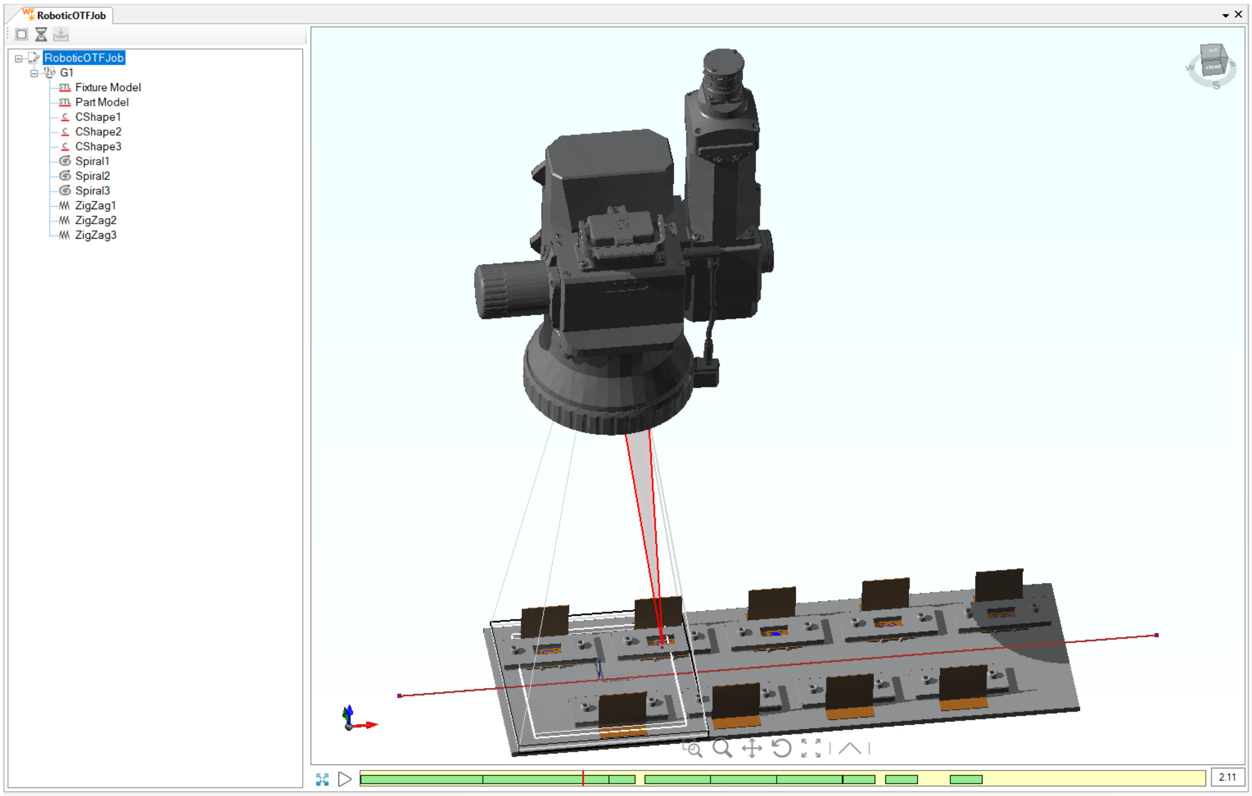



8. Once users are happy with the position of the simulated trajectory points, the Simulated Trajectory Table View can be closed by clicking on the close button in the upper right hand corner of the window.
   a. Simulated trajectory points remain in the table view even after it is closed. Also, simulated trajectory points are saved with the IPGScan job file.
9. Users can now Dryrun the job to ensure that processing can complete. See section “The Process of Dryrun and Timeline Review” for information on how to perform a Dryrun and examine process timings.

**IMPORTANT** **The scanner Field Of View configuration is also crucial to passing a Dryrun and creating a process. Please refer to section “Adjusting Canvas Settings (Scanner Field-of-View and In-View Window),” for details concerning scanner FOV setup.**

10. Upon successful completion of a Dryrun, users can scrub along the canvas timeline to view the beam trajectory for processing. See Figure 12-45.

*Figure 12-45 Successful Dryrun and Beam Trajectory Viewing*

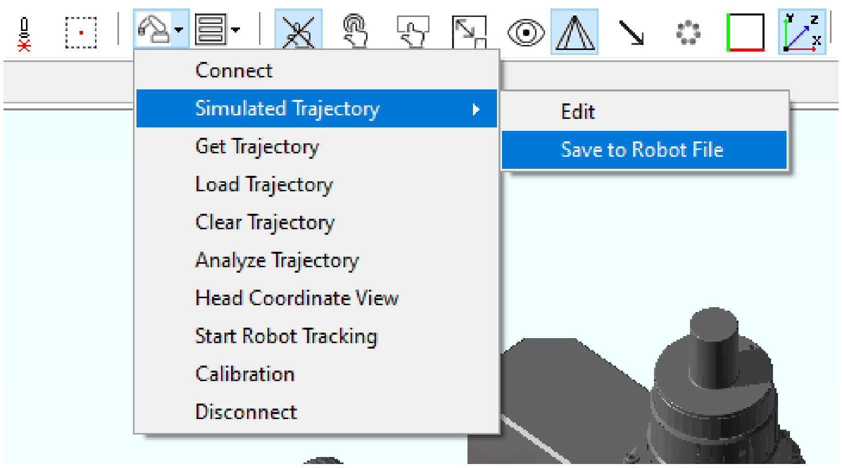


#### 12.4.3.2.1 Exporting a Simulated Trajectory

Once users have created a Simulated Trajectory, this path can be export for upload to a real robot. This provides users with a preliminary robot program and path that helps to reduce development time. The following procedure details how users can export a Simulated Trajectory to be loaded on a real robot.

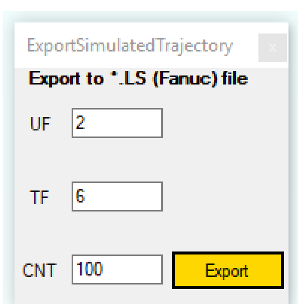
1. In the Tool bar, click on the robot icon, navigate to “Simulated Trajectory,” and click “Save to Robot File.” See Figure 12-46.

*Figure 12-46 Save to Robot File*




2. Enter the correct information for the creation of the robot program. See Figure 12-47.
    a. UF – User Frame
    b. TF – Tool Frame
    c. CNT – Motion Speed Percent (0-100)

*Figure 12-47 Export Simulated Trajectory Parameters*




3. Click “Export.”
4. Navigate to a location where the robot program can be saved and enter a file name. Click “Save.”
5. Click “Ok” to confirm that the exported robot file was saved.
6. The user can now navigate to and open the exported robot file. See Figure 12-48.

*Figure 12-48 IPGScan Simulated Trajectory Exported to a FANUC Robot File*

RoboticOTFJob.LS - Notepad

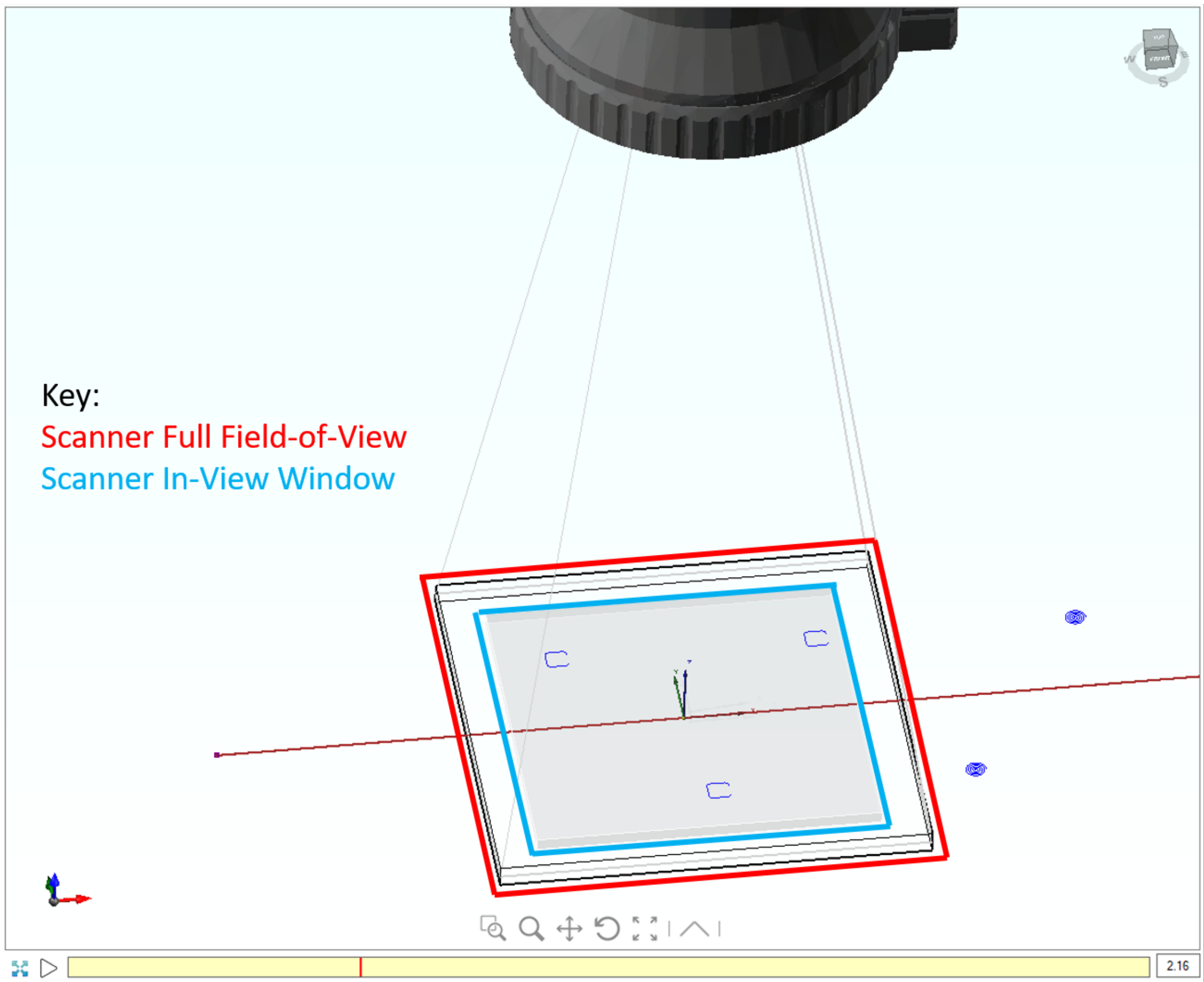
```
/PROG  ROBOTICOTFJOB
/ATTR
OWNER           = IPGSCAN;
COMMENT         = "";
PROG_SIZE       = 0;
CREATE          = DATE 22-03-21  TIME 10:00:25;
MODIFIED        = DATE 22-03-21  TIME 10:00:25;
FILE_NAME       = ;
VERSION         = 0;
LINE_COUNT      = 0;
MEMORY_SIZE     = 0;
PROTECT         = READ_WRITE;
TCD:  STACK_SIZE        = 0,
      TASK_PRIORITY     = 50,
      TIME_SLICE        = 0,
      BUSY_LAMP_OFF     = 0,
      ABORT_REQUEST     = 0,
      PAUSE_REQUEST     = 0;
DEFAULT_GROUP   = 1,*,*,*,*;
CONTROL_CODE    = 00000000 00000000;
/APPL
/MN
   1:  !THIS PROGRAM WAS AUTOMATICALLY ;
   2:  !GENERATED. BE SURE TO VALIDATE ;
   3:  !ALL MOTIONS AT REDUCED SPEED. ;
   4:  !ALWAYS FOLLOW PROPER SAFETY ;
   5:  !PRECAUTIONS. ;
   6:   ;
   7:  UTOOL_NUM=6 ;
   8:  UFRAME_NUM=2 ;
   9:L P[1] 100mm/sec CNT100 DO[11]=ON ;
  10:L P[2] 100mm/sec CNT100 DO[11]=OFF ;
/POS
P[1]{
   GP1:
        UF : 2, UT : 6,         CONFIG : 'F U T, 0, 0, 0',
        X =  -100.000  mm,      Y =   100.000  mm,      Z =    50.000  mm,
        W =       .000 deg,     P =       .000 deg,     R =      .000 deg
};
P[2]{
   GP1:
        UF : 2, UT : 6,         CONFIG : 'F U T, 0, 0, 0',
        X =   700.000  mm,      Y =   100.000  mm,      Z =    50.000  mm,
        W =       .000 deg,     P =       .000 deg,     R =      .000 deg
};
/END
```

Windows (CRLF) Ln 1, Col 1 100%

### 12.4.4 Adjusting Canvas Settings (Scanner Field-of-View and In-View Window)

For Robotic OTF applications, careful consideration of the scanners Field of View (FOV) is required for process development. Different from Point and Shoot style processes, users have an additional Field of View parameter called the "In-View Window." The In-View Window is what allows users to restrict the area within the scanners full FOV for additional process control. Figure 12-49 Outlines the difference between the scanners full FOV and the In-View Window. By reducing the area for which the scanner is allowed to process within the full FOV, users gain additional control over process timings, depth of focus tolerance, and allowable beam approach angles to the work surface.

*Figure 12-49 Scanner Field-of-View and In-View Window*

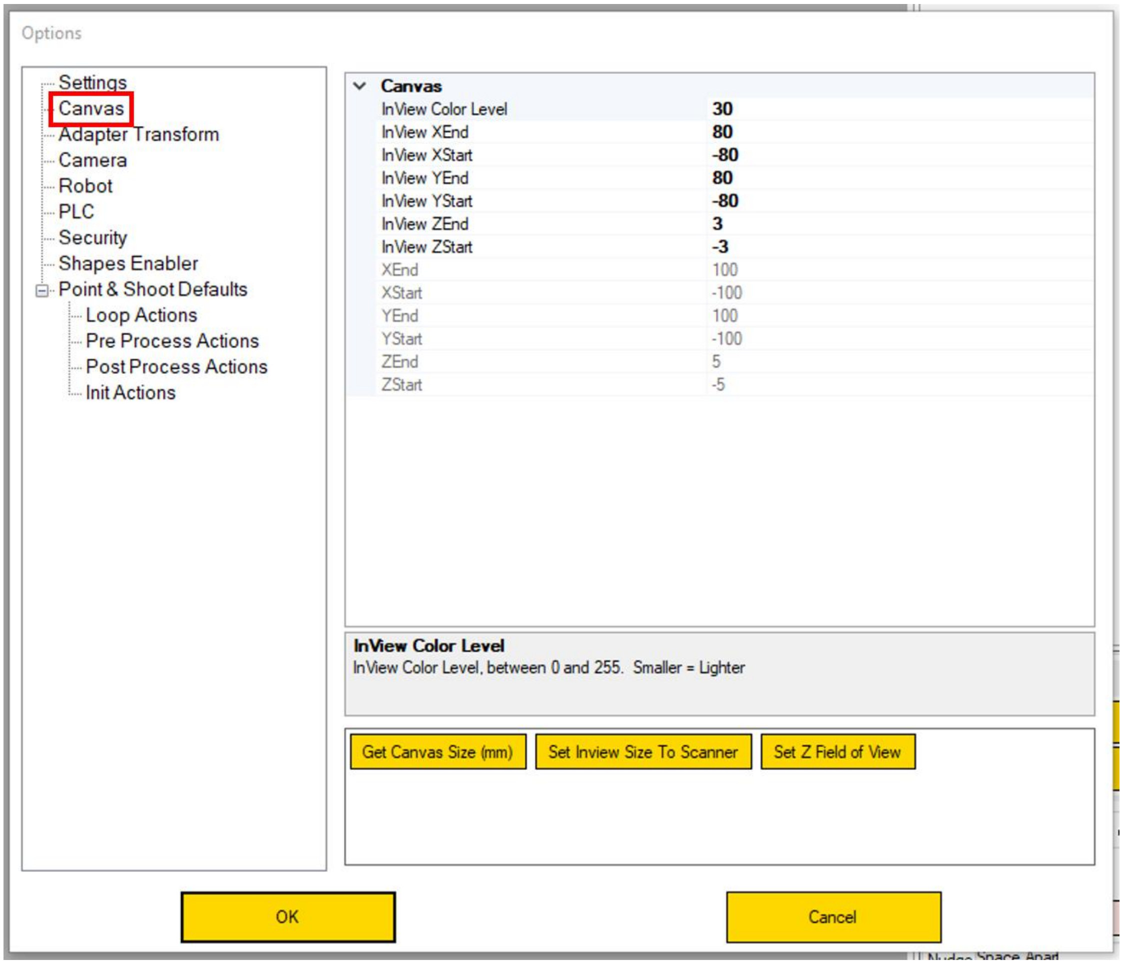



While users do not have the ability to adjust the X and Y parameters of the Scanners FOV, the Z parameter and In-View Window dimensions can all be adjusted based on the users process needs. The following details how users can navigate to the menu to adjust scanner FOV and In-View Window settings.

1. Open IPGScan and connect to a desired scanner.
2. Click "View."
3. Click "Options."

4. Select "Canvas." See Figure 12-50.
    a. From here, users can adjust the scanners FOV and In-View Window settings.

*Figure 12-50 IPGScan Canvas Settings*



### 12.4.5 The Process of Dryrun and Timeline Review

To visualize a Robotic OTF process in IPGScan, users perform a Dryrun once process objects have been placed in the Canvas and a trajectory (real or simulated) is added to the job. The purpose of the Dryrun feature is for IPGScan to analyze if the scanner can adequately process all objects in the job based on process object order, relative position, and process timings. If a job successfully passes Dryrun, users can expect to run the real process and observe the desired output. If the Dryrun process fails, users should investigate the cause of the failure prior to attempting to run the real process.

| | |
|---|---|
| **IMPORTANT** | **If a job passes Dryrun but the real process output is not as expected (i.e. distorted welds or improperly oriented), users should verify the IPGScan and Robot Workspace Calibration. Additionally, users should ensure that they have captured and imported the real robot trajectory.** |

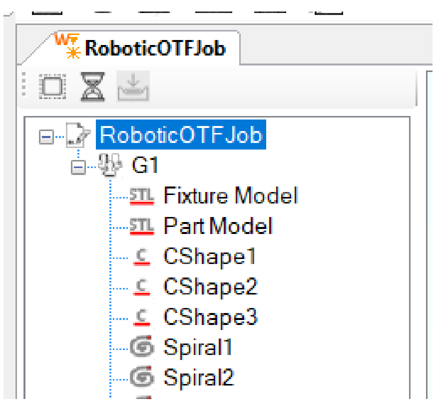
The following sections detail how users can perform a Dryrun as well as how process timings can be observed using the Basic and Advanced Timelines.

#### *12.4.5.1 Dryrun*

The following procedure outlines how users can perform a Dryrun.

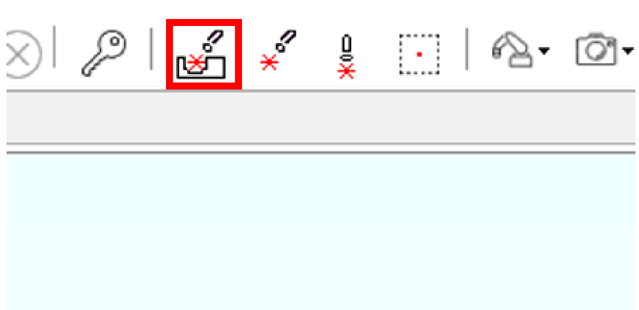
1. In the desired On-The-Fly job, select the name of the job in the Job Tree. See Figure 12-51.

*Figure 12-51 Selecting the Job Name for Dryrun*



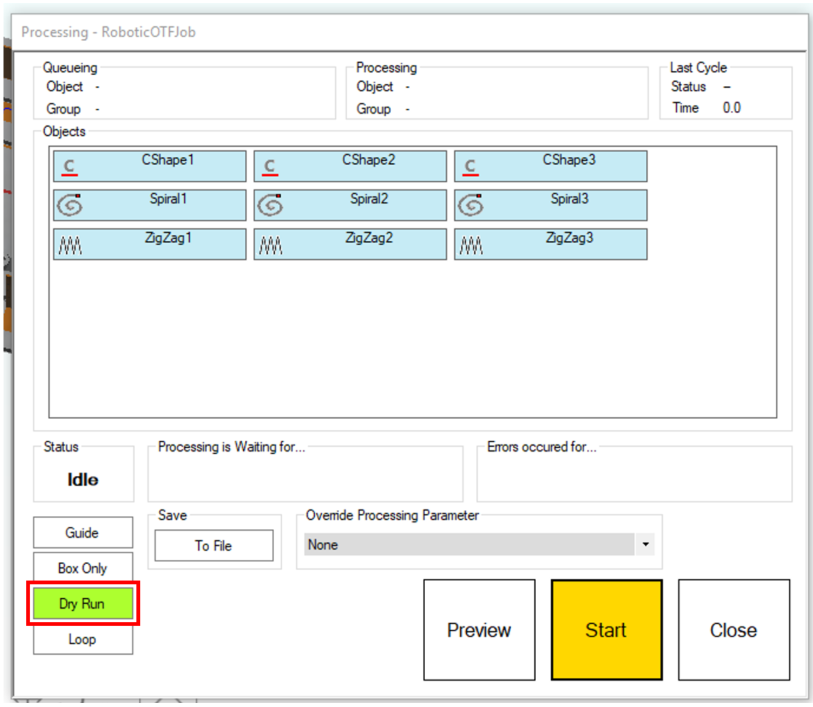

2. Click the “Start Processing” Button. See Figure 12-52.

*Figure 12-52 Start Processing*



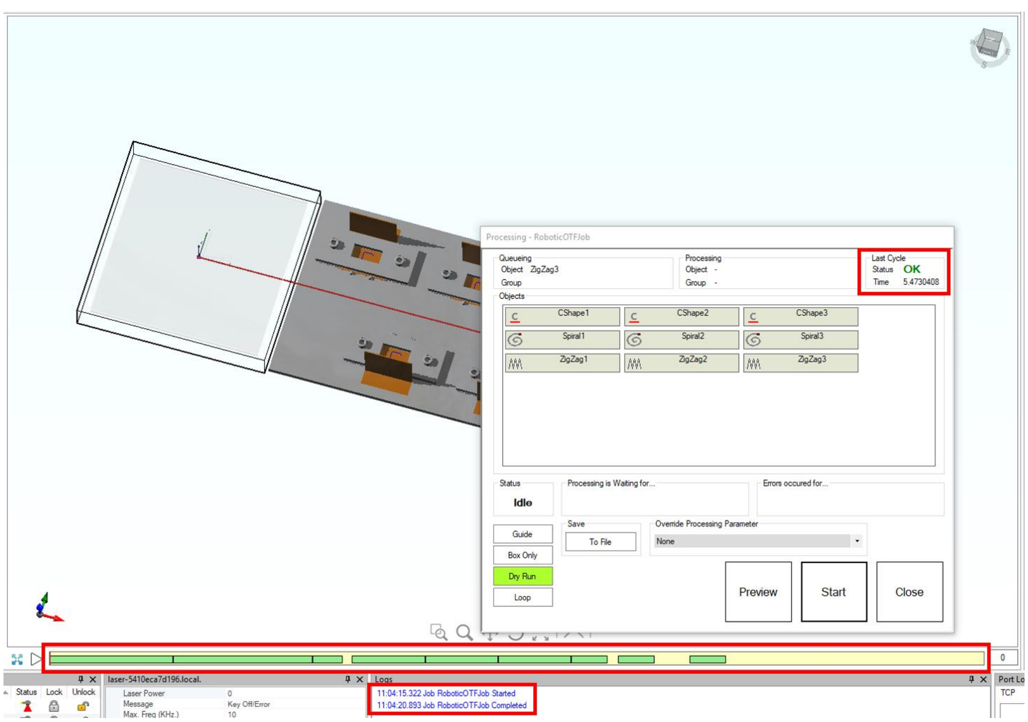
3. Select the “Dryrun” button. See Figure 12-53.

*Figure 12-53 Enabling Dryrun*



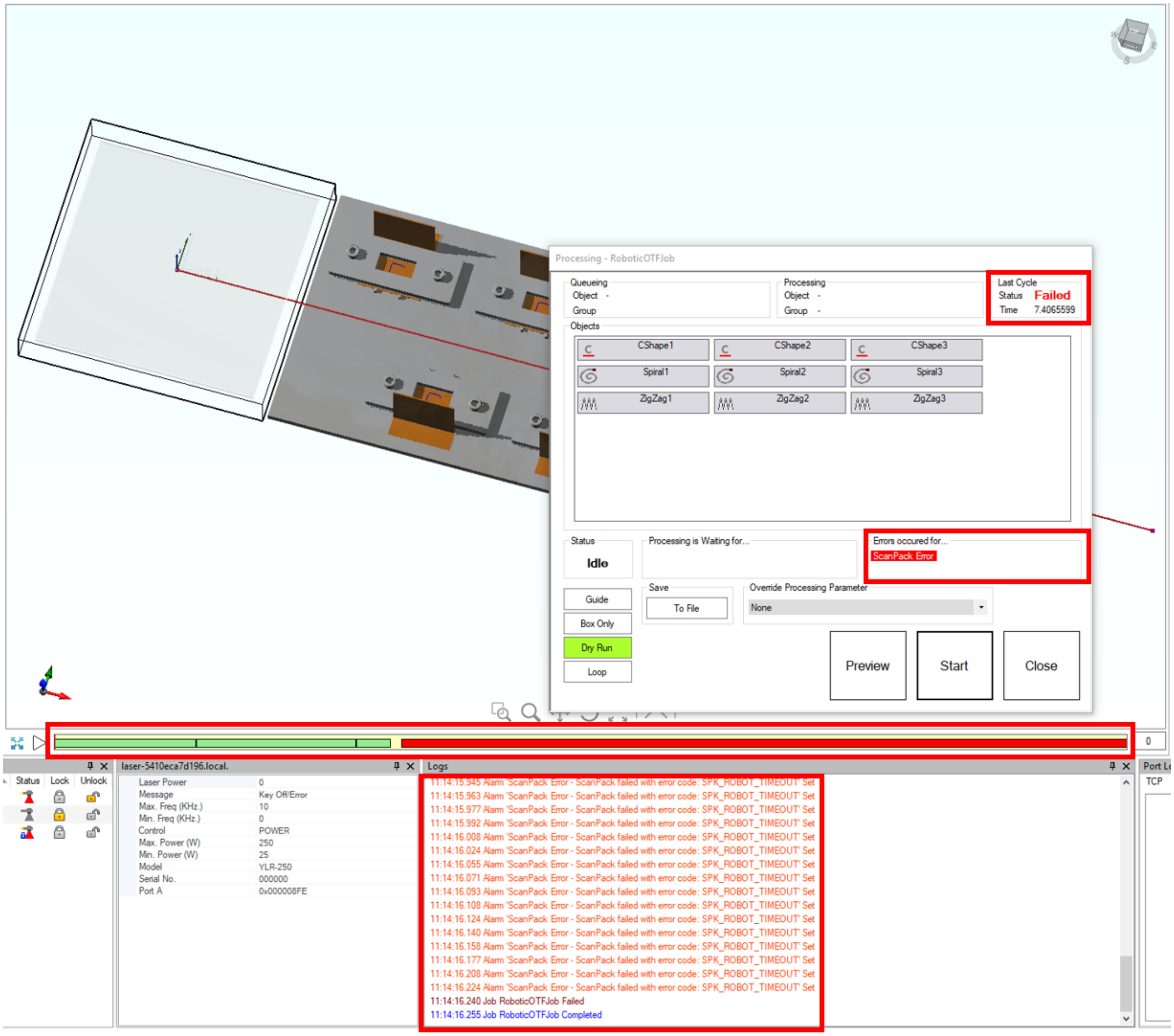

4. Click “Start.”
    a. Figure 12-54 demonstrates a successful Dryrun.

*Figure 12-54 Successful Dryrun Example*



b. Figure 12-55 demonstrates an unsuccessful Dryrun.

*Figure 12-55 Unsuccessful Dryrun Example*

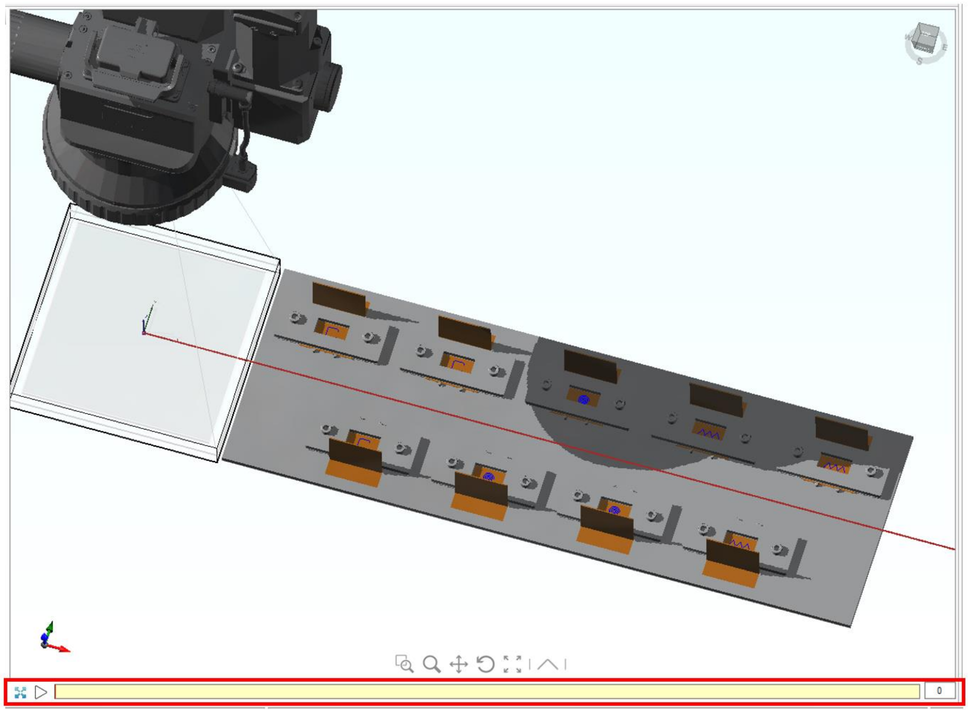



Having the ability to Dryrun an OTF process allows users to examine the process prior to attempting to run it on the real system. Using both the Basic and Advanced Timelines, users can begin to analyze and optimize the process as required.

### 12.4.5.2 Basic Timeline

The purpose of the Basic Timeline is to provide users with a quick summary of process timings. It also serves as an indicator as to whether or not a Dryrun has been performed and whether or not the Dryrun passed or failed. The Basic Timeline is located directly below the IPGScan Canvas. See Figure 12-56.

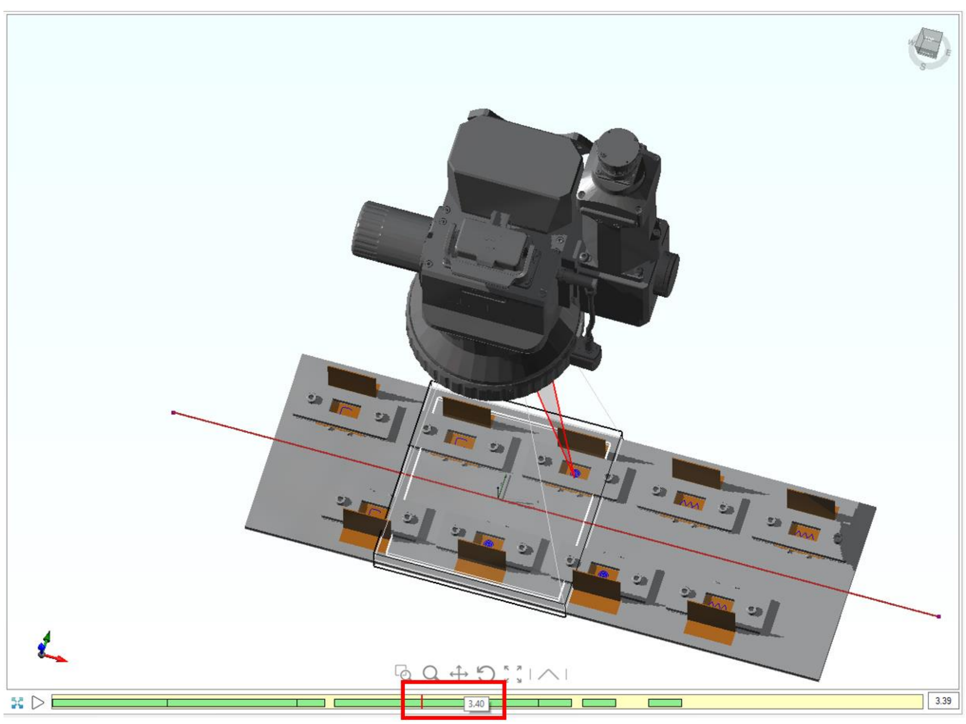
*Figure 12-56 The Basic Timeline*



The Basic Timeline offers the following functionality.

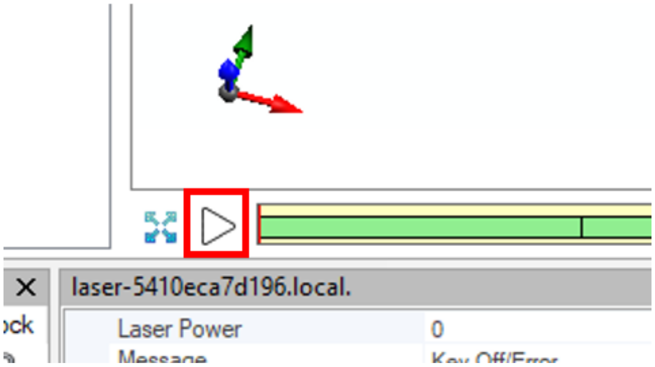
- Timeline Scrubbing – Users can click on the timeline and drag the mouse cursor to scrub through the timeline at a desired rate. See Figure 12-57.

*Figure 12-57 Scrubbing Along the Basic Timeline*



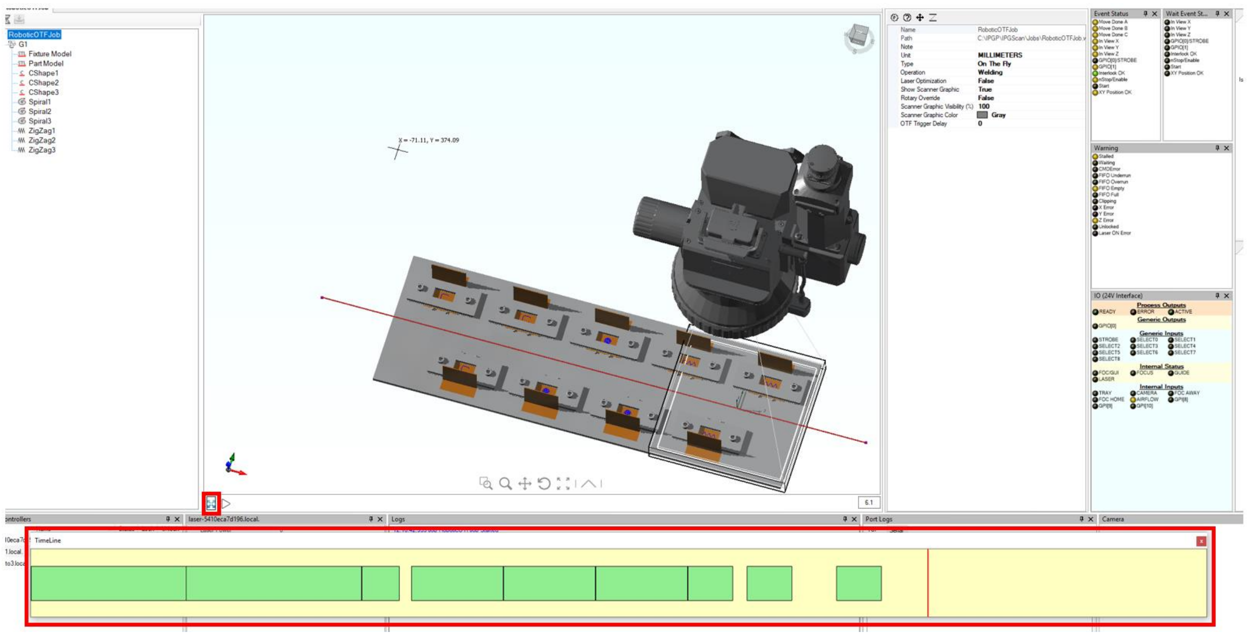
- Play Button – Starts a loop of the process visualization. See Figure 12-58.

*Figure 12-58 Basic Timeline Play Button*



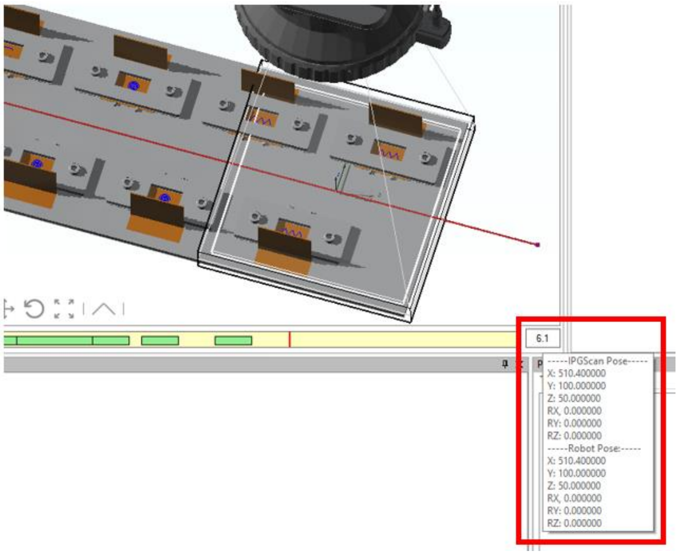

- Expansion Button – Allows users to break-out and expand the Basic Timeline. See Figure 12-59.

*Figure 12-59 Basic Timeline Expansion*



- IPGScan and Robot Pose Data – Users can view the current IPGScan Pose and Robot Pose data of where the FOV lies along the trajectory by hovering the mouse cursor over the time display. See Figure 12-60.

*Figure 12-60 Basic Timeline IPGScan and Robot Pose Display*



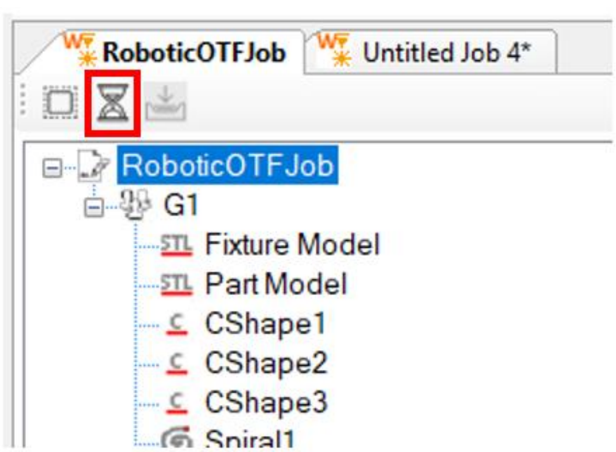

### 12.4.5.3 Advanced Timeline

The purpose of the Advanced Timeline is to provide users with a detailed summary of process timings. This includes details such as when an object is partially in the scanners FOV, when it is completely within the FOV, and when processing is taking place. Furthermore, the Advanced Timeline allows users to quickly change primary process parameters such as process object velocity, laser power, and delay timings. This can assist with quickly developing a process.

The following steps detail how users can access the Advanced Timeline.

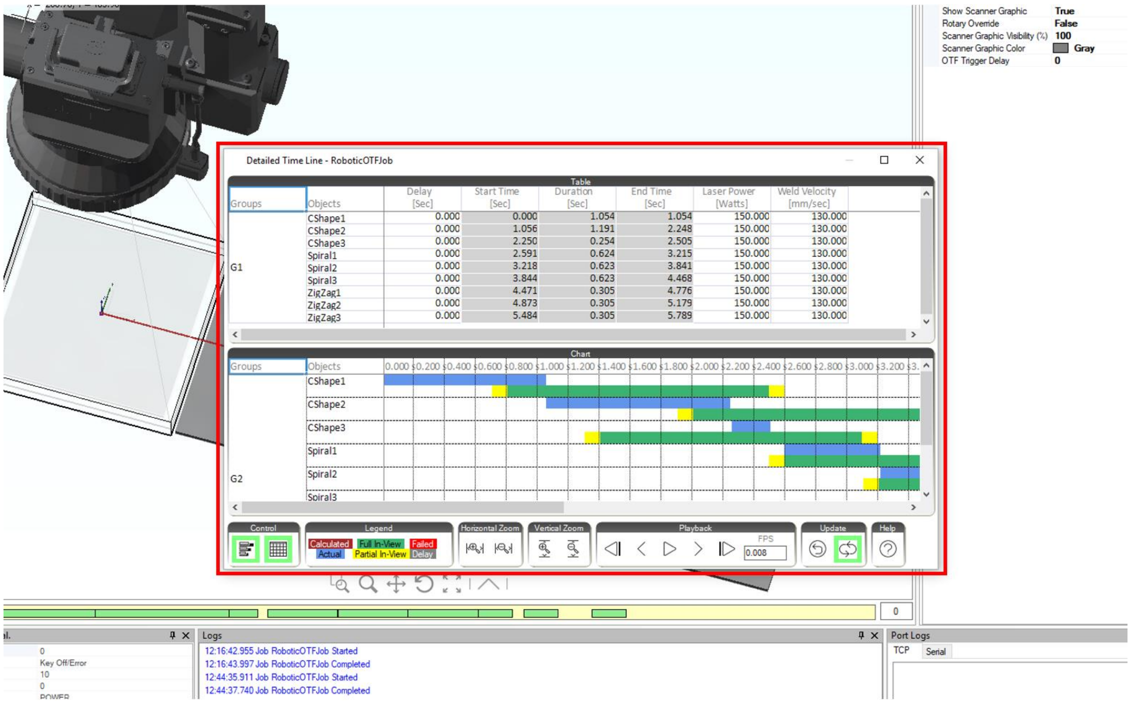
1. Open the desired OTF IPGScan job.
2. Click the "Time Line" button at the Top of the Job Tree. See Figure 12-61.

*Figure 12-61 Advanced Timeline Button*




The Advanced Timeline appears as an individual pop-up window which can be seen in Figure 12-62.

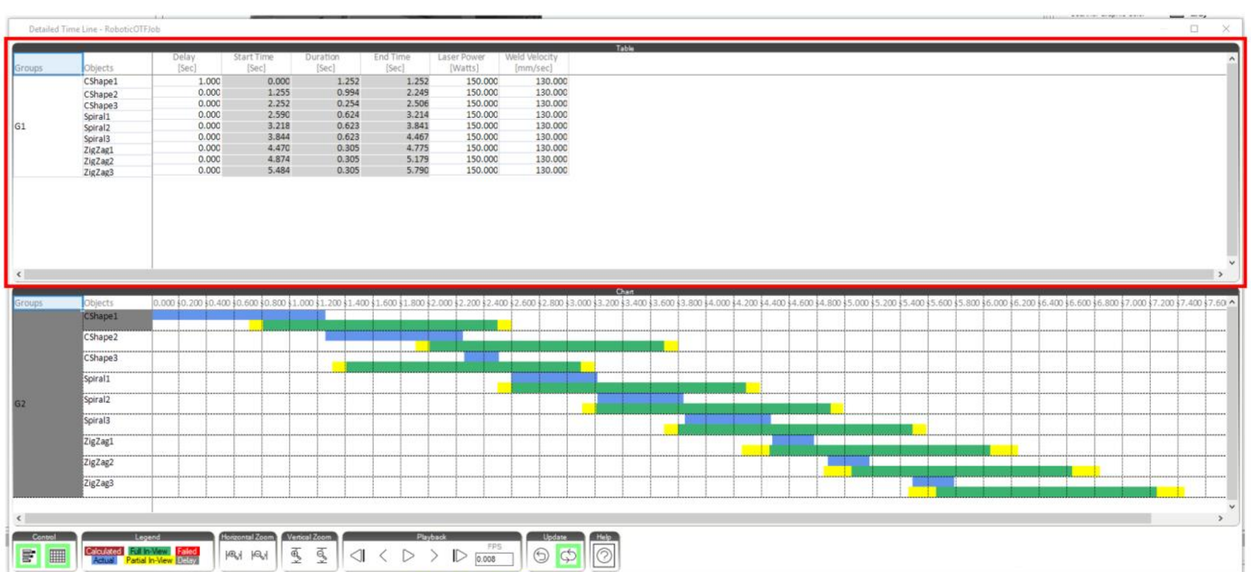
*Figure 12-62 IPGScan Advanced Timeline*




The Advanced Timeline offers the following functionality.

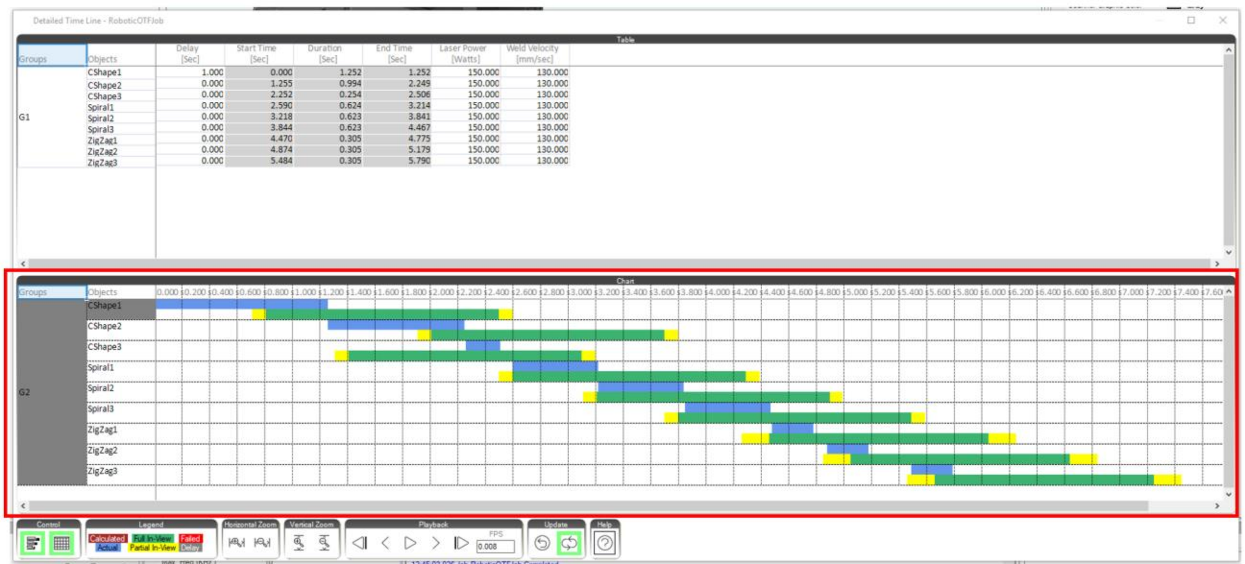
- Object Parameter Table – This outlines all the Process Objects within the job. Here users can find and modify primary parameters such as Laser Power and Weld Velocity. Users can also view Start, End, and Duration times. Additionally, users can easily modify delay timings for prior to object processing. See Figure 12-63.

*Figure 12-63 Object Parameter Table*



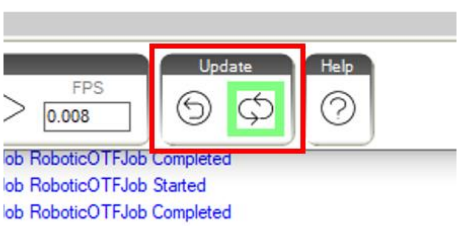
- Timeline Graph – This outlines detailed process object timings. This allows users to efficiently adjust parameters to develop the process. See Figure 12-64.
    - Legend – The Legend, which is positioned below the Timeline Graph, contains the color codes for the graph.
    - Horizontal and Vertical Zoom – Also positioned below the Timeline Graph, here users can use the Horizontal and Vertical Zoom tools to better examine process timings.

*Figure 12-64 Timeline Graph*



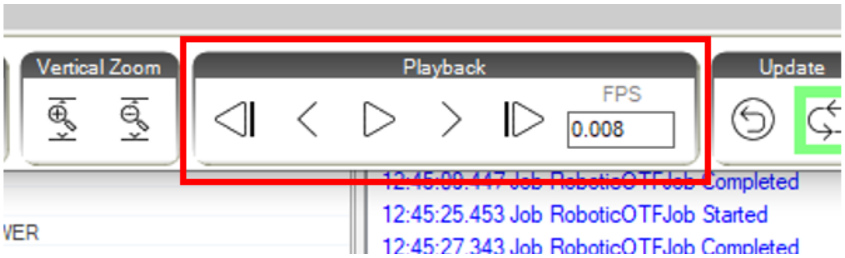
- Update Buttons – Two buttons exist for updating the Timeline Graph. By default, Automatic update is enabled. This function causes the Timeline Graph to update automatically anytime a Process Object's parameters are updated. A single instance update button also exists, if users choose to disable the automatic update functionality. See Figure 12-65.

*Figure 12-65 Timeline Update Buttons*




- Playback – The Playback buttons allow users to start a process play through or step through the process visualization. The speed for which the play through occurs is based on the defined FPS value. See Figure 12-66.

*Figure 12-66 Timeline Playback Buttons*




### 12.4.5.4 Common Process Failures

When developing a process, users will typically need to perform numerous Dryruns to view how the process is changing based on setup details. During this process, it is not uncommon to encounter process errors until the process is dialed in.

The following outlines some of the common errors that can be encountered in the Logs window during the Dryrun process as users develop the OTF process.

- “Robot Trajectory is Missing!” – Users need to upload a simulated trajectory or real robot trajectory to the job prior to attempting to Dryrun or start processing.
- “SPK_ROBOT_TIMEOUT” – This typically indicates that there was not enough time for the scanner to complete the process.
  - Potential Resolutions:
    - Job Tree Process Object Order – Ensure that any Process Objects prior to the object that failed were able to successfully pass the Dryrun.
    - Position – Ensure that the Process Object that fails falls completely within the scanner’s In-View window.
    - Expand the In-View Window – If the user does not have beam angle restrictions and if the In-View Window is smaller than the default scanner FOV, users can increase the size of the In-View Window to allow more time for the objects to be processed.
    - Robot or Process Object Velocity – Try to reduce the robot trajectory (simulated or real) velocity or the Process Object velocity. By reducing either of these velocities, the process should have more time to complete.
- “SPK_INVALID_INVIEW” – This typically indicates that users have not uploaded the CoordinationParams.xml file using the Scan Controller Utility to the scanner.
  - See section “Scan Controller Requirements” for details on uploading the CoordinationParams file.

### 12.4.6 Connecting to a Robot

Once users are ready to move to the real world setup, it is necessary to establish a connection with the robot. The following items entail the functionality within IPGScan that requires a connection with the robot.

1. Performing the IPGScan and Robot Calibration
2. Capturing a robot trajectory to be uploaded to a OTF job
3. Robot Tracking functionality
4. Part Alignment for STL functionality
5. Alignment Window functionality

**IMPORTANT** **A connection with the robot is not a requirement for running production repeatedly. The connection is only required during the setup of the process.**

The following sections outline the options for robot setup within IPGScan as well as how to connect to a robot.

#### *12.4.6.1 IPGScan Robot Options*

Prior to connecting to a robot, users must first setup the Robot Options within IPGScan. The following steps outline where users can find the Robot Options:

1. Open IPGScan.
2. Click "View."
3. Click "Options."
4. Select "Robot." See Figure 12-67.

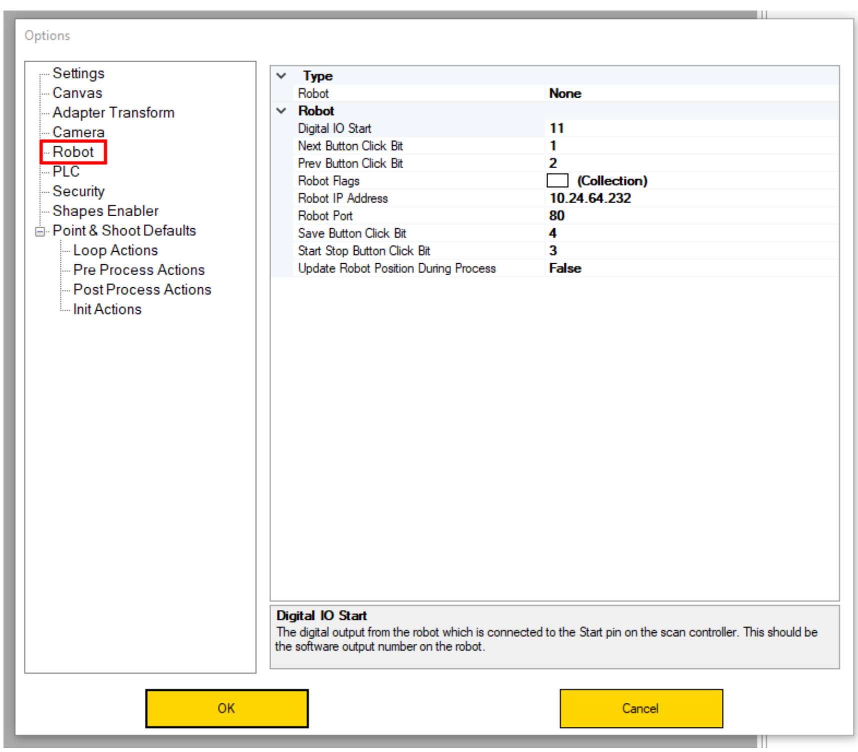



*Figure 12-67 IPGScan Robot Options*

Table 12-3 details each of the available options for setup of the robot.

*Table 12-3 Robot Option Settings*

| Option | Description |
|---|---|
| Robot | The type of robot. To turn off robot related functions, set this to 'None.' |
| Digital IO Start | The digital output from the robot, connected to the START bit on the Scan Controller (via an External Interface device). This should be the software output number on the robot. |
| Next Button Click Bit | The output from the robot which is read by IPGScan to move to the next Process Object during alignment. This should be the bit number on the Scan Controller. *Optional functionality.* |
| Prev Button Click Bit | The output from the robot which is read by IPGScan to move to the previous Process Object during alignment. This should be the bit number on the Scan Controller. *Optional functionality.* |
| Robot Flags | Which robot output values should be recorded and used as coordination flags in a trajectory. These do not have to be connected to the Scan Controller. *Optional functionality.* |
| Robot IP Address | The IP Address of the robot. |
| Robot Port | The communication port to the robot. Not required for all robot types. |
| Save Button Click Bit | The output from the robot which is read by IPGScan to save the current position to the current processing object during alignment. This should be the bit number on the Scan Controller. *Optional functionality.* |
| Start Stop Button Click Bit | The output from the robot which is read by IPGScan to start or stop outputting the current Process Object during alignment. This should be the bit number on the Scan Controller. *Optional functionality.* |
| Update Robot Position During Processing | Enables/disables the FOV display traversing along the trajectory when processing is taking place. By default, this feature is disabled to reduce computer CPU load. |

**IMPORTANT** **Upon updating Robot Options, users should close and reopen IPGScan.**

### 12.4.6.2 Robot Connection and Status

Once robot setup is complete and the proper Robot Options are configured, the user can connect to the robot. The following details how users can connect to the robot in IPGScan.

1. In the Tool Bar, click the robot icon.
2. Click "Connect." See Figure 12-68.

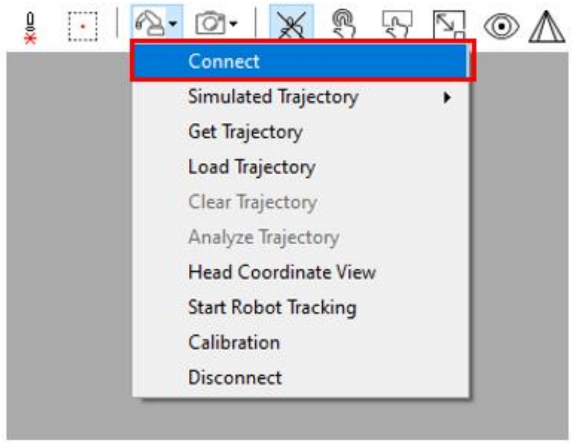
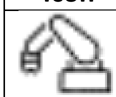
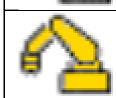



*Figure 12-68 Connecting to a Robot*

When connected to a robot, users may notice that the robot icon changes colors in the IPGScan Tool Bar. Outlines the different statuses based on each color.

| Icon | Description |
|---|---|
|  | Disconnected |
|  | Connected with the robot but no data is being received |
|  | Connected with the robot and data is being received |

Connecting to a robot in IPGScan varies from one brand of robot to another. The following details how to connect to each specific type of robot.

**IMPORTANT** **[IPGScan] – Indicates an action taken via the IPGScan interface.**
**[Robot] – Indicates an action taken via the robots interface.**

### 12.4.6.2.1 FANUC

**Robot Server**

1. Click “Robot” in the Tool Bar [IPGScan]
2. Click “Connect” [IPGScan]

**KAREL**

1. Run the program “IPG_COMM” from the SELECT Menu [Robot]
2. Click “Robot” in the Tool Bar [IPGScan]
3. Click “Connect” [IPGScan]
    a. Note: If the connection fails, try running the program “IPG_ABORT” before running “IPG_COMM” [Robot]

##### 12.4.6.2.2 Yaskawa Motoman

1. Click "Robot" in the Tool Bar [IPGScan]
2. Click "Connect" [IPGScan]

##### 12.4.6.2.3 KUKA

1. Click "Robot" in the Tool Bar [IPGScan]
2. Click "Connect" [IPGScan]
    a. When "Connect" is clicked within IPGScan and configured for a KUKA robot, the icon will turn yellow. This tells IPGScan to listen for data from the KUKA robot. When IPGScan receives data, the icon will turn blue. When IPGScan stops receiving data, the icon will turn yellow. The icon will update while other operations are executing within IPGScan such as performing the IPGScan and Robot calibration and capturing a trajectory.
3. Run IPG_OTF_BEGIN_DATA_XFER [Robot]
    a. This sub module initializes and begins RSI. When you run this program, you will get a KUKA error RSI 999. In the event of this error, stop running the KUKA program, acknowledge the error, and resume running the program.

- KUKA Connection Notes
    - Both IPG_OTF_BEGIN_DATA_XFER and IPG_OTF_END_DATA_XFER require a Boolean value as a parameter. RSI will only be started when the parameter is true. During teaching either the robot trajectory or process positions in IPGScan, pass the value true to send data from the robot. During execution and production, pass the value false to avoid the KUKA error RSI 999 and 2 second wait.
- Stopping Data Transmission
    - To stop sending data from the KUKA robot, run IPG_OTF_END_DATA_XFER. This program contains a 2 second wait. This gives the robot enough extra time to finish sending data when recording a trajectory. RSI will also stop when the current KUKA program ends.

##### 12.4.6.2.4 ABB

1. Click "Robot" in the Tool Bar [IPGScan]
2. Click "Connect" [IPGScan]

### 12.4.7 IPGScan and Robot Workspace Calibration

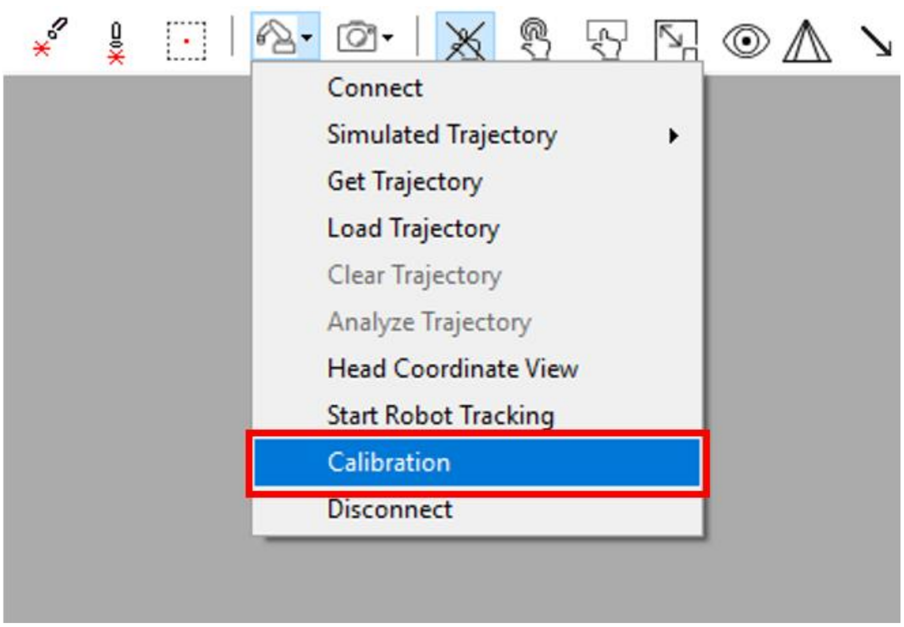
The calibration process creates an origin for IPGScan OTF jobs. The robot trajectory and all processing objects are in reference to this IPGScan origin.

| | |
|---|---|
| **IMPORTANT** | **An accurate Tool Center Point (TCP) is required for Calibration.**<br><br>**The robot TCP axis should be aligned so they are square to the physical scan head.**<br><br>**Users must be able to connect to the robot in IPGScan prior to performing the calibration.**<br><br>**To best utilize exporting simulated robot trajectories, it is recommended that users align the Robot user frame with the IPGScan world. In other words, utilize the same robot points used to teach the robot user frame for the calibration between IPGScan and the robot.** |

The following procedures outlines how to perform the IPGScan and Robot Calibration.

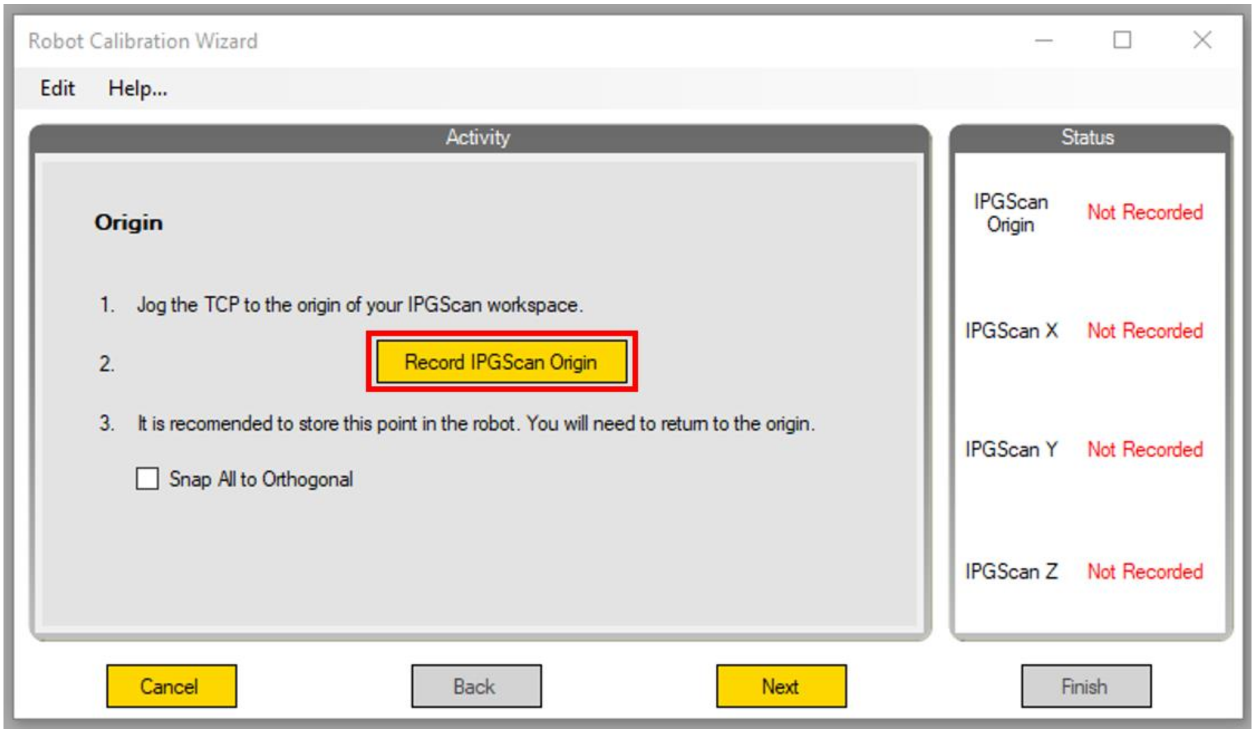
1. Click on the robot icon in the Tool Bar.
2. Click "Calibration." See Figure 12-69.

*Figure 12-69 Opening the Calibration Wizard*



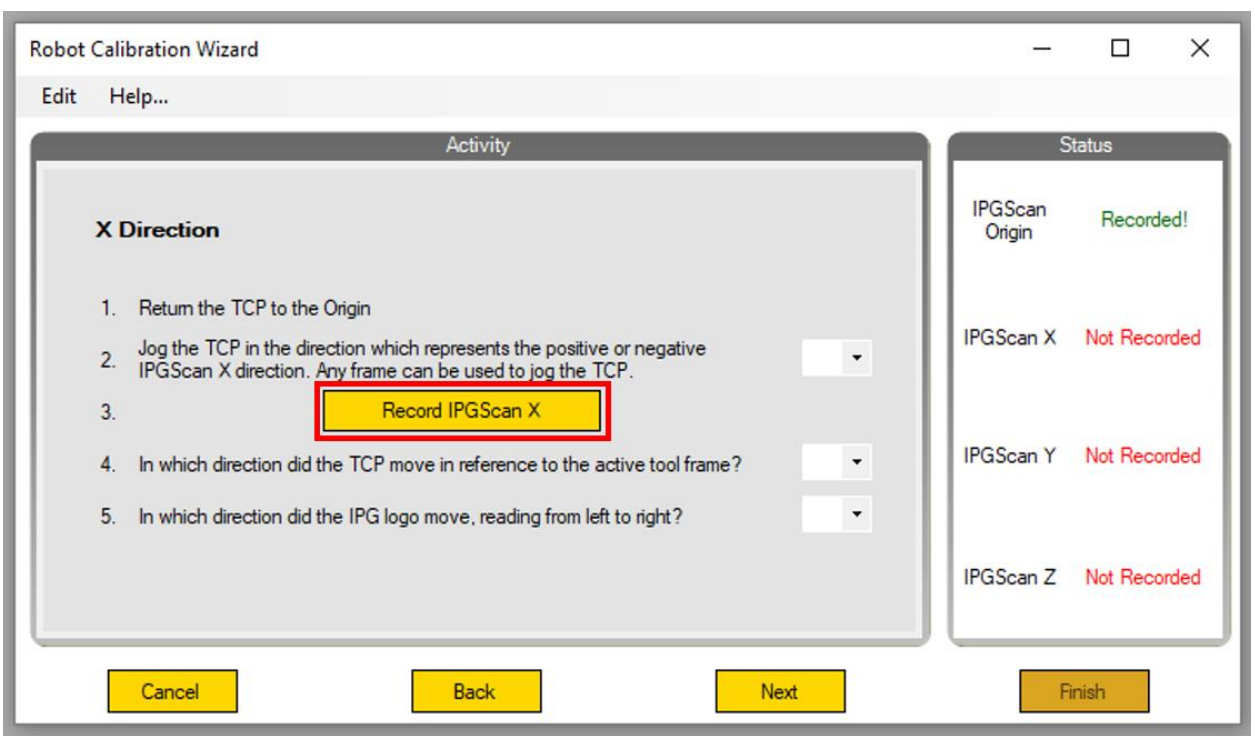

3. On the first page, the origin is recorded (see Figure 12-70). The origin can be anywhere within the robot's work area. When the robot is positioned, click the "Record IPGScan Origin" button. On the right window, it should now display that the IPGScan Origin is recorded.
    a. Snap All to Orthogonal – Checking this box will snap all angles to the nearest multiple of 90˚, which helps to reduce minor errors in positional feedback. *Optional*.

*Figure 12-70 Calibration Wizard Origin*



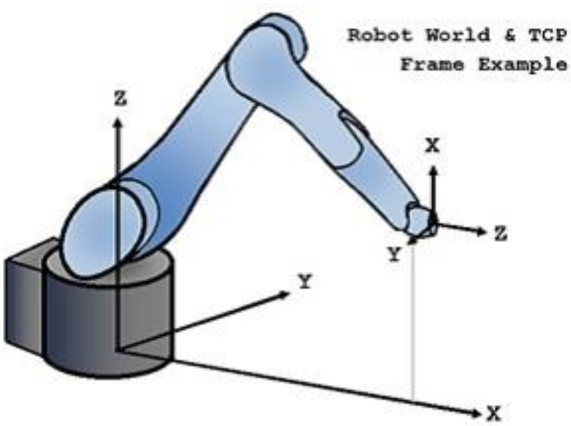

4. Click “Next.”
5. On each of the following screens, the IPGScan X, Y, and Z references are set. The robot must start at the origin before recording the new position on each page (see Figure 12-71).

*Figure 12-71 Calibration Wizard X, Y, Z Axis*



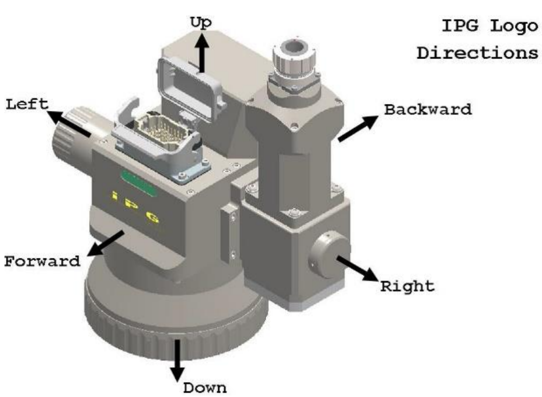
6. First, return the robot to the origin.
7. The second step requires the robot to be moved in only the X, Y, or Z direction, with respect to the desired IPGScan reference frame. For instance, if users are aligning the IPGScan world to the robot user frame, they should jog about one of the robot user frame axis to define the desired frame direction in IPGScan.
8. When the robot is positioned, click the “Record IPGScan …” button to record that axis.
9. Set the direction for which the robot was moved with respect to the robots TCP frame. Figure 12-72 provides an example of how the robot TCP frames may be aligned.

*Figure 12-72 Robot World & TCP Example*




10. Set the direction for which the IPG Logo (located on the front of the scan head) moved. See Figure 12-73.

*Figure 12-73 IPG Scan Head Logo Directions*




11. When all points have been recorded, click “Finish.”

### 12.4.7.1 Robot Manufacturer Specific Calibration Notes

The following sections detail specific notes to be aware of when calibrating specific brands of robots.

#### 12.4.7.1.1 KUKA

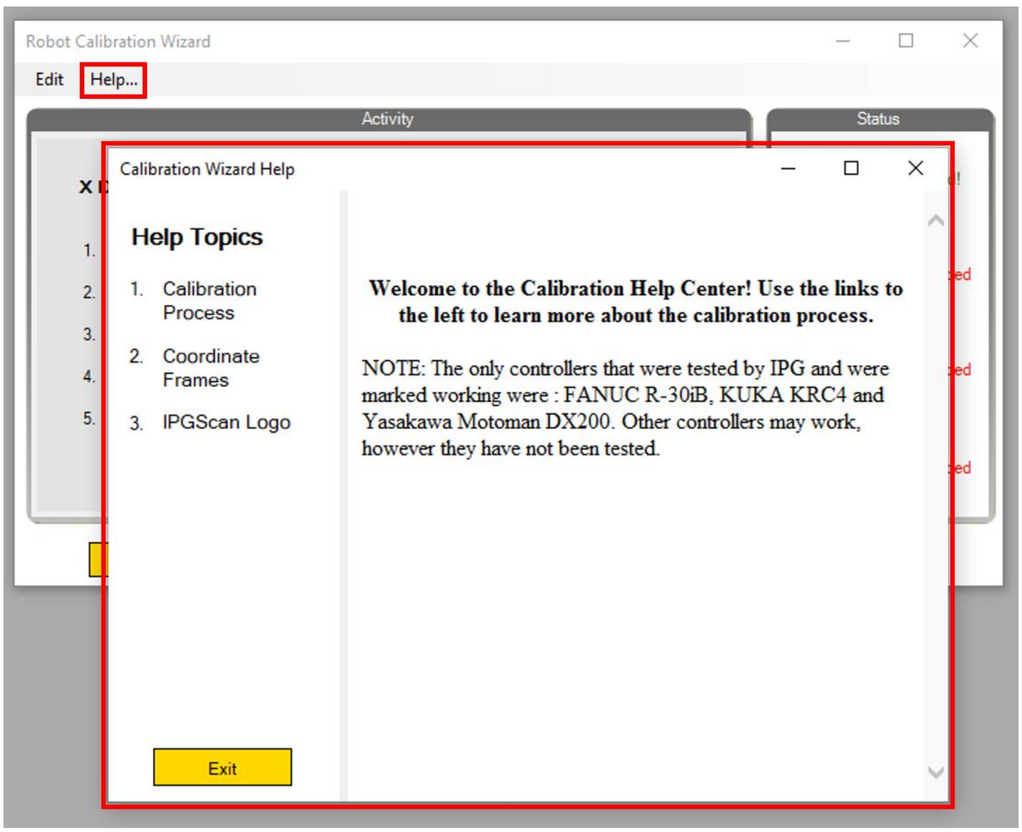
When calibrating IPGScan with a KUKA robot, RSI must be started before the robot can send information to the PC. A suggested method for IPGScan workspace calibration can be found in the robot module "IPG_OTF_CALIBRATION." The point P1 should be saved as the IPGScan workspace origin.

The module "IPG_OTF_CALIBRATION" first starts RSI and then moves to the origin. There is a 10 second wait after the first PTP move. This wait proves the robot operator time to pause execution of the module. When execution of the module is paused, the robot can be jogged and RSI will continue to facilitate communication between the robot and the PC.

### *12.4.7.2 Calibration Wizard "Help" Pop-Up*

Within the calibration wizard, users can open a "Help" dialogue pop-up that provides additional details for the calibration process. See Figure 12-74.

*Figure 12-74 Calibration Wizard "Help" Pop-Up*

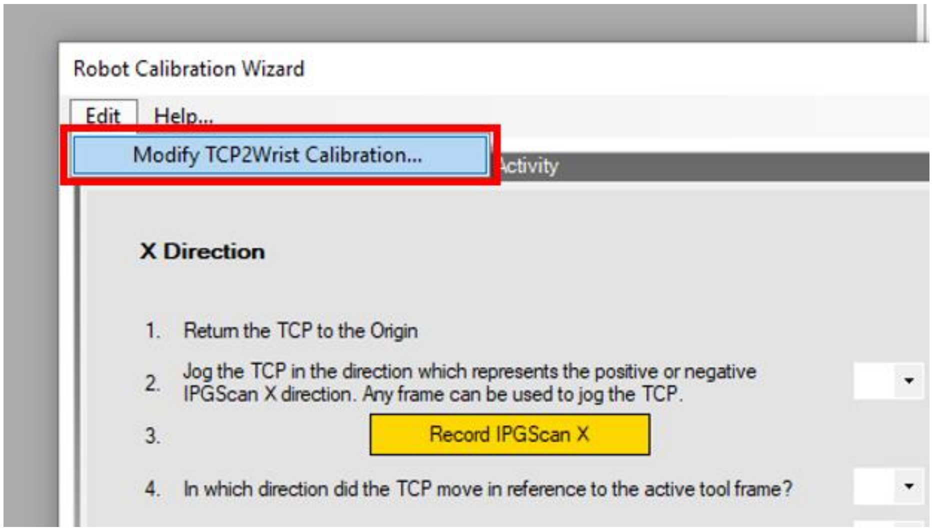



### *12.4.7.3 Calibration Wizard "Modify Calibration File" Pop-Up*

Users can open a wizard to modify an existing TCP2Wrist calibration file. This can be done using the following procedure from the Calibration Wizard.

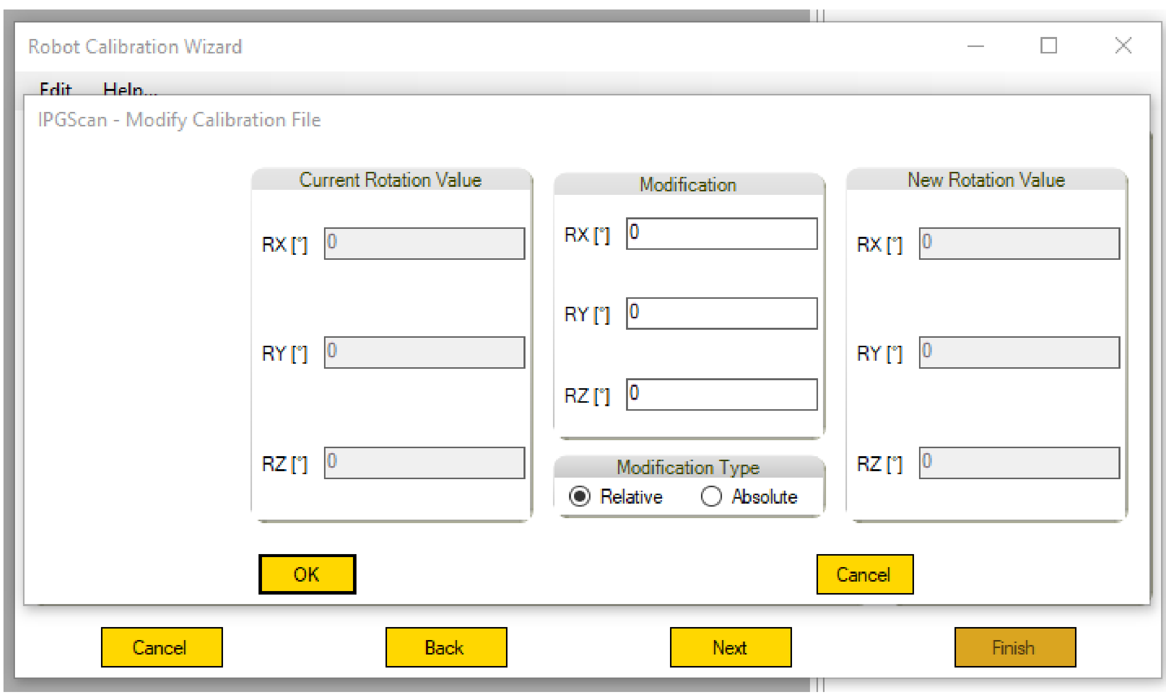
1. Click "Edit."
2. Click "Modify TCP2Wrist Calibration." See Figure 12-75.

*Figure 12-75 Opening the "Modify Calibration File" Window*




Within the Modify Calibration File pop-up, users can modify an existing TCP2Wrist calibration file by entering the desired offsets in the Modification input and by clicking ok. See Figure 12-76.

*Figure 12-76 Modify Calibration File Window*



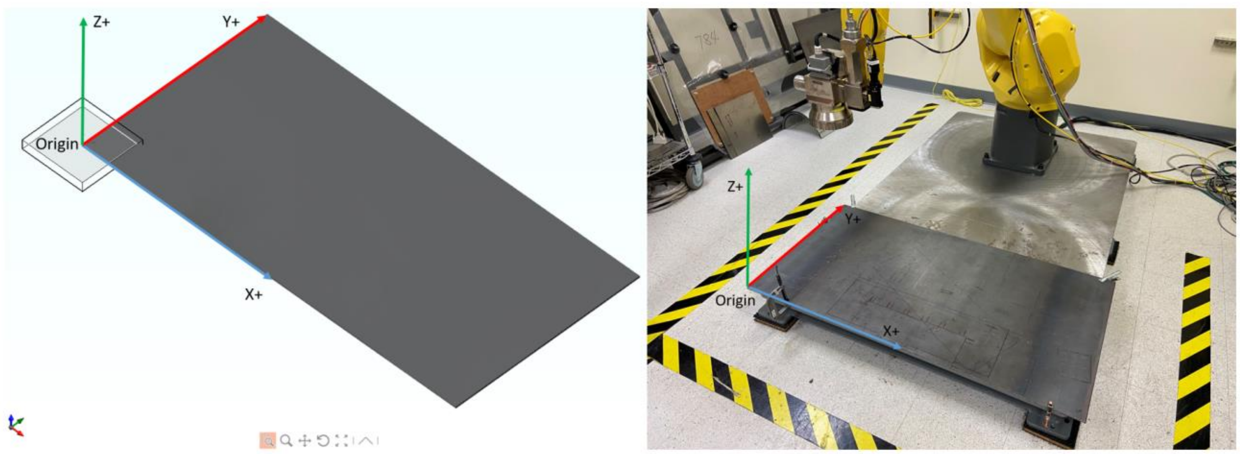
### 12.4.8 Part Alignment for STL

Once users have performed the IPGScan and Robot calibration, it is now possible to perform Part Alignment for STL. This functionality allows users to quickly snap CAD models, Process Objects, and simulated trajectories in the proper location in IPGScan that corresponds with the real world. In doing this, users are then able to align the software/offline world with the real world.

The following steps outline how to perform STL alignment in IPGScan.

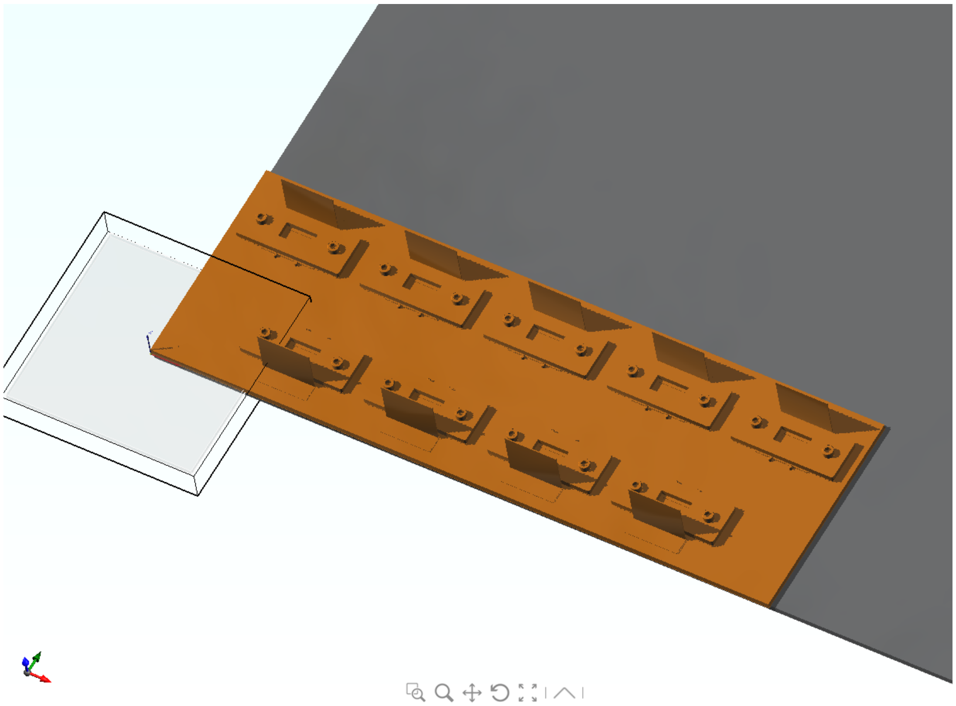
1. Open or create an IPGScan "On The Fly" type job.
2. Connect to the desired robot and perform a calibration if not already done.
    a. For this demonstration, the IPGScan world was calibrated according to a 1230mm x 610mm piece of sheet metal (see Figure 12-77), which is the typical working area for this robot setup.

*Figure 12-77 IPGScan and Real World Calibration*



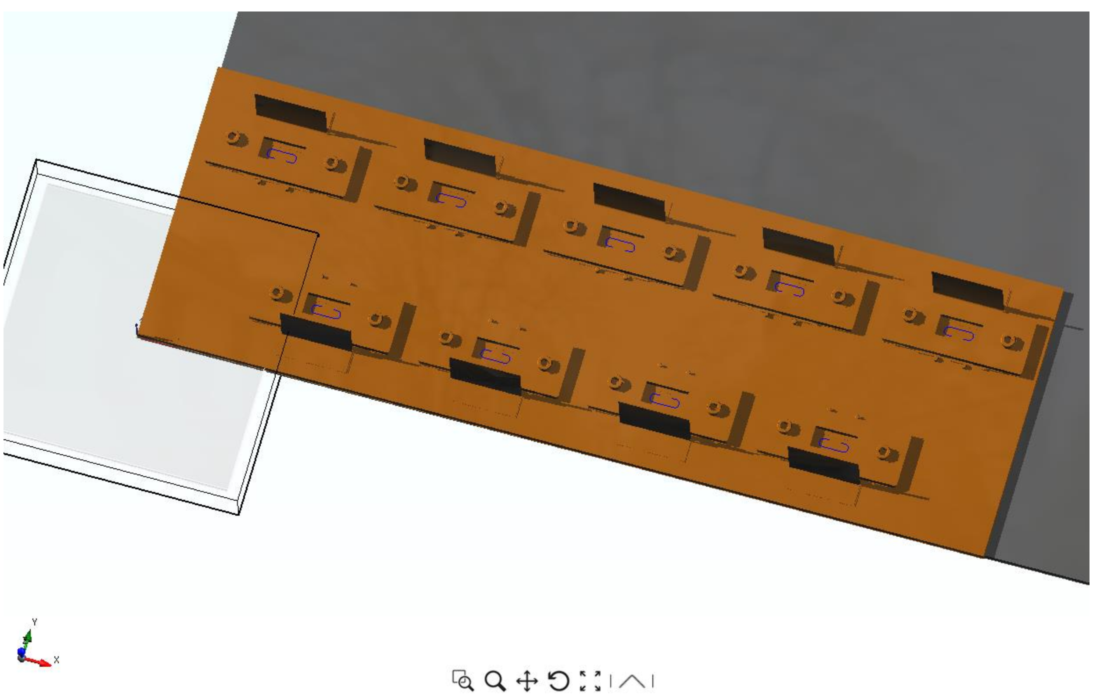

3. Import the desired STL model into IPGScan.
    a. This demonstration utilizes a fixture with coupons clamped in it. See Figure 12-78.

*Figure 12-78 Importing and STL Model*



4. (Optional) Place any desired process objects (i.e. welds) along the STL model. See Figure 12-79.

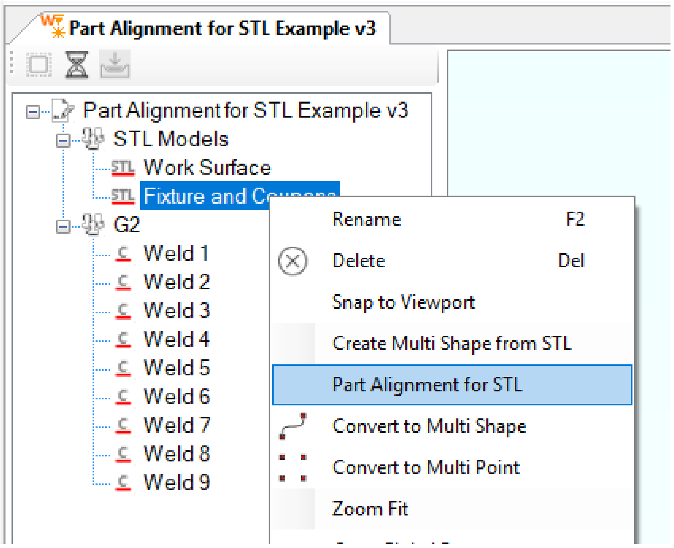
*Figure 12-79 - Placing Process Objects Relative to an STL Model*



5. Next, the user will need to determine a minimum of three points to calibrate the “real world” part to the IPGScan STL model.

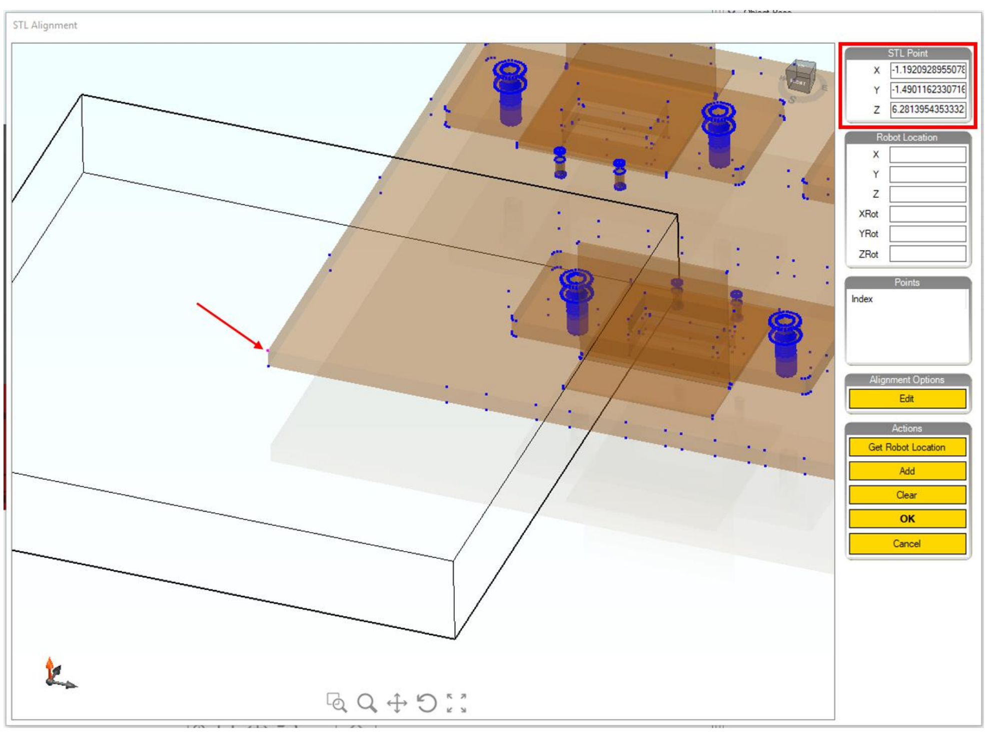
6. Right click the STL name of the model to be calibrated in the Job Tree. Click on "Part Alignment for STL" as seen in Figure 12-80. This will open the STL Alignment Window.

*Figure 12-80 Part Alignment for STL*



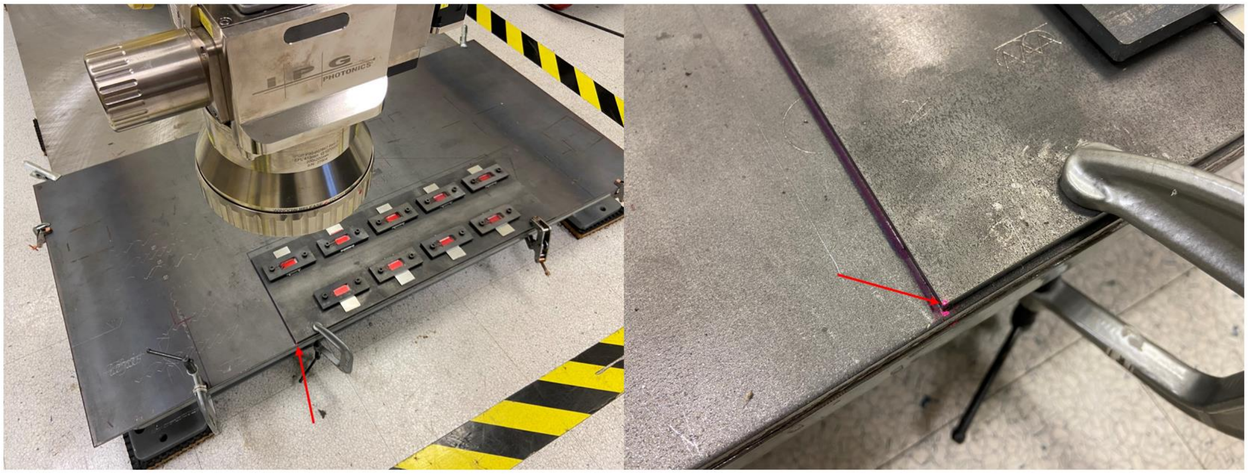

7. Select a defined point on the STL model to calibrate (see Figure 12-81). Notice the "STL Point" data pane populates with the selected points pose.

*Figure 12-81 Selecting a Point to Calibrate*



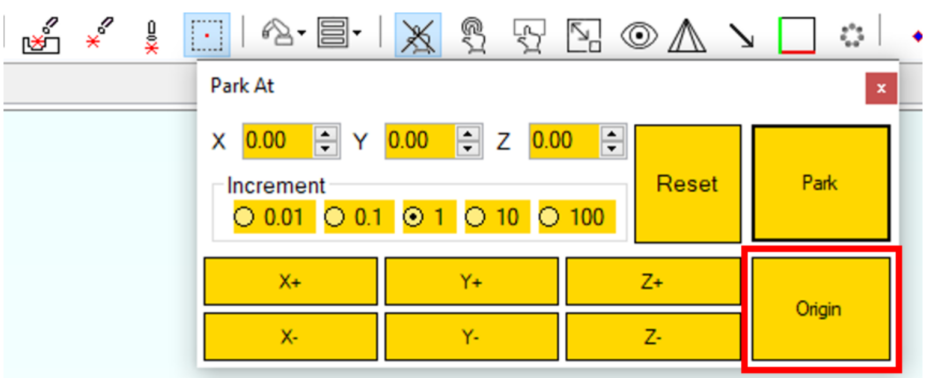

8. Jog the robots TCP to the defined point on the STL model in the "real world." This will look similar to what is seen in Figure 12-82.

*Figure 12-82 Jogging the Robot TCP to the Definded STL Calibration Point*



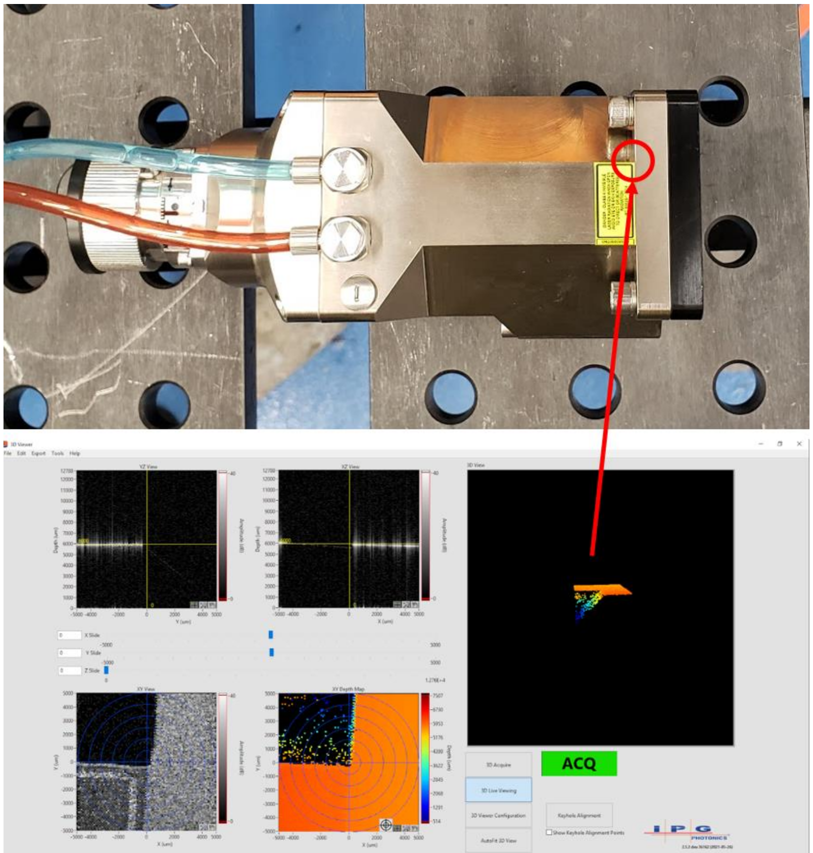
a. In the above example, the guide laser is parked at 0, 0, 0 in the scanner Field of View (FOV). Working distance is measured using a ruler to ensure the proper stand-off height of the head to the point on the fixture.
b. Please note, to most accurately locate the robots TCP at the defined point on the part or fixture, it is recommended that users utilize a camera or LDD unit. It also is recommended that users have setup the robots TCP at 0, 0, 0, in the scanner's FOV (as outlined in the "Robot and IPGScan Calibration" reference document). By adhering to both of these recommendations, users can enable the "Park At" feature and park the galvos at 0, 0, 0, in the scanners FOV (see Figure 12-83).

*Figure 12-83 Using "Park At" to Position the Galvos at 0, 0, 0 in the Scanners FOV*



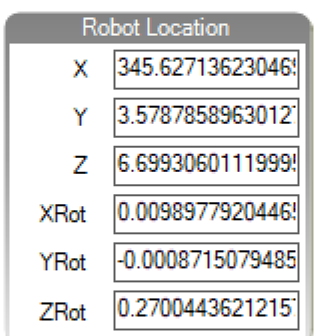

c. With the galvos parked at 0, 0, 0 in the FOV, the user can then accurately align the robots TCP with the desired part location by observing the camera or LDD crosshair setup while the robot is jogged into the correct position on the part. Figure 12-84 demonstrates an example of how the LDD imagining system can be used to locate features on a 3D surface.

*Figure 12-84 Example of using LDD to Locate Part Features for Alignment*



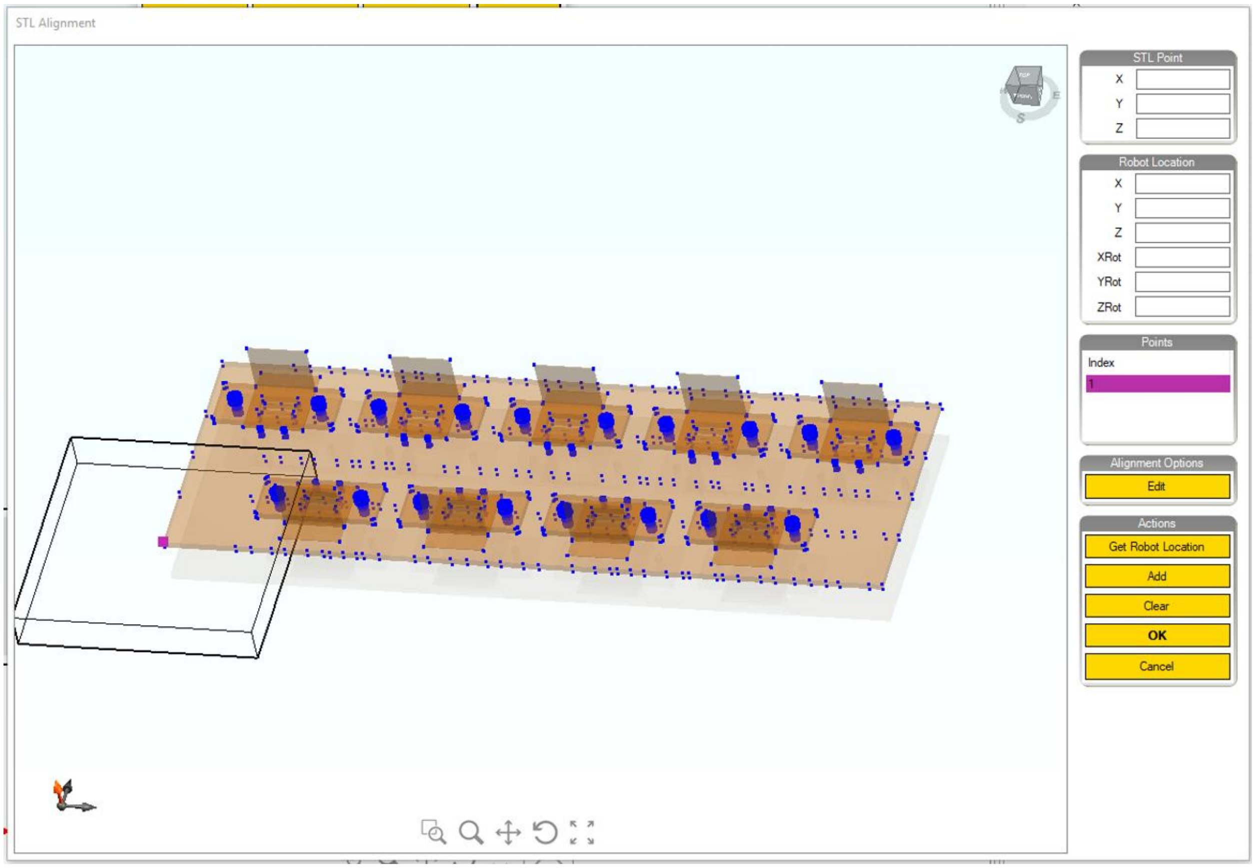

9. With the robot TCP in the proper location, click “Get Robot Location” in the STL Alignment Window. This will populate the “Robot Location” coordinates with the proper pose information (see Figure 12-85).

*Figure 12-85 Robot Pose Data*



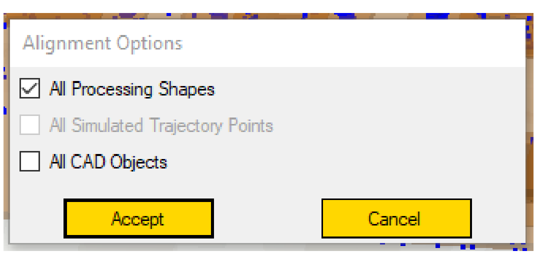

10. Click “Add” in the STL Alignment Window. This will add a calibrated point to the “Points” list (see Figure 12-86).

*Figure 12-86 Calibrating the Point*




11. Repeat steps 6-9 for at least 2 more unique points.
    a. Although not required, it is recommended that 4 points be taught. By increasing the number of calibration points beyond three, the accuracy of placement should increase. A positional average is calculated based on the calibration points.
12. Before completing the calibration process, select any desired Alignment Options. See section "Alignment Options," for additional details concerning the available alignment options.
    a. For this example, only the "All Processing Shapes" box was checked. This will cause all of the weld objects to snap with the STL model. See Figure 12-87.

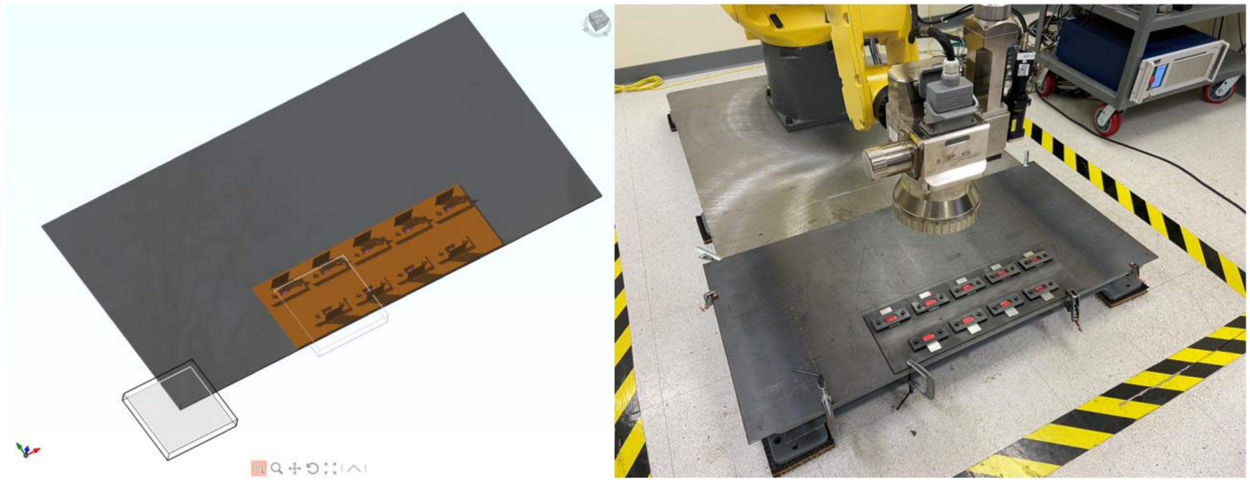
*Figure 12-87 - Example Alignment Options*




13. Once a minimum of three points are calibrated, click "OK."
14. The STL model should now be calibrated in IPGScan accordingly with the "real world."

a. Figure 12-88 details the STL Model in IPGScan after performing STL Alignment.

*Figure 12-88 IPGScan STL Model Calibrated to the Real World*

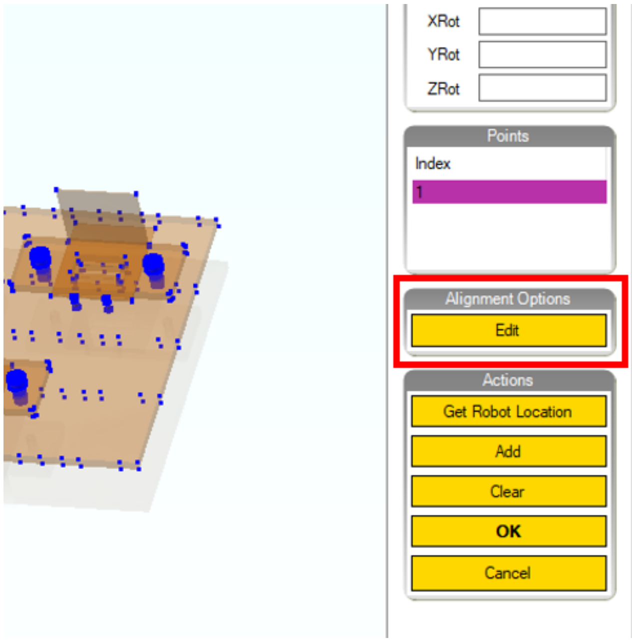


### 12.4.8.1 Alignment Options

The following steps detail the Alignment Options.

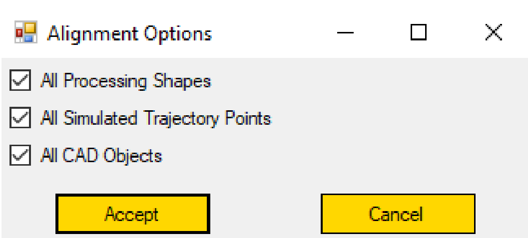
1. Within the STL Alignment window, users have the ability to set Alignment Options. Click “Edit” to open the Alignment Options Window. See Figure 12-89.

*Figure 12-89 Opening the Alignment Options Window*




2. In the Alignment Options Window, users have the ability to select what objects will snap (maintain their positional relationship) with the STL model that is being aligned. See Figure 12-90.
    a. All Processing Shapes – If checked, all weld objects will snap with the aligned STL model.
    b. All Simulated Trajectory Points – If checked, any simulated trajectory will be snapped with the aligned STL model.
    c. All CAD Objects – If checked, any other CAD objects within the job will be snapped with the aligned STL model.

*Figure 12-90 Alignment Options*



## 12.4.9 Process Object Placement/Touch-up

IPGScan offers numerous tools for the positioning of Process Objects. The following sections detail the available functionality that allows users to place/touch-up process objects in an OTF job.

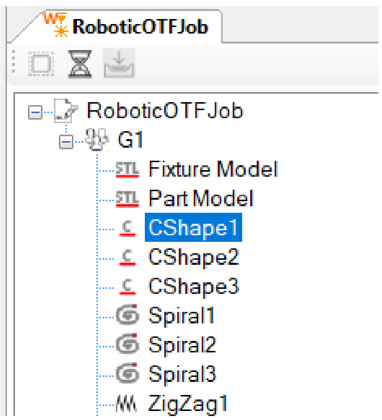
### *12.4.9.1 The Alignment Window*

When utilizing an On-The-Fly type job, users have access to the Alignment window. Within this window, two different methods of alignment for Process Objects exists, which includes Process Alignment and Robot Alignment.

To access the Alignment Window, perform the following steps.

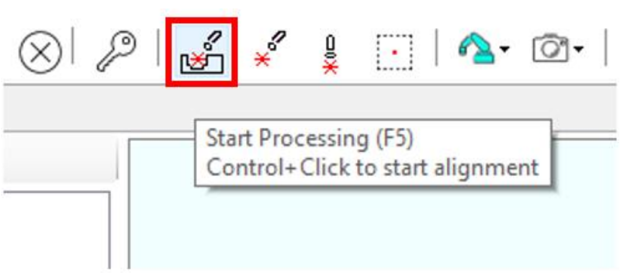
1. Open or create the desired OTF type job.
2. Select the desired Process Object(s) for alignment. See Figure 12-91 as an example.

*Figure 12-91 Selecting the Desired Process Object(s)*



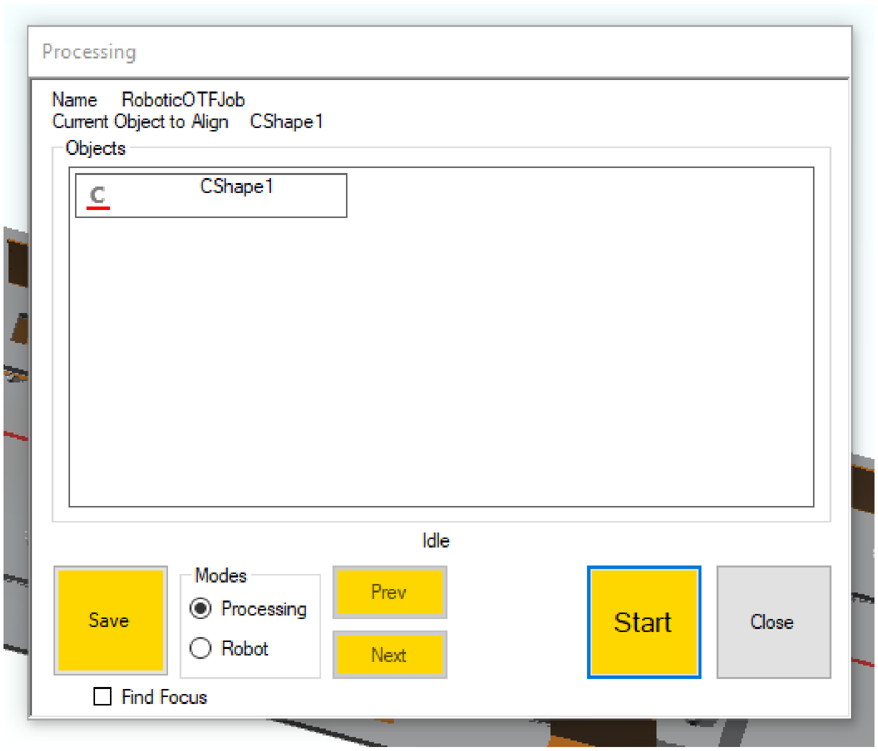

3. Hold ctrl on the keyboard and click (ctrl + click) the “Start Processing” button. See Figure 12-92.

*Figure 12-92 Opening the Alignment Window*




4. The Alignment Window will now be open. See Figure 12-93.

*Figure 12-93 The Alignment Window*



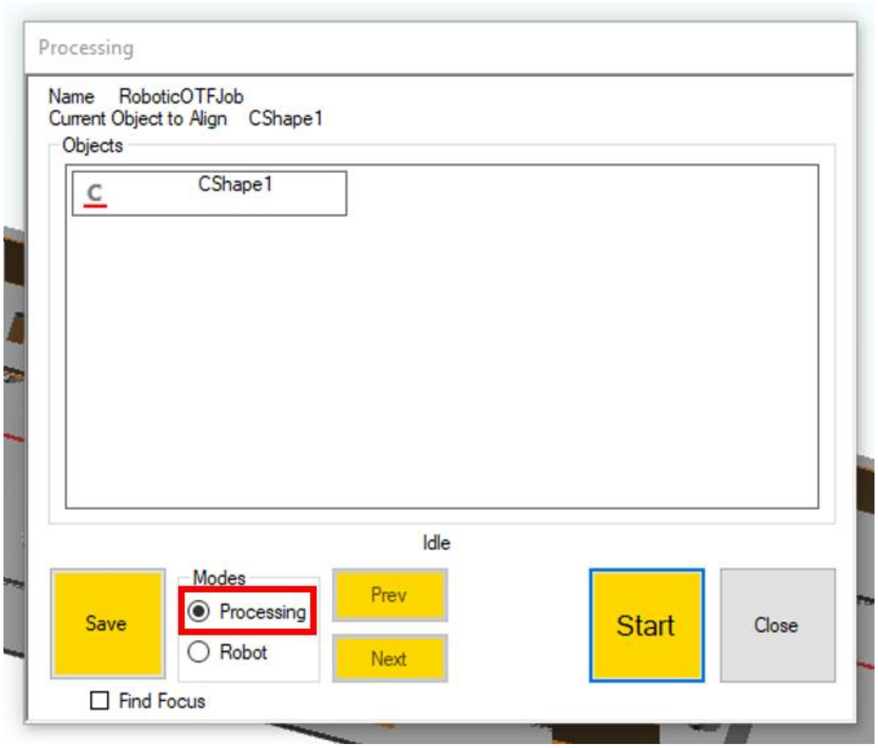

With the Alignment Window open, users can start to utilize the Processing and Robot Alignment modes.

##### 12.4.9.1.1 Process Alignment

Process Alignment allows users to position Process Objects within the IPGScan world by driving the robot in the real world. When Process Alignment is started, the guide laser projects an image of the given object in the scanners FOV. Users can then jog the robot into the proper positon so that the Process Object is positioned as desired in relation to the part.

To utilize Process Alignment, ensure the "Processing" radio button is selected prior to clicking "Start." See Figure 12-94.

*Figure 12-94 Selecting Processing Mode*



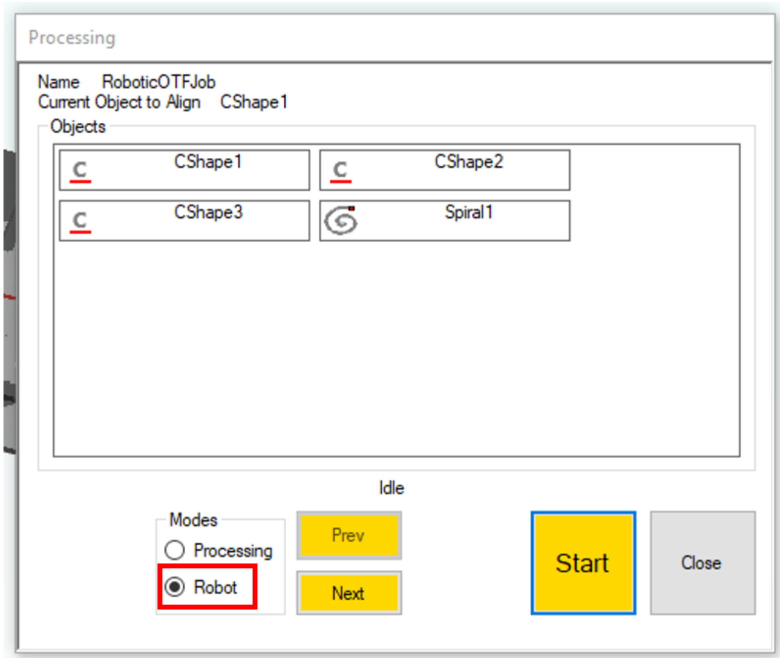

Once a Process Object is positioned as desired, click “Save” to update the objects position (global pose) within IPGScan.

#### 12.4.9.1.2 Robot Alignment

Robot Alignment allows users to preview as much of the selected process objects as possible in its programmed location based upon the current location of the robot. This can be used when programming the trajectory of the robot to determine if a location will be suitable for outputting the tested object.

To enable Robot Alignment, simply select the desired Process Objects, open the Alignment Window, and select the Robot option radio button. See Figure 12-95.

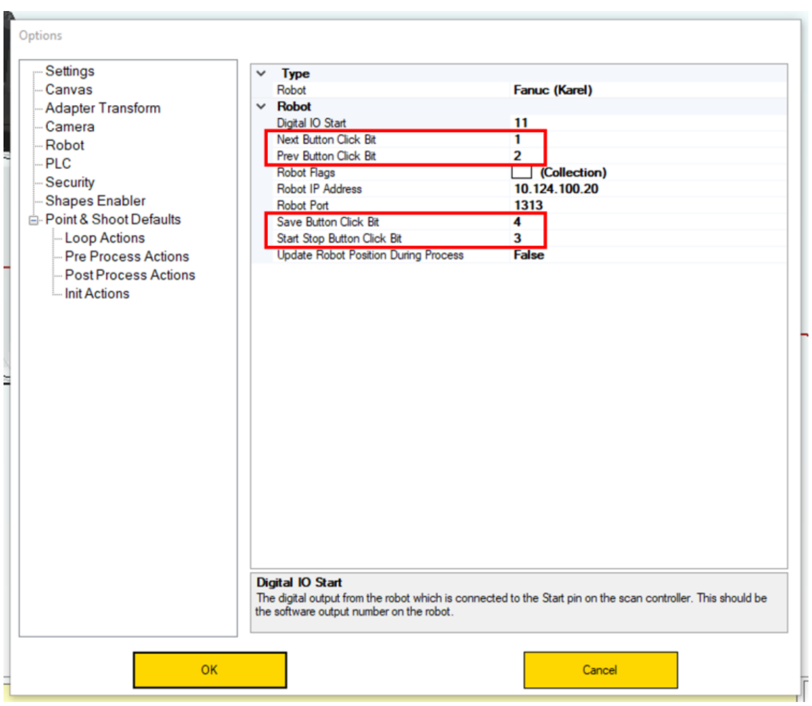
*Figure 12-95 Selecting Robot Mode*




### 12.4.9.1.3 Alignment with the Robot

From the robot options, the different "Bit" options refer to the alignment window. These bit options should be set to outputs on the robot connected to the Scan Controller. Changing the state of these outputs will trigger a corresponding button press on the alignment window. Figure 12-96 outlines the available click bit options.

*Figure 12-96 Robot Option Click Bits*



For example, if in the robot options “Save Button Click Bit” is set to 0 (Strobe), changing the Strobe bit will save the current pose of the robot to the current processing object. Use of these bits is optional.

Table 12-4 shows the bits of user accessible digital inputs to the Scan Controller from the 24V Interface Board.

*Table 12-4 Bits of User Accessible Digital Inputs*

| Signal | Bit |
|---|---|
| SELECT 0 | 16 |
| SELECT 1 | 17 |
| SELECT 2 | 18 |
| SELECT 3 | 19 |
| SELECT 4 | 20 |
| SELECT 5 | 21 |
| SELECT 6 | 22 |
| SELECT 7 | 23 |
| SELECT 8 | 24 |
| STROBE | 0 |

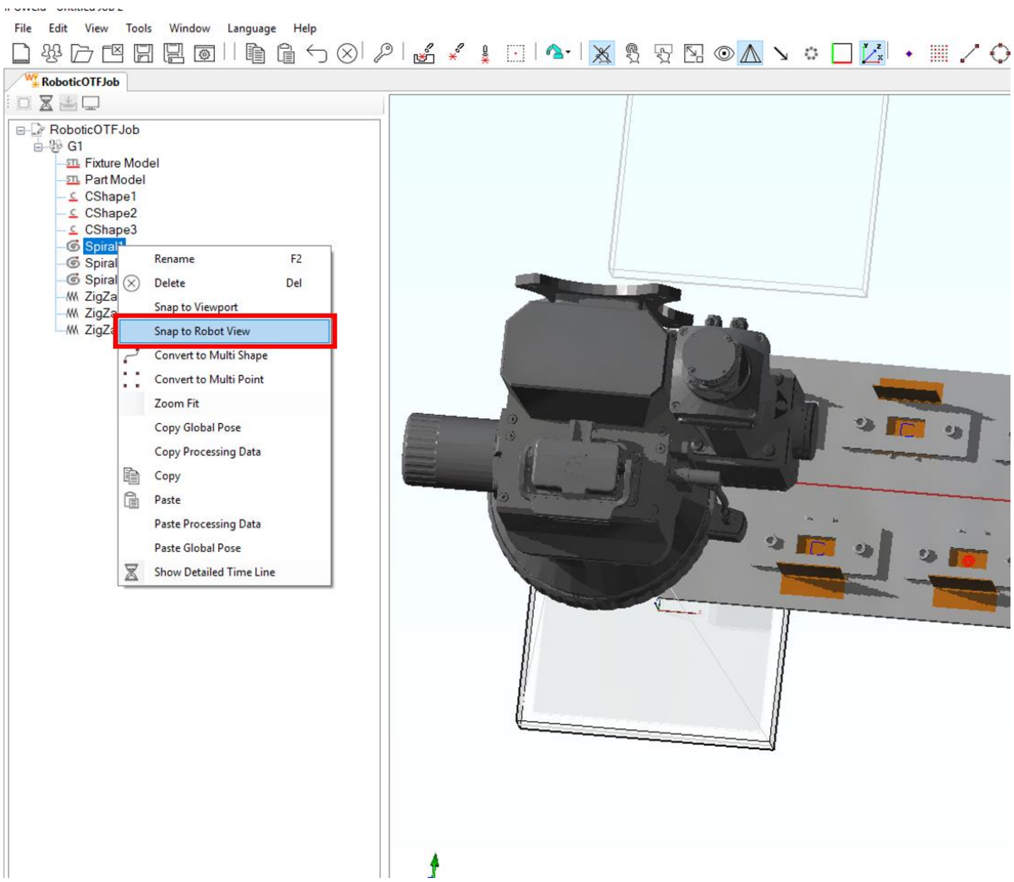
###### 12.4.9.1.4 Snap To

“Snap to Robot View” and “Snap to Viewport” allow users to quickly snap objects into position from the Job Tree.

Snap to Robot View allows users to positon objects based on the robot’s position (robot TCP).

1. Enable Robot Tracking by selecting Robot Menu → Robot Tracking.
2. Move the robot to the desired location. The movement of the robot will be reflected on the IPGScan Canvas.
3. Right click on any object and select “Snap to Robot View.” See Figure 12-97.

*Figure 12-97 Snap to Robot View*

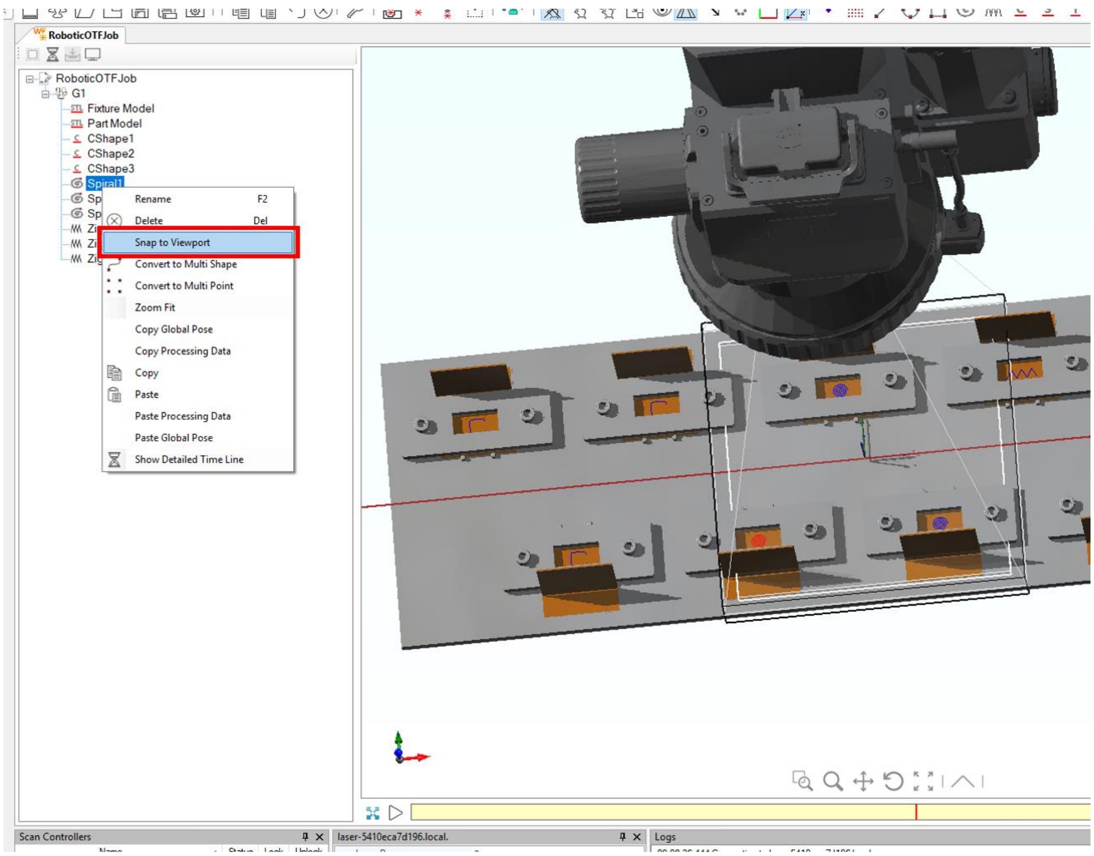



4. The current robot pose is now saved as the object's global pose.

Snap to Viewport allows users to position objects based on the viewport's current position in a trajectory.

1. Record and load a trajectory or create a simulated trajectory.
2. At the bottom of the canvas, scroll along the basic timeline by clicking on it and moving the mouse left and right.
3. When the viewport is at a desired location in the trajectory, right click on any object and select "Snap to Viewport." See Figure 12-98.

*Figure 12-98 Snap to Viewport*




4. The current viewport pose is now saved as the object's global pose.

## 12.5 Robot Program Structure

IPGScan records the motion of the robot program for later use in a processing job. The robot program must be setup accordingly.

The Start signal changing from inactive to active signals IPGScan to begin recording. The Start signal changing from active to inactive signals IPGScan to finish recording.

Because IPGScan works off of a recording, only time-constant functions should be used within the recorded section of the job. For example, if the robot waits for an input, the robot and scanner will be desynchronized if the input is not triggered at the same time during each cycle.

Figure 12-99 demonstrates example pseudo-code for a robot program for OTF.

*Figure 12-99 Example Robotic OTF Pseudo-Code*

```
// reset the start signal to off in case it is on at the start of
// the program
   digitalOutput[startSignal] off
// move to a beginning home position
   moveJ pHome
// any other preparations for the trajectory can be done here
// move to the first point in the trajectory
   moveJ pBegin
// turn on the Start signal begins recording this trajectory
   digitalOutput[startSignal] on
// any number of moves can be performed
// additional operations like setting outputs or waiting based on
// time can be performed
// moves which are not time-constant should not be used because
// IPGScan will only use a recording during execution
   moveL pIntermediary1
   moveL pIntermediary2
// turning off the Start signal will end the recording of this
// trajectory
   digitalOutput[startSignal] off
// return to the home position
   moveJ pHome
```

### 12.5.1 FANUC On-The-Fly Programming

For FANUC OTF programming, it is recommended that the following practices be implemented into robot program creation.

- Utilize a lead-in point with a FINE termination type before the position where Start is set active.
- Utilize a FINE termination type point for the position where Start is set active.
- Set Start active on a separate line from the motion point.

Figure 12-100 represents an example robotic OTF program for a FANUC robot.

*Figure 12-100 Example FANUC OTF Program*

```
/PROG  FANUC_OTF_EXAMPLE
/ATTR
OWNER           = MNEDITOR;
COMMENT         = "";
PROG_SIZE       = 1301;
CREATE          = DATE 22-05-16  TIME 12:23:52;
MODIFIED        = DATE 22-05-16  TIME 13:01:40;
FILE_NAME       = LOM_OTF2;
VERSION         = 0;
LINE_COUNT      = 29;
MEMORY_SIZE     = 1557;
PROTECT         = READ_WRITE;
TCD:  STACK_SIZE        = 0,
      TASK_PRIORITY     = 50,
      TIME_SLICE        = 0,
      BUSY_LAMP_OFF     = 0,
      ABORT_REQUEST     = 0,
      PAUSE_REQUEST     = 0;
DEFAULT_GROUP   = 1,*,*,*,*;
CONTROL_CODE    = 00000000 00000000;
/APPL
  ARC Welding Equipment : 1,*,*,*,*;
/MN
   1:  !FRAMES ;
   2:  UFRAME_NUM=2 ;
   3:  UTOOL_NUM=2 ;
   4:   ;
   5:  !IPG SCANNER ENABLE ;
   6:  DO[12:ENABLE]=ON ;
   7:   ;
   8:  !ROBOT VELOCITY DURING WEDLING ;
   9:  R[1:LINEAR_SPEED]=200    ;
  10:   ;
  11:  !SAFEHOME ;
  12:J P[1] 20% FINE    ;
  13:  WAIT DI[1:READY]=ON AND DI[2:ACTIVE]=ON    ;
  14:   ;
  15:  !LEAD-IN POSITION ;
  16:J P[2] 20% FINE    ;
  17:  !START POSITION ;
  18:L P[4] R[20:SPEED 1]mm/sec FINE    ;
  19:  DO[11:START]=ON ;
  20:  !TRAJECTORY MOTION POINTS ;
  21:L P[5] R[20:SPEED 1]mm/sec CNT100    ;
  22:J P[3] R[26:JOINT SPEED]% FINE ACC50    ;
  23:L P[13] R[21:SPEED 2]mm/sec CNT100 ACC50    ;
  24:J P[14] R[26:JOINT SPEED]% FINE ACC50    ;
  25:  !START INACTIVE ;
  26:L P[15] R[22:SPEED 3]mm/sec FINE ACC50    ;
  27:  DO[11:START]=OFF ;
  28:  !SAFEHOME ;
  29:J P[1] 20% FINE    ;
```

### 12.5.2 KUKA On-The-Fly Programming

OTF KUKA modules need to start RSI at the beginning of the program to facilitate recording.

1. To start RSI, IPG_OTF_BEGIN_DATA_XFER should be called at the beginning of the module.
2. At the end of the program, a small time delay should be called before ending the job or ending RSI to give the robot time to send out all of the information. This delay only needs to be called during recording. Calling it after setting the Start signal to inactive will not impact the final recording.
3. IPG_OTF_END_DATA_XFER can be used to both end RSI and add a time delay. Both modules require a Boolean parameter, true if the run will be a recording and false if the run will not be a recording. Both the RSI functions and time delay will only be called during recording.

Figure 12-101 represents an example OTF KUKA program in inline forms.

*Figure 12-101 Example KUKA Program*

```
DEF kuka-program-example( )
INI

; Begin data transfer with IPGScan
IPG_OTF_BEGIN_DATA_XFER(TRUE)

; Go to Home Position

PTP SAFEHOME  Vel=100 % PDAT2 Tool[0] Base[0]

; Wait until IPGScan is waiting for the start signal

WAIT FOR ( IN 1 'SCAN_READY' AND IN 2 'SCAN_ACTIVE' )

; Turn on start signal syncronized with the beginning of trajectory

SYN OUT 11 'SCAN_START'  State= TRUE  at END  Delay= 0 ms

; Move to the start of the trajectory

LIN TRAJ_START  Vel=0.1 m/s CPDAT1 Tool[0] Base[0]

; Turn off start signal syncronized with the end of trajectory

SYN OUT 11 'SCAN_START'  State= FALSE  at END  Delay= 0 ms

; Move to the end of the trajectory

LIN TRAJ_END  Vel=0.1 m/s CPDAT2 Tool[0] Base[0]

; Return to SAFEHOME

PTP SAFEHOME  Vel=100 % PDAT1 Tool[0] Base[0]

; End data transfer with IPGScan
IPG_OTF_END_DATA_XFER(TRUE)

END
```

### 12.5.3 ABB On-The-Fly Programming

1. The system module IPG_OTF contains three routines which setup, start, and stop EGM. Please refer to Table 12-5.

*Table 12-5 ABB Robotic OTF Routines*

| Routine | Parameters | Description | Example |
|---|---|---|---|
| `rIPGSetupOtf` | string `sStartSignal` – name of the signal connected to the start digital input on the scan controller `string sExtConfigName` – name of the external motion interface data name `string sUdpUcDevice` – UdpUc device name | Sets up the helper functions for the EGM. | `rIPGSetupOtf "DO07_ScannerStart", "default", "EGM_Config";` |
| `rIPGStartOtf` | bool recording – is this function executing during a trajectory recording (true) `num delayTime` – time to delay turing on of the start signal (not used) | starts EGM stream | `rIPGStartOtf TRUE, 0;` |
| `rIPGStopOtf` | none | Stops EGM stream | `rIPGStopOtf` |

2. To record a trajectory, *rIPGSetupOtf* should be called before *rIPGStartOtf*. The state of the start signal is not recorded, so all points in the steam are saved to the trajectory. *rIPGStopOtf* should be called when the recorded trajectory has completed. The start signal should be in place during recording for consistency. It is suggested to use a Trigger command for repeatability and to call this Trigger command right before calling *rIPGStartOtf*. These functions should not be called during execution to increase performance.

### 12.5.4 Recording and Loading Trajectories

Once users are satisfied with a given robot trajectory, this trajectory should be recorded and loaded into IPGScan.

It is critical that the trajectory be captured exactly how the robot will be running in the production environment or how the user intends the robot to run in its final state. This means that if users plan on running the robot in an Automatic mode (instead of Teach mode), the trajectory should be captured with the robot in Automatic mode. Additionally, if changes are ever made to the robot program after the trajectory is captured and loaded into IPGScan, users should recapture and load the new trajectory in IPGScan.

| | |
|---|---|
| **IMPORTANT** | **It is imperative that the robots trajectory be representative of what the robot is actually doing during a system cycle. If the trajectory loaded into the IPGScan job does not match what the robot is actually doing, the scanner and robot** |

**synchronization will be off and process results may be distorted and/or in improper output locations.**

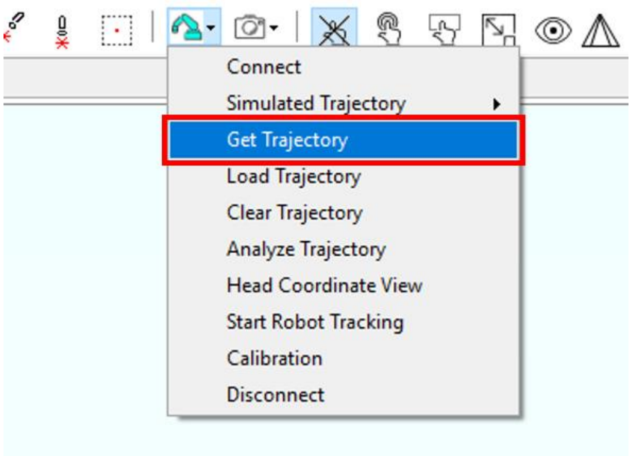
The following sections detail how users can capture and load a trajectory within IPGScan.

### 12.5.4.1 Recording a Robot Trajectory

Please refer to the following steps for recording a robot trajectory.

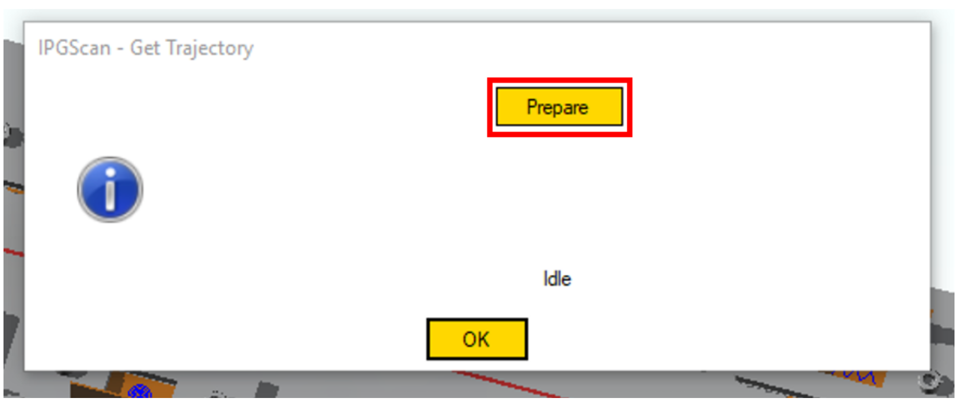
1. Click on the Robot icon in the Tool Bar.
2. Click "Get Trajectory" from the robot menu. See Figure 12-102.

*Figure 12-102 Get Trajectory*



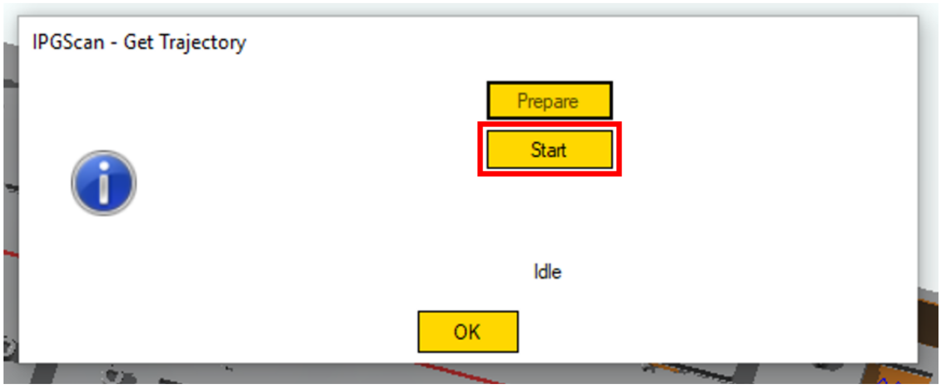

3. Press "Prepare" to prepare IPGScan and the robot for recording. See Figure 12-103.

*Figure 12-103 Preparing for Recording*




4. Press "Start" to begin recording the trajectory in IPGScan. See Figure 12-104.
   a. Recording will begin with the Start signal is set active on the user Interface board.

*Figure 12-104 Starting Recording*




5. Run the robot program as it is expected to run during processing.
    a. The robot program should set Start active to begin recording the trajectory. At the end of the desired trajectory, the robot should set Start inactive to stop the recording process.
6. After completing a recording, a window will display to choose a save location and a file name for the recording file.

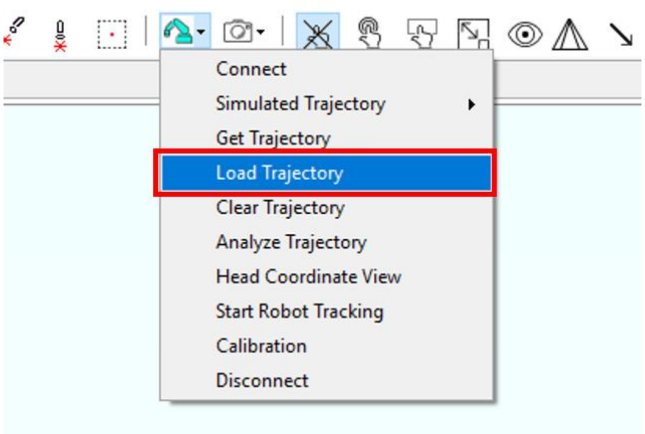
| | |
|---|---|
| **IMPORTANT** | **When recording with FANUC_KAREL, the recording job on the robot has a very high priority and you may notice that the iPendant does not respond as expected to button presses. This is only during recording with FANUC_KAREL and will finish after the recording completes. If you are still having trouble, run IPG_ABORT to abort running the OTF KAREL jobs.** |

### *12.5.4.2 Loading a Robot Trajectory*

The following steps detail how to load a trajectory in IPGScan.

1. Click on the Robot icon in the Tool Bar.
2. Select "Load Trajectory" from the robot menu. See Figure 12-105.

*Figure 12-105 Loading a Trajectory*




3. Navigate the file system and select the desired trajectory file.

4. Click "Open."

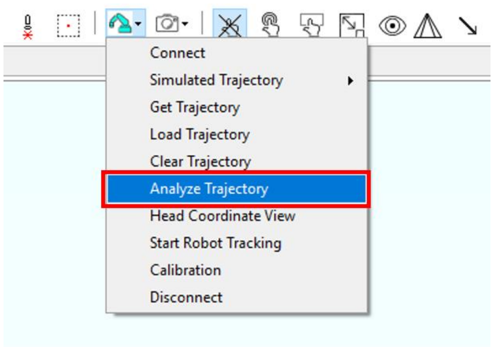
### 12.5.4.3 Analyze Trajectory (Acceleration Spikes)

Analyze Trajectory is a feature that allows users to open a graph which displays acceleration curves throughout a given trajectory. The purpose of this graph is for users to be able to gain a better understanding of where high acceleration zones may be experienced throughout a given robot trajectory. Given that processing is best suited during constant motion, processing during high acceleration zones may result in diminished output quality due to inconsistent robot motion. For this reason, users can utilize the Analyze Trajectory graph to better visualize where the high acceleration zones occur in the trajectory and better plan Process Object timings (using delay actions) to take place during low acceleration zones.

To access the Analyze Trajectory graph, perform the following steps.

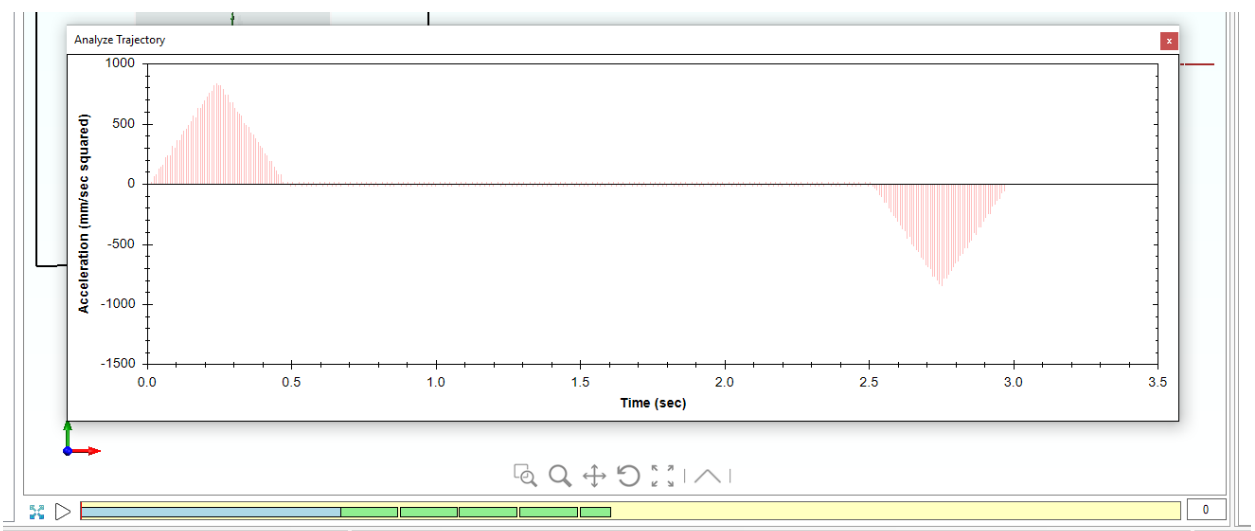
1. Click on the Robot icon in the Tool Bar.
2. Select "Analyze Trajectory." See Figure 12-106.

*Figure 12-106 Analyze Trajectory*




3. With the Analyze Trajectory graph open, users can examine times where high acceleration zones occur and compare that to the Process Object timings on the Basic Timeline. See Figure 12-107.

*Figure 12-107 Analyze Trajectory Graph*



### 12.5.5 Coordination Flags

Coordination Flags allow users to define zones throughout a given robot trajectory. These zones can be utilized to control process timings.

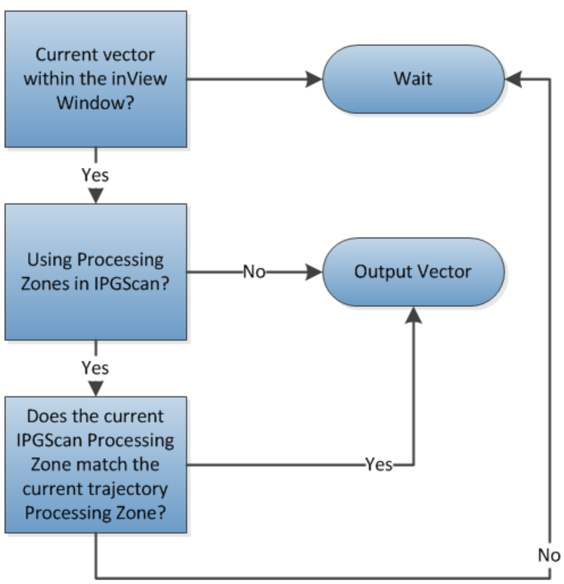
**IMPORTANT** **Coordination Flags do not work with ABB robots.**

Action Control → "Set Coordination Flags" is used to set Processing Zones in IPGScan. A Processing Zone is active for all processing objects following the Action Control until: 1) the job ends or 2) another Processing Zone is set.

The active Coordination Flags in the robot trajectory must match those of the current Processing Zone in order for an object to be output. Calling "Set Coordination Flags" with the default parameter of "0x00" will ignore all Processing Zones.

The process that ScanPack uses to determine whether the upcoming processing vectors should be output is found in Figure 12-108.

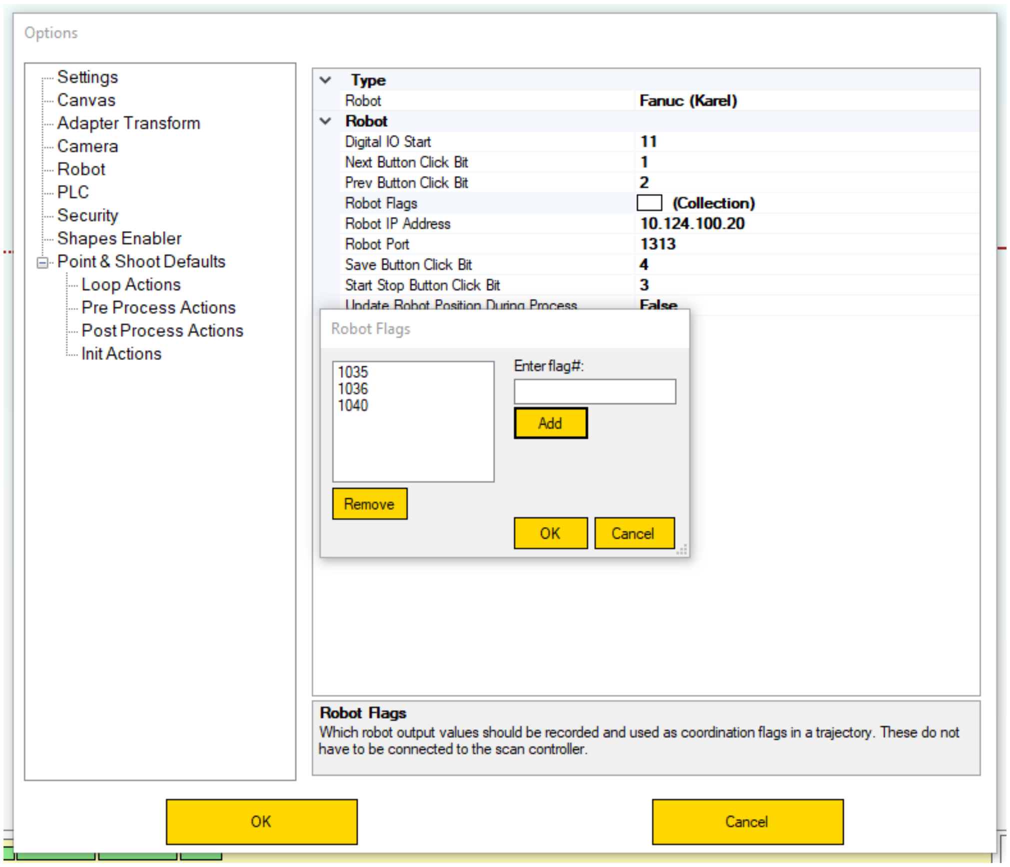
*Figure 12-108 Coordination Flag Logic*




#### *12.5.5.1 Selecting Coordination Flags*

Coordination Flags can be selected under Options → Robot → Robot Flags. See Figure 12-109.

*Figure 12-109 Setting Robot Flags*




Enter values that correspond with any robot output. Coordination flags must be set before recording a trajectory.

### 12.5.5.1.1 FANUC

For FANUC robots, enter the output number as displayed on the robot. FANUC systems using the Robot Server are limited to 32 flags, systems using KAREL are limited to 5 flags. If more than 5 or 32 flags are set in the options menu, the first 5 or 32 values will be used, respectively. If a value is invalid it will not be used.

### 12.5.5.1.2 Yaskawa Motoman

For Yaskawa Motoman robots, enter the logical relay address number. This can be found in the Concurrent I/O manual and in the In/Out menu. Yaskawa Motoman robots are limited to 5 flags. If more than 5 flags are set in the options menu, the first 5 values will be used. If a logical relay address number is invalid, it will not be used.

### 12.5.5.1.3 KUKA

For KUKA robots, enter the number of flags to record. If nothing is entered, 0 is assumed. Values to be used as flags must be set in the RSI files before loading them onto the robot. The RSI files supplied are limited to 9 flags not including the Start signal.

#### *12.5.5.2 Setting the Coordination Flag Action Control*

In the Action Control, “Set Coordination Flags,” the value of the flag is a 32-bit hexadecimal number. Each bit represents the state of one flag. Flag 0 corresponds to the first digital output listed in the IPGScan settings. The flag number does not correspond to the output number. Flags are numbered based upon their order in the options. When a flag is active, that bit is equal to 1. When a flag is inactive, that bit is equal to 0. Table 12-6 shows the first 8 bits of the 32-bit number.

*Table 12-6 Robot Flag First 8-bits*

| FLAG7 | FLAG6 | FLAG5 | FLAG4 | FLAG3 | FLAG2 | FLAG1 | FLAG0 |
|---|---|---|---|---|---|---|---|

## 12.6 Trigger Delay

### 12.6.1 Overview

On-The-Fly Trigger Delay may be necessary in instances where the robot trigger does not take place exactly where the user expects it too. For instance, triggers are often times slightly delayed from a robot program position simply because it takes the robot time to register that the digital bit must be set active/inactive. This ultimately can result in an offset of laser output from the users desired program location using the guide laser.

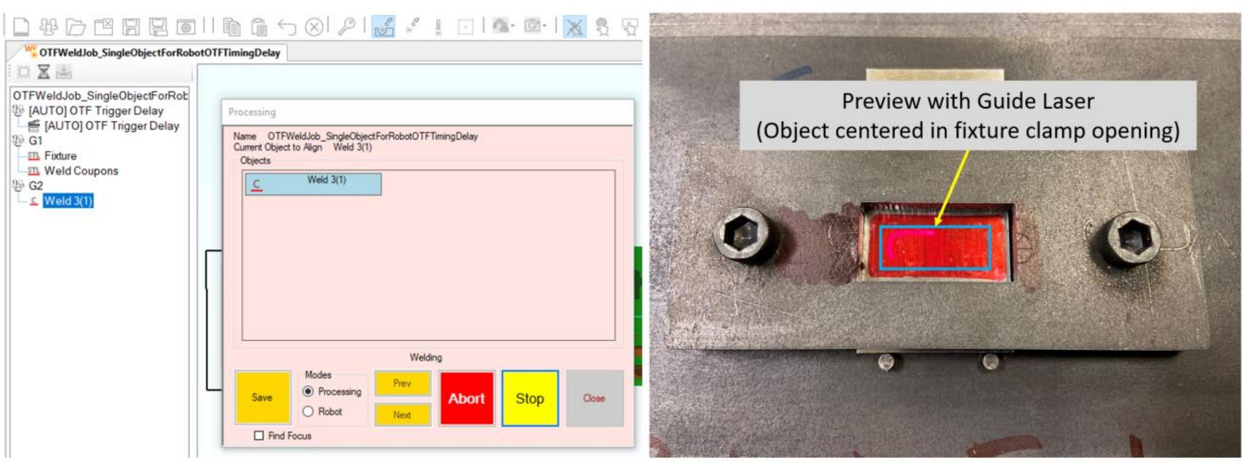
**IMPORTANT** **A consistent trigger is required in order for OTF Trigger Delay to be effective. In most cases, a discrete IO connection is going to provide the most consistent trigger. Fieldbus interfaces will typically introduce latency and not provide a consistent trigger. In an example such as this, the OTF Trigger Delay would not be able to compensate for the inconsistent trigger.**

### 12.6.2 Procedure

The following steps outline a procedure that users can follow to utilize the OTF Timing Delay functionality and more accurately dial in the laser output to their programmed location. This process can either be completed by setting up a dedicated test (that uses robot and weld speeds similar to a production process) or by running multiple cycles on an existing system.

1. Start by examining and touching-up the program location of a given object using the guide beam. For OTF processes, this should be done using the Processing Alignment Window.
    a. In this example, a staple shape weld is centered in the opening of a fixture clamp. See Figure 12-110.
    b. It is good practice to save the robot position for this weld position. This will make it easy to return to this location later on in the procedure.

*Figure 12-110 Placing a Process Object Using the Guide Laser*

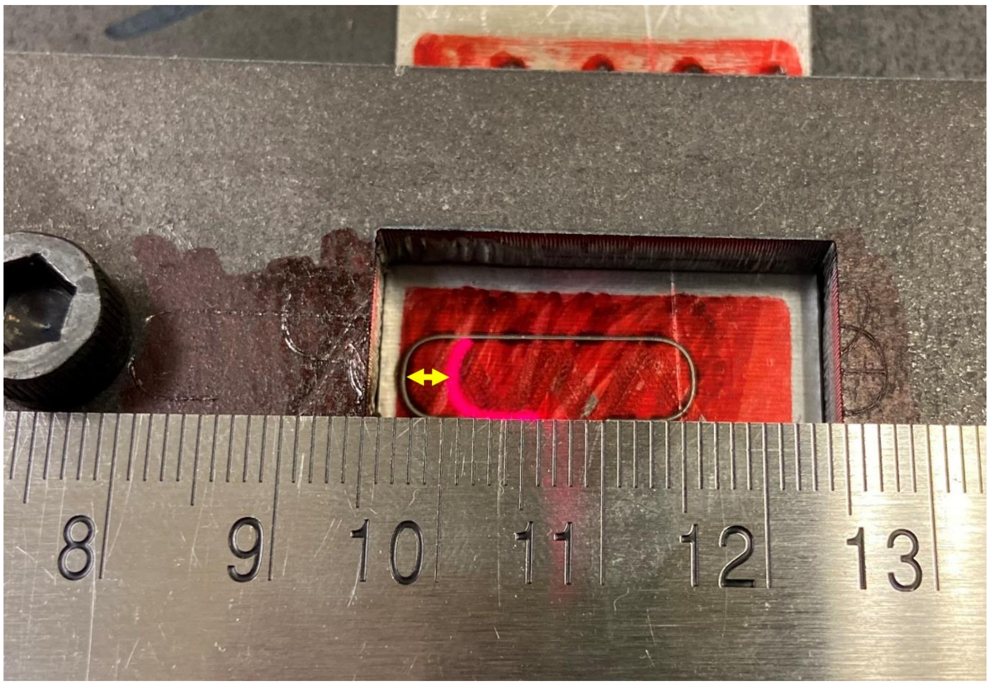



2. With the process object position touched-up, the OTF process can now be run with the laser.
    a. Prior to cycling the actual equipment, be sure to perform a Dryrun of the process to ensure the job will properly execute.

b. Take note of the robot travel speed for the OTF process. This value will be required for calculating the required trigger delay. In this example, a robot speed of 150mm/sec is used.

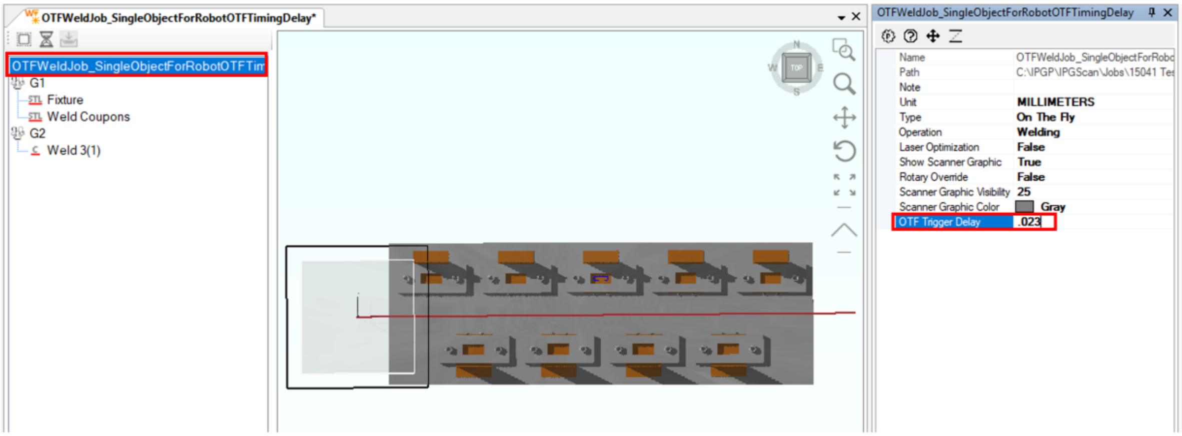
3. Once the process completes, reposition the head to preview the object placement with the guide laser (as outlined in step 1).
4. Measure the offset distance between the guide laser and the actual laser output. See Figure 12-111.
   a. In this example, a distance of 3.5mm was measured.

*Figure 12-111 Measuring the Offset Distance between the Guide Laser Projection and the Laser Output Position*



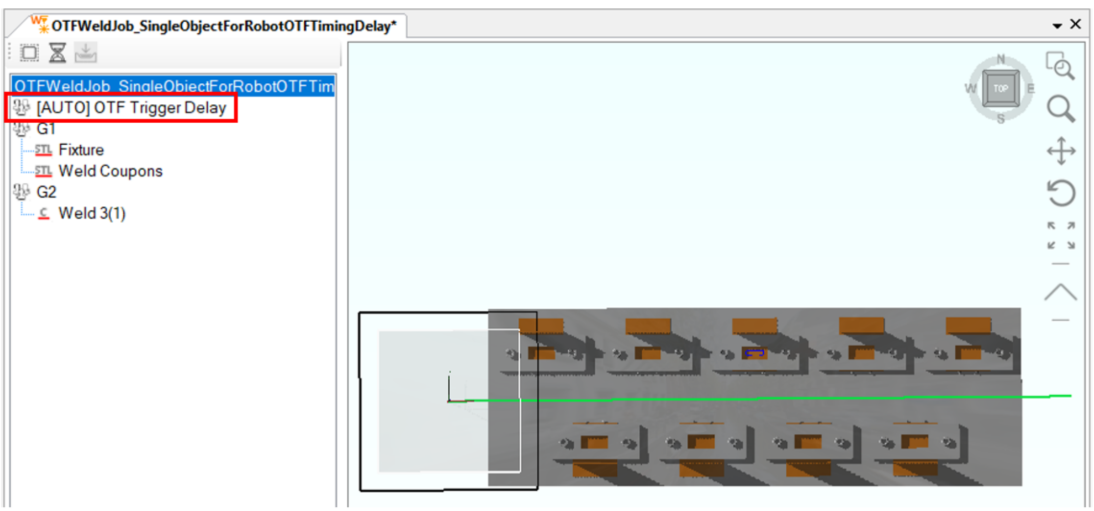

5. Perform the following calculation to determine the proper OTF Trigger Delay time.
   a. $\frac{Measured\ Offset\ [mm]}{Robot\ Travel\ Speed\ [\frac{mm}{sec}]} = OTF\ Trigger\ Delay\ [sec]$
      i. Measured Offset – This is the distance that is measured between the guide laser position and the real laser output.
      ii. Robot Travel Speed – This is the travel speed that the robot was moving at when the real laser output was marked on the part.
      iii. OTF Trigger Delay – This is the calculated value that will be entered into IPGScan.
      iv. Example: $\frac{3.5\ mm}{150\ mm/sec} = .023\ sec$
6. Enter the calculated OTF Trigger Delay time for the given OTF Job. See Figure 12-112.

*Figure 12-112 Entering an OTF Trigger Delay for a OTF Job*




a. Once the OTF Trigger Delay is entered, users will notice that the robot trajectory is a different color and that a Group is automatically generated in the Job Tree. See Figure 12-113.

*Figure 12-113 Auto Generated Group and Trajectory Color Change*



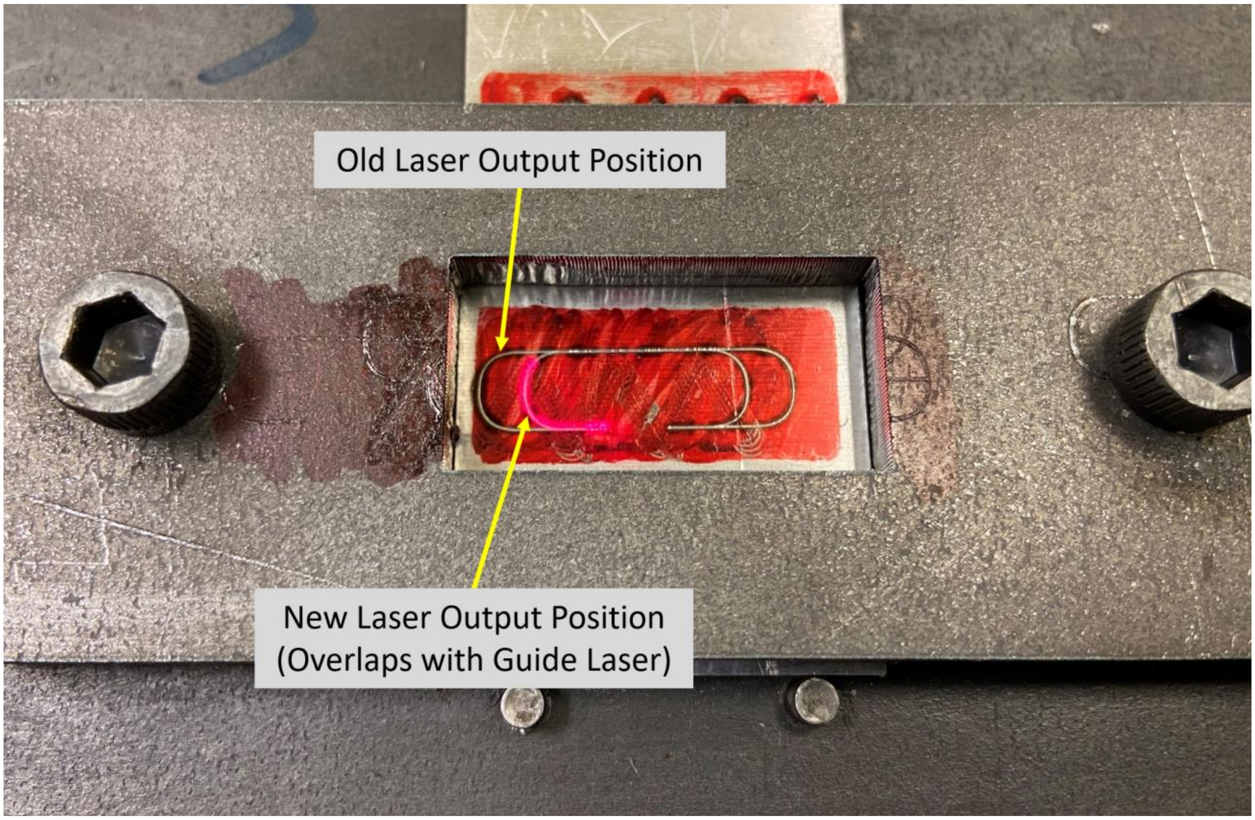

7. With the OTF Trigger Delay added, cycle the process just as it was done in step 2.

| | |
|---|---|
| **IMPORTANT** | **No robot program changes should have been made from step 2 to step 7. Any changes that may have been made could affect the process output and may result in unexpected results.** |

8. Once the process completes, reposition the head to preview the object placement with the guide laser (as outlined in step 1).
   a. Here users should now notice that the new laser output is closer to the guide laser position or overlapping the guide laser position (see Figure 12-114). If some offset still

exists, users can continue to modify the OTF Trigger Delay value to further dial in the process.

*Figure 12-114 Laser Ouput Now Aligns with the Guide Laser Position*



# 13 Coordinated Motion Processing (Non-Robotic On-The-Fly)

## 13.1 Overview

Coordinated Motion Processing means to process a moving target without having the target stop for the realignment of the object. It also can entail processing an object while the scanner is in motion and the target is stationary. This method of processing is also commonly referred to as Mark On-The-Fly, Coordinated Stage Motion, or On-The-Fly (Non-Robotic) processing.

When using a real encoder, the encoder's resolution (meters/encoder pulse) should be at equal to or less than the laser beam diameter. IPG's software also provides a simulated encoder input so a real encoder may not be needed for some applications.

### 13.1.1 Requirements/Recommendations

The following list details requirements and recommendations for Coordinated Motion Processing.

- The Stage Configuration Utility software must be installed.
  - See section "Appendix - Stage Configuration Utility" for additional utility details.
- Encoder must have a quadrature output.
- Encoder reset is an option.
- The use of the Motion Interface is required.
  - Please refer to the "External Interface User Guide (DOCOXUGSCNXX0002)" for additional information concerning this interface.

## 13.2 Configuration Parameters

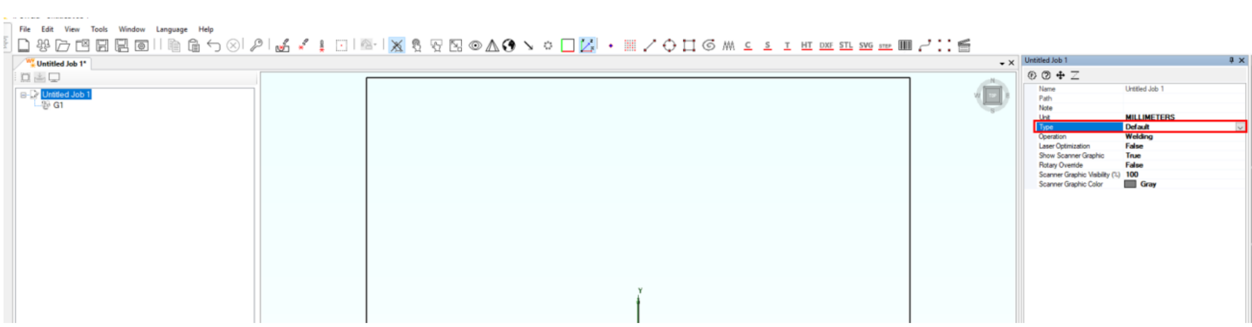
For information regarding hardware connections and the Stage Configuration Utility, please refer to the "Scanner Series User Guide (PN-21-010211)" and "External Interface User Guide (DOCOXUGSCNXX0002)." For Stage Configuration Utility details, see section "Appendix - Stage Configuration Utility."

## 13.3 Job Creation

The following steps outline how to create an example Coordinated Motion job in IPGScan.

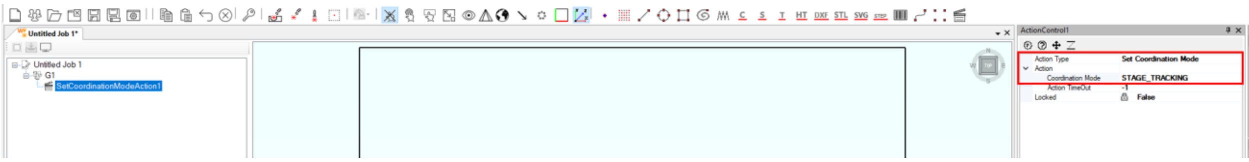
1. Create an IPGScan job of "Default" Type. See Figure 13-1.

*Figure 13-1 Creating a Default Type Job*



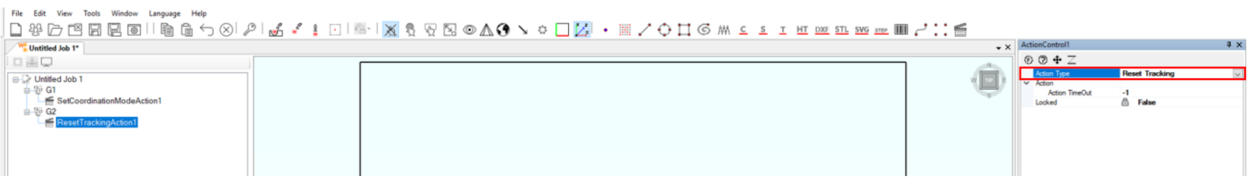
2. Create an Action Control in Group 1.
3. Configure the Action Control to be of "Set Coordination Mode" type. Coordination Mode should be set to "STAGE_TRACKING." See Figure 13-2.

*Figure 13-2 Set Coordination Mode Action Control*



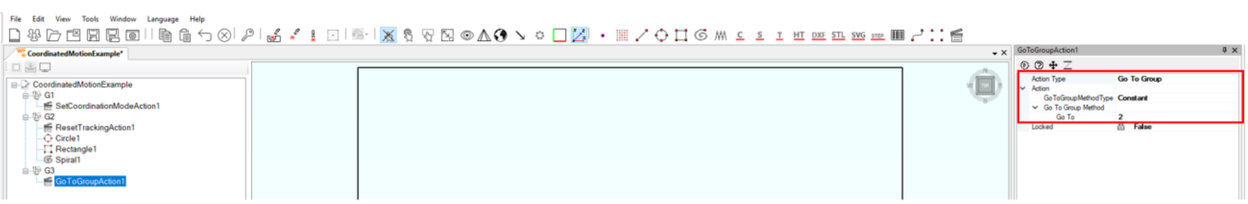
4. Create a second Group.
5. Within Group 2, create an Action Control and configure it to be of “Reset Tracking” type. See Figure 13-3.

*Figure 13-3 Reset Tracking Action Control*



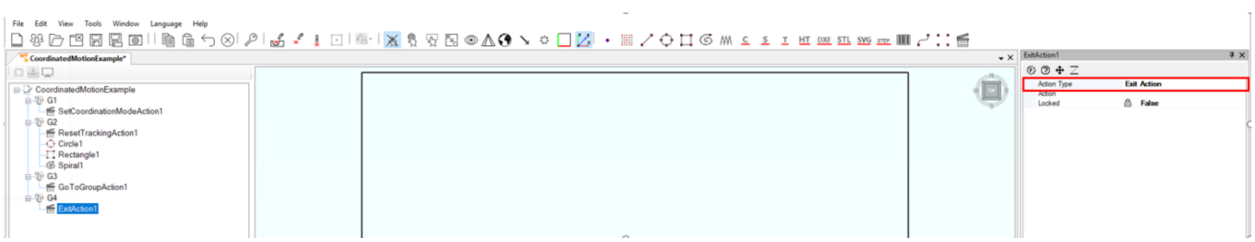
6. Create any desired shapes and place throughout the Field of View.
7. Create a third Group.
8. Within Group 3, create an Action Control and configure it to be a “Go To Group” type. The Method Type should be set to “Constant” and the Go To value should be the Group ID of the Group containing the Reset Tracking Action Control and the Process Objects (G2 in this example). See Figure 13-4.

*Figure 13-4 Go To Group Action Control*



9. Create a fourth Group.
10. Within Group 4, create an Action Control and configure it to be a “Exit Action.” This action allows IPGScan to complete the loop. See Figure 13-5.

*Figure 13-5 Exit Action Control*



Assuming all other setup has been complete, users can no Start Processing and run the drive.

# 14 Error Codes

The following are error codes that may be encountered when using IPGScan and their descriptions are listed below.

- “SPK_HARDWARE_STOP” or “Hardware Stop”
  - Scanner external stop is active. In other words the ENABLE bit is set inactive on the utilized External Interface device. Please refer to the “External Interface User Guide (DOCOXUGSCNXX0002)” for appropriate interface pinouts. This is not a hardware safety stop.
- “SPK_ROBOT_MISSING_TRAJ”
  - No trajectory is loaded in an OTF job.
- “SPK_ROBOT_TIMEOUT”
  - The OTF job failed Dryrun. Ensure that object positions fall within the InView Win- dow, check if robot and vector weld speeds may need to be modified, and ensure that features are being processed in the right direction.
- “SPK_INVALID_POSITION”
  - Feature is outside of the field of view/processing window. Realign feature inside window to fix this error. If feature is within the window, but this error is still shown, check the z-position of the feature.
- “SPK_OUTPUT_SOFTWARE_ABORT_ERROR”
  - Current process was aborted.
- “SPK_INVALID_INVIEW”
  - CoordinationParams.xml file is missing.

# 15 Service and Support

There are no operator serviceable parts inside. Please refer all servicing to qualified IPG personnel.

## 15.1 Technical Support

Many issues and questions regarding the safety, set-up, operation, and maintenance of IPG products can be resolved by carefully reading this User Guide. However, if you have questions regarding the safety, set-up, operation, or maintenance of your IPG product, call the IPG Photonics Customer Service department located in Oxford, Massachusetts, USA, at 508- 373-1100.

If issues cannot be resolved by using this User Guide or over the telephone with the IPG Technical Support Group, it may be necessary to return the product to IPG. Refer to section 17 for more details.

# 16 Warranty

## 16.1 Limited Express Product Warranties

IPG warrants to the original Buyer or, if Buyer is an authorized IPG reseller or distributor, to Buyer's original customer of the IPG Product, that the IPG Product conforms to applicable IPG Product specifications and is free from defects in materials and workmanship. These non- transferable warranties start on the shipment date from IPG (or other date specifically referencing the warranty start date in IPG's sales order/order acknowledgment), and continue until the end of the warranty period listed in IPG's sales order/order acknowledgment. If there is no warranty period listed, then warranty period is one year. Products or major components manufactured by parties other than IPG bear the original manufacturer's warranty and warranty period. The obligations of IPG are limited to the repair or replacement (at IPG's option) of any Product that does not meet the IPG warranty during the warranty period. IPG warrants repaired or replaced Products under warranty only for the remaining un-expired period of time in the original warranty. IPG reserves the right to issue a credit note for any defective Products that have proved defective through normal usage; Buyer debit memos are not allowed. This warranty governs over any conflicting terms in Buyer's purchase order or other IPG documents except as expressly provided herein.

## 16.2 Warranty Limitations

This warranty excludes and does not cover defects or damage resulting from any of the following: contamination of external optical surfaces; unauthorized modification, misuse or mishandling, disassembly or opening, neglect, or damage from accident; operation outside environmental specifications or product ratings; user software or interfacing; components and accessories manufactured by companies other than IPG, which have separate warranties; improper or inadequate installation, site preparation or maintenance; or failure to follow information and precautions contained in the operating manual. Additional warranty exceptions, limitations and exclusions may apply for laser systems manufactured by IPG and its affiliates as set forth in the applicable quotation and sales order/order acknowledgment. All products or components (including software) identified as experimental, prototypes or to be used in field trials are not warranted and are provided to the Buyer on an “as is” basis. IPG assumes no responsibility for Buyer or third-party supplied material, components, systems or equipment. Products and repaired Products may contain components that have been previously used in other products, however such Products meet IPG Product specifications for newly manufactured Products. The Buyer must give prompt notification to IPG of any claim under the warranty in writing. IPG has no responsibility for warranty claims more than 30 days after the Buyer discovers or becomes aware of the claimed defect. Buyer is responsible for providing appropriate utilities and operating environment as stated in the operating manual and the specifications. This warranty applies only to the original Buyer at the initial installation or delivery point. Buyer must make all claims under this warranty and no claim will be accepted from any third party.

**EXCEPT FOR THE LIMITED WARRANTIES EXPRESSLY SET FORTH ABOVE, IPG SPE- CIFICALLY DISCLAIMS ANY AND ALL OTHER WARRANTIES AND REPRESENTATIONS TO BUYER, INCLUDING WITHOUT LIMITATION, ANY AND ALL IMPLIED WARRANTIES, SUCH AS FREEDOM FROM INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.**

## 16.3 Limitation of Remedies and Liabilities

**THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. IN NO EVENT SHALL IPG BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY OR PUNITIVE DAMAGES (EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES) ARISING FROM OR RELATING TO THIS ORDER OR THE PRODUCTS (INCLUDING, LOSS OF PROFITS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY. IPG'S MAXIMUM LIABILITY ARISING UNDER THESE TERMS AND CONDITIONS WILL NOT EXCEED, IN THE AGGREGATE, THE TOTAL AMOUNT PAID FOR THE PRODUCTS BY BUYER. THESE LIMITATIONS MAY NOT APPLY TO YOU UNDER THE LAWS OF CERTAIN JURISDICTIONS.**

## 16.4 Software

### 16.4.1 Firmware License Agreement

**CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THIS PACKAGE OR SIGNIFYING YOUR ACCEPTANCE BY CLICKING THE APPROPRI- ATE DIALOG BOX. OPERATING THE PRODUCT, CLICKING THE APPROPRIATE DIALOG BOX OR USING ANY PART OF THE SOFTWARE SIGNIFIES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, PROMPTLY RETURN THE PRODUCT UNUSED ALONG WITH ANY OTHER RELATED ITEMS THAT WERE INCLUDED IN THE SAME ORDER FOR FULL CREDIT.**

You, as the Customer, agree as follows:

**1. DEFINITIONS**

“IPG” shall mean the IPG Photonics Corporation affiliate providing Licensed Software to Customer pursuant to this Agreement.

“IPG Software” shall mean those portions of the Licensed Software owned by IPG or IPG affiliates.

“Licensed Software” shall mean the software, in object code form only, supplied by IPG pursuant to this Agreement.

“Licensed Product” shall mean the Licensed Software and/or its accompanying documentation.

“Third Party Software” shall mean those portions of the Licensed Software owned or licensed by a third party, including but not limited to operating system code, that is embedded within the Licensed Software.

**2. LICENSE**

2.1 Except as provided in section 2.2 below, you are granted a non-transferable, nonexclusive license to use the Licensed Software only as embedded in or to be used on a single IPG product. You may copy the Licensed Product, for backup purposes only, in support of your use of the Licensed Software, limited to one copy. No other copies shall be made unless authorized in writing by IPG. You must reproduce and include all applicable copyright notices on any copy. You may not reverse compile or otherwise reverse engineer, or modify the Licensed Software. The Licensed Software, comprising proprietary trade secret information of IPG and/ or its licensors, shall be held in confidence by Customer and Customer shall not disclose it to third parties, unless disclosure is required by law or legal process. In response to such legal requirements, the Customer shall promptly notify IPG in advance of such disclosure and reasonably cooperate in attempts to maintain the confidentiality of the Licensed Software. No title to the intellectual property is transferred. Licensed Software shall not be copied, reproduced, or used for any other purpose outside of operation of the IPG product, and shall not be used on any other piece of hardware other than the IPG product with which it was provided.

2.2 If you transfer the IPG product on which the Licensed Software is used, you may transfer the Licensed Software to the end user of the product provided that the end user agrees to be bound by terms no less restrictive than the provisions of this Agreement, and provided that all proprietary markings are maintained. Any other transfer is void and automatically terminates this license. You shall use your best efforts to enforce such agreement and shall promptly report any violation or suspected violation to IPG. In the event you do not enforce such agreement after a breach, you shall, to the extent permissible by applicable law, grant IPG the right to enforce such agreement.

2.3 The Licensed Software may include Third Party Software licensed to IPG in addition to the licenses below. The owner of this Third Party Software (the “Third Party”) and its licensors are intended third party beneficiaries of this Agreement, and the provisions of this Agreement relating to the Licensed Software, as the same incorporates Third Party Software, are made expressly for the benefit of, and are enforceable by, the Third Party and its licensors. The Third Party and its licensors retain ownership of all copies of the Third Party Software. The Licensed Software is warranted by IPG in accordance the above Warranty and the Third Party does not provide an additional warranty. **All Third Party Software included in the Licensed Software is provided “AS IS” without warranty from the Third Party, and each Third Party disclaims all warranties, either express or implied, including but not limited to the implied warranties of merchantability, title, non-infringement or fitness for a particular purpose with regard to the Third Party Software. The Third Party shall not have any liability for special, indirect, punitive, incidental or consequential damages.**

2.4 In addition to the IPG Software licensed above, IPG may provide certain files embedded in or to be used on the IPG hardware product which may be subject to the terms of the GNU General Public License (GPL) or the GNU Lesser General Public License (LGPL), the current text of which may be found at: http://www.gnu.org, or another open source license. The IPG Software is proprietary software not subject to the GPL or LGPL or other open source license, and Customer has no license to take any action, and shall take no action, which would have the effect of subjecting the IPG Software or any portion of the IPG Software to the terms of the GPL or LGPL or other open source license. Customer may consult the user documentation for identifications and further information.

2.5 For the rights granted in this Agreement, Customer shall pay to IPG the price for the IPG hardware product in which the Licensed Software is embedded.

2.6 You understand that IPG may, at any time and in its sole discretion, update or modify the Licensed Product or discontinue updating and/or supporting the Licensed Product. In the case of an update or modification, you agree to allow IPG to automatically install the Licensed Software on the IPG hardware product on which the Licensed Software is used. Any updated or modified Licensed Product made available and/or installed by IPG on the IPG hardware product shall become part of the Licensed Software and subject to this Agreement.

**3. TERM AND TERMINATION**

3.1 You may terminate the license granted hereunder at any time by destroying the Licensed Product together with all copies thereof and notifying IPG in writing that all use of the Licensed Product has ceased and that same has been destroyed.

3.2 IPG may terminate this Agreement or any license hereunder upon notice to Customer if Customer breaches any of the terms and conditions of this Agreement or if Customer attempts to assign this Agreement or any license hereunder without IPG's prior written consent. Within twenty (20) days after any termination of this Agreement, Customer shall certify in writing to IPG that all use of the Licensed Product has ceased, and that the same has been destroyed.

3.3 All provisions of this Agreement related to disclaimers of warranty, limitation of liability, IPG's intellectual property rights, or export shall survive any expiration or termination and remain in effect. Termination of this Agreement or any license hereunder shall not relieve Customer of its obligation to pay any and all outstanding charges hereunder nor entitle Customer to any refund of such charges previously paid.

## 16.5 Software License Agreement for LaserNet™

If your product includes LaserNet™ Software, the following applies:

***IPG Laser GmbH®***

**Single Use License**

**PLEASE READ THIS SOFTWARE LICENSE AGREEMENT ("LICENSE") CAREFULLY BEFORE USING THE SOFTWARE OR THE EQUIPMENT. BY CLICKING ON THE "ACCEPT" BUTTON, USING THIS SOFTWARE, OR USING THE EQUIPMENT THAT CON- TAINS THIS SOFTWARE, YOU ARE CONSENTING TO BE BOUND BY THIS AGREE- MENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, CLICK THE "DO NOT ACCEPT" BUTTON AND THE INSTALLATION PROCESS WILL NOT CON- TINUE, RETURN THE PRODUCT TO THE MANUFACTURER.**

1. **General**. The software, documentation and any fonts accompanying this License whether on disk, in read only memory, on any other media or in any other form (collectively the "IPG Software") are licensed, not sold, to you by IPG Laser GmbH and its affiliates ("IPG") for use only under the terms of this License, and IPG reserves all rights not expressly granted to you. The rights granted herein are limited to IPG's and its licensors' intellectual property rights in the IPG Software and do not include any other patents or intellectual property rights. You own the media on which the IPG Soft- ware is recorded but IPG and/or IPG's licensor(s) retain ownership of the IPG Soft- ware itself. The terms of this License will govern any software upgrades provided by IPG that replace and/or supplement the original IPG Software product, unless such upgrade is accompanied by a separate license in which case the terms of that license will govern.
2. **Permitted License Uses and Restrictions**
   a. This License allows you to install and use one copy of the IPG Software on a single computer at a time. This License does not allow the IPG Software to exist on more than one computer at a time, and you may not make the IPG Software available over a network where it could be used by multiple computers at the same time. You may make one copy of the IPG Software in

machine-readable form for backup purposes only; provided that the backup copy must include all copyright or other proprietary notices contained on the original.

*b.* Except as and only to the extent permitted in this License and by applicable law, you may not copy, decompile, reverse engineer, disassemble, modify, or create derivative works of the IPG Software or any part thereof. *THE IPG SOFTWARE IS NOT INTENDED FOR USE IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL SYSTEMS, OR OTHER EQUIPMENT IN WHICH THE MALFUNTION OF THE IPG SOFTWARE WOULD RESULT IN THE FORESEEABLE RISK OF INJURY OR DEATH TO THE OPERATOR OF THE EQUIPMENT OR SYSTEM OR TO OTHERS.*

3. **Transfer**. You may not rent, lease, lend, redistribute or sub-license the IPG Software. You may, however, make a one-time permanent transfer of all of your license rights to the IPG Software (in its original form as provided by IPG) to another party, provided that: (a) the transfer must include all of the IPG Software, including all its component parts, original media, printed materials and this License; (b) you do not retain any copies of the IPG Software, full or partial, including copies stored on a computer or other storage device; and (c) the party receiving the IPG Software reads and agrees to accept the terms and conditions of this License.
4. **Termination**. This License is effective until terminated. Your rights under this License will terminate automatically without notice from IPG if you fail to comply with any term(s) of this License. Upon the termination of this License, you shall cease all use of the IPG Software and destroy all copies, full or partial, of the IPG Software.
5. **Limited Warranty on Media**. IPG warrants the media on which the IPG Software is recorded and delivered by IPG to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of original purchase. Your exclusive remedy under this Section shall be, at IPG's option, replacement of the IPG Software which is returned to IPG or an IPG authorized representative. *THIS LIM- ITED WARRANTY AND ANY IMPLIED WARRANTIES ON THE MEDIA INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, OF SATISFACTORY QUALITY, AND OF FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF ORIGI- NAL PURCHASE*.
6. **Disclaimer of Warranties**. YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT USE OF THE IPG SOFTWARE IS AT YOUR SOLE RISK AND THAT THE ENTIRE RISK AS TO SATISFACTORY QUALITY, PERFORMANCE, ACCURACY AND EFFORT IS WITH YOU. EXCEPT FOR THE LIMITED WARRANTY ON MEDIA SET FORTH ABOVE AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE IPG SOFTWARE IS PROVIDED "AS IS", WITH ALL FAULTS AND WITH- OUT WARRANTY OF ANY KIND TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAW, IPG AND ITS SUPPLIERS FURTHER DISCLAIM ALL WARRAN- TIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, IMPLIED WAR- RANTIES OF TITLE, NON- INFRINGEMENT, ACCURACY, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTIES THAT MAY ARISE FROM COURSE OF DEALING, COURSE OF PERFORMANCE, OR TRADE PRACTICE. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFOR- MANCE OF THE IPG SOFTWARE OR RELATED MATERIALS REMAINS WITH YOU. APPLICABLE LAW MAY NOT ALLOW THE EXCLUSION OF IMPLIED WAR- RANTIES, SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU. IPG MAY MAKE CHANGES TO THE IPG SOFTWARE AND HAS NO OBLIGATION TO DIS- TRIBUTE NEWER VERSIONS.
7. **Limitation of Liability**. NEITHER IPG NOR ANY OF ITS SUPPLIERS SHALL BE RESPONSIBLE OR LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, EXEMPLARY, PUNITIVE OR OTHER

DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS, LOSS OF DATA OR LOST PROFITS), UNDER ANY CONTRACT, NEGLIGENCE, STRICT LIABILITY OR OTHER THEORY ARISING OUT OF OR RELATING IN ANY WAY TO THE IPG SOFTWARE EVEN IF IPG OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. YOUR SOLE REMEDY FOR DISSATISFAC- TION WITH THE IPG SOFTWARE IS TO STOP USING THE IPG SOFTWARE. THE SOLE AND EXCLUSIVE MAXIMUM LIABILITY TO IPG FOR ALL DAMAGES, LOSSES AND CAUSES OF ACTION, WHETHER IN CONTRACT, TORT (INCLUD- ING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE, SHALL BE THE TOTAL AMOUNT PAID BY YOU, IF ANY, FOR USE OF THE IPG SOFTWARE. THESE LIMITATIONS MAY NOT APPLY TO YOU UNDER THE LAWS OF CERTAIN JURISDICTIONS.

The foregoing limitations will apply even if the above stated remedy fails of its essential purpose.

8. **Export Law Assurances**. You acknowledge that IPG Software is subject to U.S. and European Union export jurisdiction. You agree to comply with all applicable international and national laws that apply to the IPG Software, including the U.S. Export Administration Regulations, as well as end-user, end-use and destination restrictions issued by U.S. and other governments.
9. **Government End Users.** The IPG Software and related documentation are “Commercial Items”, as that term is defined at 48 C.F.R. §2.101, consisting of “Commercial Computer Software” and “Commercial Computer Software Documentation”, as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States.
10. **Controlling Law and Severability**. This License will be governed by and construed in accordance with the laws of the State of Massachusetts, as applied to agreements entered into and to be performed entirely within Massachusetts between Massachusetts residents. This License shall not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded. If for any reason a court of competent jurisdiction finds any provision, or portion thereof, to be unenforceable, the remainder of this License shall continue in full force and effect.
11. **Complete Agreement; Governing Language**. This License constitutes the entire agreement between the parties with respect to the use of the IPG Software licensed hereunder and supersedes all prior or contemporaneous understandings regarding such subject matter. No amendment to or modification of this License will be binding unless in writing and signed by IPG. Any translation of this License is done for local requirements and in the event of a dispute between the English and any non-English versions, the English version of this License shall govern.

## 16.6 Microsoft Corporation Embedded Software End User License Agreement

If your product includes Microsoft Corporation embedded software, then the following applies:

You have acquired a device (“EQUIPMENT”) that includes software licensed by IPG Photonics Corporation or its affiliates (collectively, "IPG") from an affiliate of Microsoft Corporation (“MS”). Those installed software products of MS origin, as well as associated media, printed materials, and “online” or electronic documentation (“SOFTWARE”) are protected by international intellectual property laws and treaties. IPG, MS and its suppliers (including Microsoft Corporation) own the title, copyright, and other intellectual property rights in the SOFTWARE. The SOFTWARE is licensed, not sold. All rights reserved.

IF YOU DO NOT AGREE TO THIS END USER LICENSE AGREEMENT ("EULA"), DO NOT USE THE EQUIPMENT OR COPY THE SOFTWARE. INSTEAD, PROMPTLY CONTACT IPG FOR INSTRUCTIONS ON RETURN OF THE UNUSED EQUIPMENT(S) FOR A REFUND.

**ANY USE OF THE SOFTWARE, INCLUDING BUT NOT LIMITED TO USE OF THE EQUIPMENT, WILL CONSTITUTE YOUR AGREEMENT TO THIS EULA (OR RATIFICATION OF ANY PREVIOUS CONSENT).**

This EULA is valid and grants the end-user rights ONLY if the SOFTWARE is genuine and a genuine Certificate of Authenticity for the SOFTWARE is included. For more information on identifying whether your software is genuine, please see http://www.microsoft.com/piracy/howtotell.

GRANT OF SOFTWARE LICENSE. This EULA grants you the following license:

1. You may use the SOFTWARE only on the EQUIPMENT.
2. Restricted Functionality. You are licensed to use the SOFTWARE to provide only the limited functionality (specific tasks or processes) for which the EQUIPMENT has been designed and marketed by IPG. This license specifically prohibits any other use of the software programs or functions, or inclusion of additional software programs or functions that do not directly support the limited functionality on the EQUIPMENT. Notwithstanding the foregoing, you may install or enable on a EQUIPMENT, systems utilities, resource management or similar software solely for the purpose of administration, performance enhancement and/or preventive maintenance of the EQUIPMENT.
3. If you use the EQUIPMENT to access or utilize the services or functionality of Micro- soft Windows Server products (such as Microsoft Windows Server 2003), or use the EQUIPMENT to permit workstation or computing devices to access or utilize the services or functionality of Microsoft Windows Server products, you may be required to obtain a ClientAccess License for the EQUIPMENT and/or each such workstation or computing device. Please refer to the end user license agreement for your Microsoft Windows Server product for additional information.
4. **NOT FAULT TOLERANT**. THE SOFTWARE IS NOT FAULT TOLERANT. IPG HAS INDEPENDENTLY DETERMINED HOW TO USE THE SOFTWARE IN THE EQUIP- MENT, AND MS HAS RELIED UPON IPG TO CONDUCT SUFFICIENT TESTING TO DETERMINE THAT THE SOFTWARE IS SUITABLE FOR SUCH USE.
5. **NO WARRANTIES FOR THE SOFTWARE**. THE SOFTWARE is provided “AS IS” and with all faults. THE ENTIRE RISK AS TO SATISFACTORY QUALITY, PERFOR- MANCE, ACCURACY, AND EFFORT (INCLUDING LACK OF NEGLIGENCE) IS WITH YOU. ALSO, THERE IS NO WARRANTY AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE SOFTWARE OR AGAINST INFRINGEMENT. IF YOU HAVE RECEIVED ANY WARRANTIES REGARDING THE EQUIPMENT OR THE SOFTWARE, THOSE WARRANTIES DO NOT ORIGINATE FROM, AND ARE NOT BINDING ON, MS.
6. **No Liability for Certain Damages**. EXCEPT AS PROHIBITED BY LAW, MS SHALL HAVE NO LIABILITY FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL OR INCI- DENTAL DAMAGES ARISING FROM OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE SOFTWARE. THIS LIMITATION SHALL APPLY EVEN IF ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE. IN NO EVENT SHALL MS BE LIABLE FOR ANY AMOUNT IN EXCESS OF U.S. TWO HUNDRED FIFTY DOLLARS (U.S.$250.00).
7. **Restricted Uses**. The SOFTWARE is not designed or intended for use or resale in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft

navigation or communication systems, air traffic control, or other devices or systems in which a malfunction of the SOFTWARE would result in foreseeable risk of injury or death to the operator of the equipment or system, or to others.

8. **Limitations on Reverse Engineering, Decompilation, and Disassembly**. You may not reverse engineer, decompile, or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
9. **SOFTWARE as a Component of the Equipment-Transfer**. This license may not be shared, transferred to or used concurrently on different computers. The SOFTWARE is licensed with the EQUIPMENT as a single integrated product and may only be used with the EQUIPMENT. If the SOFTWARE is not accompanied by EQUIPMENT, you may not use the SOFTWARE. You may permanently transfer all of your rights under this EULA only as part of a permanent sale or transfer of the EQUIPMENT, provided you retain no copies of the SOFTWARE. If the SOFTWARE is an upgrade, any transfer must also include all prior versions of the SOFTWARE. This transfer must also include the Certificate of Authenticity label. The transfer may not be an indirect transfer, such as a consignment. Prior to the transfer, the end user receiving the SOFT- WARE must agree to all the EULA terms.
10. **Consent to Use of Data**. You agree that MS, Microsoft Corporation and their affiliates may collect and use technical information gathered in any manner as part of product support services related to the SOFTWARE. MS, Microsoft Corporation and their affiliates may use this information solely to improve their products or to provide customized services or technologies to you. MS, Microsoft Corporation and their affiliates may disclose this information to others, but not in a form that personally identifies you.
11. **Internet Gaming/Update Features**. If the SOFTWARE provides, and you choose to utilize, the Internet gaming or update features within the SOFTWARE, it is necessary to use certain computer system, hardware, and software information to implement the features. By using these features, you explicitly authorize MS, Microsoft Corporation and/or their designated agent to use this information solely to improve their products or to provide customized services or technologies to you. MS or Microsoft Corporation may disclose this information to others, but not in a form that personally identifies you.
12. **Internet-Based Services Components**. The SOFTWARE may contain components that enable and facilitate the use of certain Internet-based services. You acknowledge and agree that MS, Microsoft Corporation or their affiliates may automatically check the version of the SOFTWARE and/or its components that you are utilizing and may provide upgrades or supplements to the SOFTWARE that may be automatically downloaded to your EQUIPMENT. Microsoft Corporation or their affiliates do not use these features to collect any information that will be used to identify you or contact you. For more information about these features, please see the privacy statement at http://go.microsoft.com/fwlink/?LinkId=25243.
13. **Links to Third Party Sites**. You may link to third party sites through the use of the SOFTWARE. The third party sites are not under the control of MS or Microsoft Corporation, and MS or Microsoft are not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. MS or Microsoft Corporation is not responsible for webcasting or any other form of transmission received from any third party sites. MS or Microsoft Corporation are providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by MS or Microsoft Corporation of the third party site.

14. **Notice Regarding Security**. To help protect against breaches of security and malicious software, periodically back up your data and system information, use security features such as firewalls, and install and use security updates.
15. **No Rental/Commercial Hosting**. You may not rent, lease, lend or provide commercial hosting services with the SOFTWARE to others.
16. **Separation of Components**. The SOFTWARE is licensed as a single product. Its component parts may not be separated for use on more than one computer.
17. **Additional Software/Services**. This EULA applies to updates, supplements, add-on components, product support services, or Internet-based services components ("Supplemental Components"), of the SOFTWARE that you may obtain from IPG, MS, Microsoft Corporation or their subsidiaries after the date you obtain your initial copy of the SOFTWARE, unless you accept updated terms or another agreement governs. If other terms are not provided along with such Supplemental Components and the Supplemental Components are provided to you by MS, Microsoft Corporation or their subsidiaries then you will be licensed by such entity under the same terms and conditions of this EULA, except that (i) MS, Microsoft Corporation or their subsidiaries providing the Supplemental Components will be the licensor with respect to such Supplemental Components in lieu of the "COMPANY" for the purposes of the EULA, and (ii) TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SUPPLEMENTAL COMPONENTS AND ANY (IF ANY) SUPPORT SERVICES RELATED TO THE SUP- PLEMENTAL COMPONENTS ARE PROVIDED AS IS AND WITH ALL FAULTS. ALL OTHER DISCLAIMERS, LIMITATION OF DAMAGES, AND SPECIAL PROVISIONS PROVIDED BELOW AND/OR OTHERWISE WITH THE SOFTWARE SHALL APPLY TO SUCH SUPPLEMENTAL COMPONENTS. MS, Microsoft Corporation or their subsidiaries reserve the right to discontinue any Internet-based services provided to you or made available to you through the use of the SOFTWARE.
18. **Recovery Media**. If SOFTWARE is provided by IPG on separate media and labeled "Recovery Media" you may use the Recovery Media solely to restore or reinstall the SOFTWARE originally installed on the EQUIPMENT.
19. **Backup Copy**. You may make one (1) backup copy of the SOFTWARE. You may use this backup copy solely for your archival purposes and to reinstall the SOFTWARE on the EQUIPMENT. Except as expressly provided in this EULA or by local law, you may not otherwise make copies of the SOFTWARE, including the printed materials accompanying the SOFTWARE. You may not loan, rent, lend or otherwise transfer the backup copy to another user.
20. **End User Proof of License**. If you acquired the SOFTWARE on a EQUIPMENT, or on a compact disc or other media, a genuine Microsoft "Proof of License"/Certificate of Authenticity label with a genuine copy of the SOFTWARE identifies a licensed copy of the SOFTWARE. To be valid, the label must be affixed to the EQUIPMENT, or appear on IPG's software packaging. If you receive the label separately other than from IPG, it is invalid. You should keep the label on the EQUIPMENT or packaging to prove that you are licensed to use the SOFTWARE.
21. **Product Support**. Product support for the SOFTWARE is not provided by MS, Microsoft Corporation, or their affiliates or subsidiaries. For product support, please refer to IPG support number provided in the documentation for the EQUIPMENT. Should you have any questions concerning this EULA, or if you desire to contact IPG for any other reason, please refer to the address provided in the documentation for the EQUIPMENT.

22. **Termination**. Without prejudice to any other rights, IPG may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE and all of its component parts.
23. **EXPORT RESTRICTIONS**. You acknowledge that SOFTWARE is subject to U.S. and European Union export jurisdiction. You agree to comply with all applicable international and national laws that apply to the SOFTWARE, including the U.S. Export Administration Regulations, as well as end-user, end-use and destination restrictions issued by U.S. and other governments. For additional information, see http://www.microsoft.com/exporting.

# 17 Product Returns

## 17.1 Returns to the United States

All product returns require a Return Merchandise Authorization (RMA) from IPG.

To obtain an RMA, call the Customer Service department of IPG Photonics Corporation at 508-373-1100 (US) or +49 2736 44 20 451 (Germany).

If you return a product with a RMA, please perform the following procedure:

1. Products must be carefully packed in a suitable shipping container(s). Buyer assumes all responsibility for products damaged in shipment to IPG.
2. Buyer must issue a purchase order for the value of the replaced parts/service items and IPG will issue credit or invoice when the parts/service is received. Speak to IPG Service Manager for the amount authorized under the required purchase order.
3. All requests for repair or replacement under this warranty must be made to IPG within 30 days after discovery of the defect (but not later than 7 days after warranty expiration).
4. All products returned to IPG but which meet applicable specifications, not defectively manufactured or used not in accordance with this User's Guide, will result in the Buyer being charged IPG's standard examination charge.
5. Complete packing list with product model and serial number will ensure prompt repair.
6. Be sure to include with the returned product your 'ship to' address for the return of the serviced product.

### 17.1.1 Shipping Instructions

**Warranty Returns**

Domestic & International Buyers* pay for one-way freight costs and insurance to IPG. IPG will pay for freight return cost and insurance back to the Buyer.

**Non-Warranty Returns**

Domestic & International Buyers* pay for two-way freight costs and insurance to IPG. If shipment consists of returns that are both warranty and non-warranty, the shipment will be considered as non-warranty.

Shipping address for returns to US:

IPG Photonics Corporation
50 Old Webster Road
Oxford, MA 01540
Attn: Product Returns
Tel: 508-373-1100

| | |
|---|---|
| IMPORTANT | **International Returns must include applicable DUTIES AND TAXES. You must mark air bills with “US GOODS, RETURNED FOR REPAIR.”** |

## 17.2 Returns to Germany

1. IPG Laser GmbH will only accept returns for which an approved Return Material Authorization (RMA) has been issued by IPG Laser GmbH. You should address to the customer support team at +49-(0)2736-44-20-451 or support.europe@ipgphotonics.com to discuss the return and request an RMA number. You must return defective products freight prepaid and insured to IPG Laser at the address shown herein. All products which have returned to IPG Laser but which are found to meet all previously applicable specifications for such products or which indicate damage to the fiber connectors not resulting from defect manufacturing, shall be subject to IPG Laser' standard examination charge in effect at the time and these costs shall be charged to the Buyer. All products returned to IPG Laser which are not accompanied by an itemized statement of defects, shall be returned to the Buyer at the Buyer's expense and IPG Laser shall not carry out any evaluation of such products. IPG Laser warrants to Buyer that its services, labor and replacement parts, assemblies and modules will be free of defects in material and workmanship for ninety (90) days from the date of shipment or performance of services.
2. Warranty Returns - Domestic & *International Buyers should pay for one-way freight costs to IPG Laser. IPG Laser will reimburse Buyers for applicable reasonable third-party freight costs and IPG Laser will pay for freight return cost back to the Buyer.
3. Non-Warranty Returns - Domestic & *International Buyers are responsible for two-way freight costs. If shipment consists of returns that are both warranty and non-warranty, the shipment will be considered as non-warranty. Any UNAUTHORIZED shipments billed to IPG Laser without authorization will be re-invoiced to the Buyer. Confirming purchase orders are required for non-warranty returns.
4. *International Returns must include applicable DUTIES AND TAXES, and you must mark air bills with “RETURNED FOR REPAIR”. In any event, where IPG Laser accepts a shipment, IPG Laser will invoice to the Buyer for any charges as stated above.
5. Returns for credit will not be accepted unless authorized in advance, in writing by IPG Laser, in accordance with IPG Laser' Terms and Condition, including the warranty provisions. In most cases, restocking fees will apply.
6. All returns must be packaged adequately to avoid damage during shipment.
7. Complete packing list with product model and serial number will insure prompt repair, if the other terms of this form are followed.
8. See the IPG Terms and Conditions for the applicable warranty for the products before you request the return of the products.
9. RMA number will expire 31 days after the date of issue. Thereafter, units received in under the expired RMA number will result in a longer turnaround time. Include a copy of the completed RMA form with the return of your unit(s).

### 17.2.1 Shipping Instructions

Shipping address for returns to Germany:

IPG Laser GmbH
Siemensstrasse 7
D-57299 Burbach, Germany
Attn: Product Returns
Tel: +49-(0)2736-44-20-451

# 18 Appendix – Scan Controller Utility

## 18.1 Scan Controller Utility Overview

The primary purpose of the Scan Controller Utility is to provide users with a means of configuring scanners. The configuration and setup of scanners that is performed within this utility is typically performed once when the scanner is initially received and setup, and then not required again. Common tasks that are performed within the Scan Controller Utility include loading LaserSpecification files or calibration files, backing up scanner files, changing the scanners IP address, or assigning a particular name to a scanner. Additional functionality exists within the Scan Controller Utility, but this is often reserved for IPG support.

Figure 18-1 details the layout of the Scan Controller Utility.

*Figure 18-1 Scan Controller Utility Layout Window*

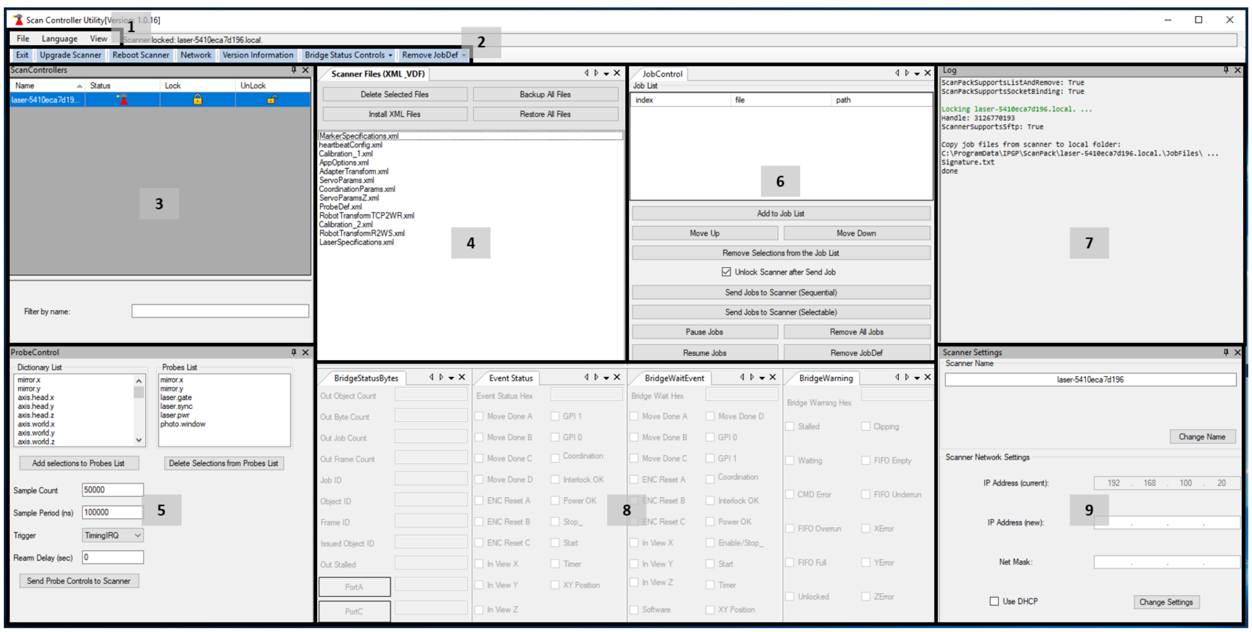



*Table 18-1 Scan Controller Utility Layout Items*

| Number | Description |
|---|---|
| 1 | File Menu |
| 2 | Actions Bar |
| 3 | Scan Controllers Window |
| 4 | Scanner Files Window |
| 5 | Probe Control Window |
| 6 | Job Control Window |
| 7 | Log Window |
| 8 | Bridge Status Window |
| 9 | Scanner Settings Window |

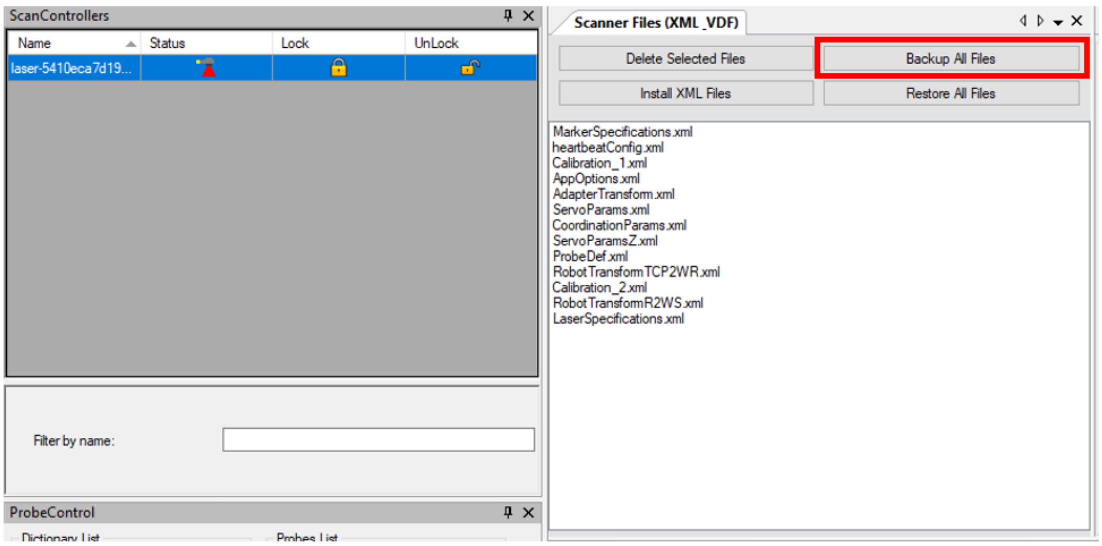
## 18.2 Backing-Up Scan Controller Files

Backing-up Scan Controller files prior to uploading new calibration files is a good operating procedure. In doing so, users can easily access original calibration files during instances where improper files may have been uploaded.

Users can backup Scan Controller calibration files using the following procedure.

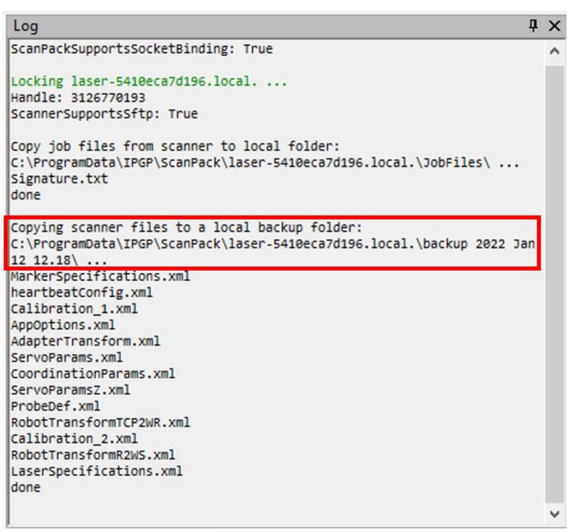
1. Connect to the desired scanner in the Scan Controller Utility.
2. Click “Backup All Files.” See Figure 18-2

*Figure 18-2 Backing-Up Scan Controller Files*




3. Users can then find the backup files in the location specified in the Logs Window. See Figure 18-3.

*Figure 18-3 Backup Location*



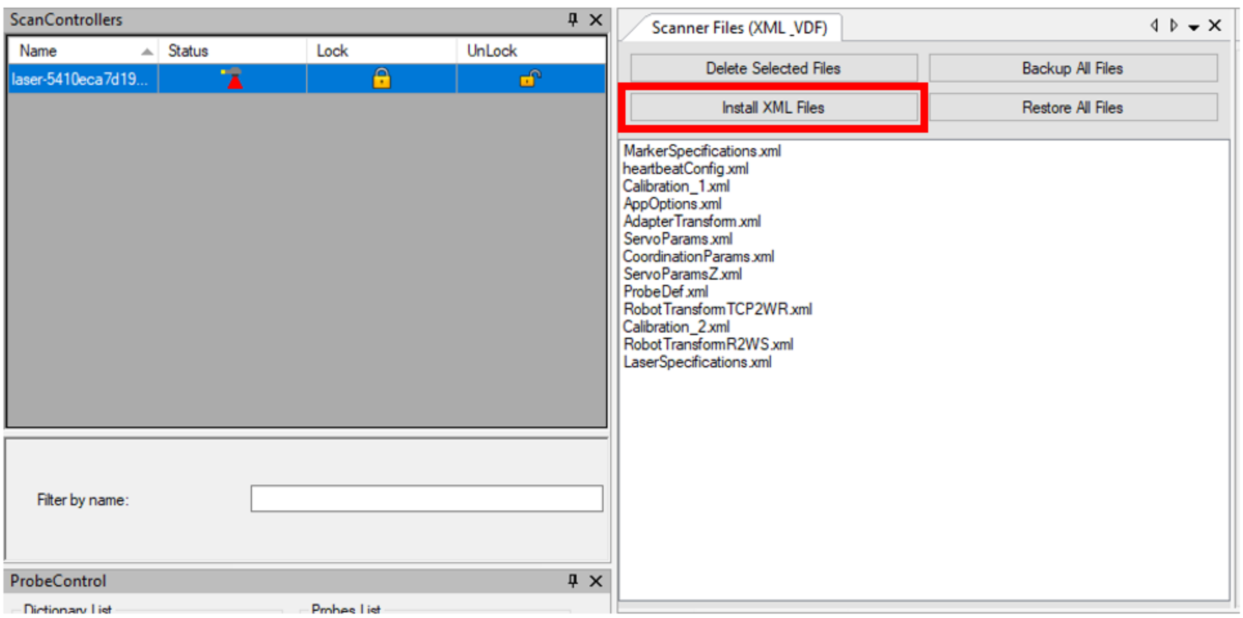
## 18.3 Uploading a LaserSpecification File

IPG makes a wide variety of both lasers and scanners. Because of the variation in laser control across different series of lasers, the scanner needs to know which laser is being used. This is accomplished in the form of a LaserSpecification file, which can be uploaded to the scanner. With the proper LaserSpecification file uploaded, the scanners laser control outputs will then be appropriate given the lasers specifications/capabilities.

The following procedure provides an example of how to upload a LaserSpecification file for a YLS-6000 laser. Users should upload the proper LaserSpecification file that corresponds with the laser purchased.

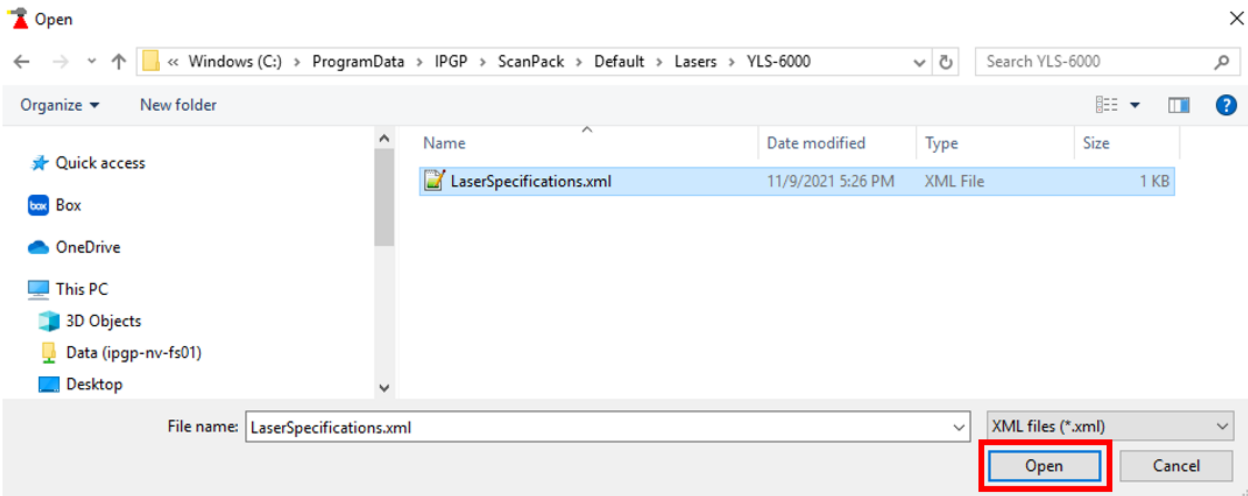
1. Connect to the desired scanner in the Scan Controller Utility.
2. Click "Install XML Files." See Figure 18-4.

*Figure 18-4 Installing a LaserSpecification File*




3. Open the "Default" folder.
4. Open the "Lasers" folder.
5. Open the proper laser folder for the laser that is being used. In this example, the YLS-6000 folder is opened.
6. Select the "LaserSpecifications" file and click "Open." See Figure 18-5.

*Figure 18-5 Selecting and Uploading the LaserSpecification File*




7. Click “OK” to confirm that the install completed.

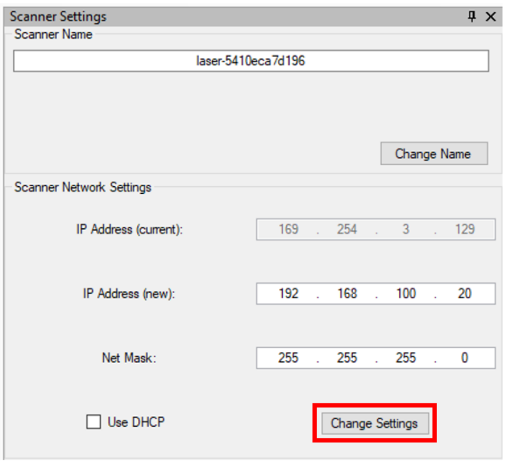
## 18.4 Changing the Scan Controllers IP Address

Scanners can be configured with network settings for DHCP or for a static IP address. By default, scanners ship from IPG with network settings configured for DHCP.

In order to set a static IP address for the scanner, refer to the following procedure.

1. Connect to the desired scanner in the Scan Controller Utility.
2. Using the Scanner Settings Window, enter the desired IP address and Net Mask. See Figure 18-6.

*Figure 18-6 Setting a Static IP Address*

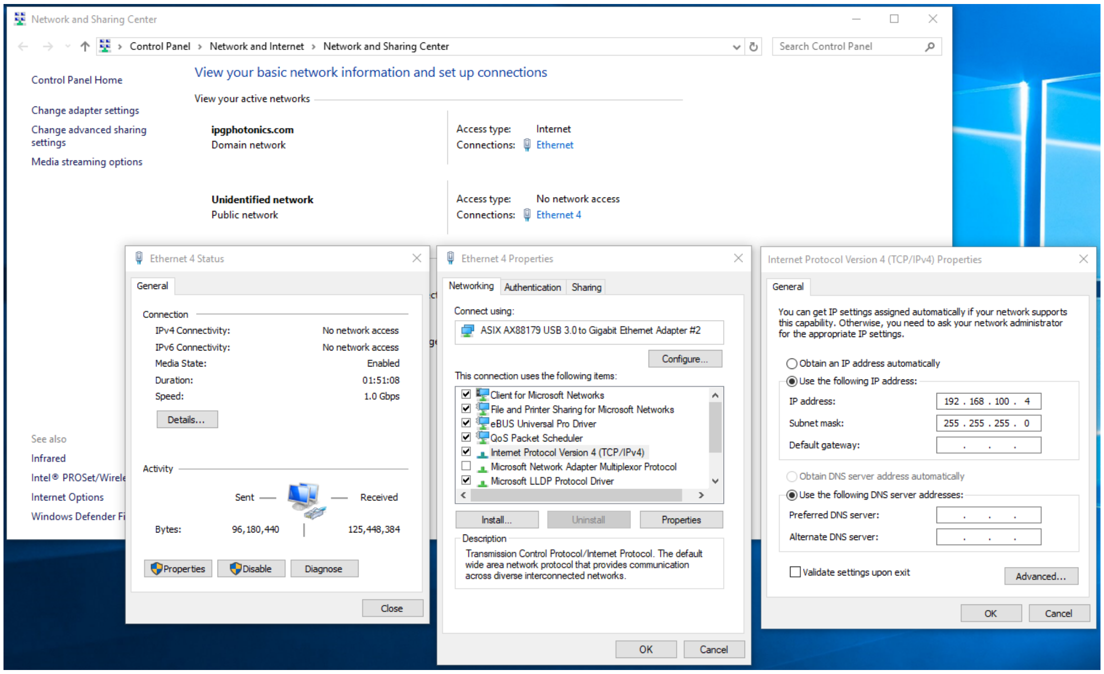



3. Click "Change Settings."

| | |
|---|---|
| **IMPORTANT** | **Allow the scanner sufficient time to change the IP address after clicking "Change Settings." This process could take up to 5 minutes and the scanner will cycle power automatically. Do not power cycle during this time.** |

4. Change the appropriate Local Area Adapter settings on the computer to connect to the scanner. See Figure 18-7.

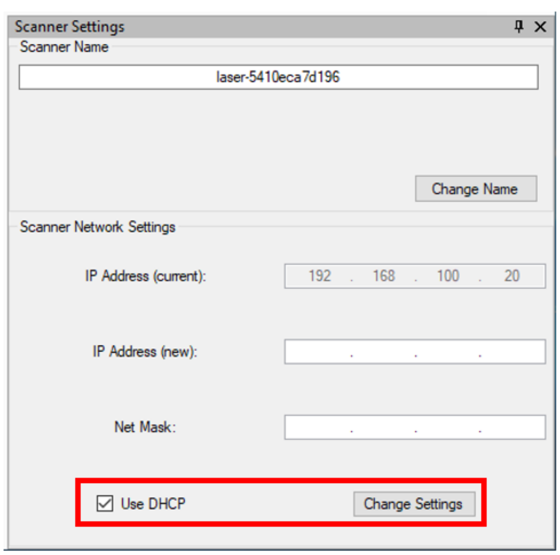
*Figure 18-7 Computer Local Area Adapter Settings*




5. Users should now be able to reconnect to the scanner in the Scan Controller Utility or IPGScan.

If users wish to set the scanner back to DHCP settings, connect to the scanner, check the "Use DHCP" box, and click the "Change Settings" button (see Figure 18-8). Allow the scanner sufficient time to make the change and be sure to set the appropriate Local Area Adapter settings on the PC before attempting to reconnect.

*Figure 18-8 Setting a Scanner to DHCP Settings*



### 18.4.1 Unknown or Forgotten IP Address

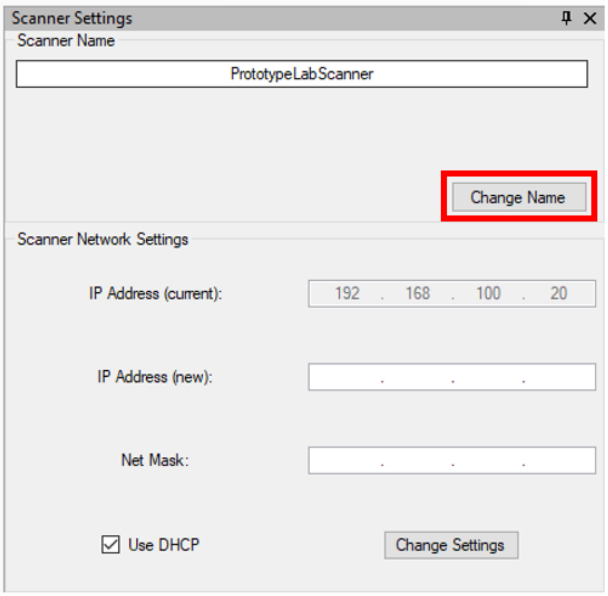
If users do not know what a scanners static IP address is, the serial port on the Scan Controller can be used to reset the scanner back to DHCP settings or to set a new static IP Address. For instruction on how to do this, please refer to the Scanner Series User Guide (P21-010211).

## 18.5 Changing the Scan Controller Name

When users have many scanners in a given facility, it may provide clarity on which scanner is being used by assigning the scanners with a particular name. This can be done using the following procedure.

1. Connect to the desired scanner in the Scan Controller Utility.
2. Enter the desired name of the scanner in the Scanner Settings Window.
3. Click "Change Name" and allow the scanner sufficient time to complete the process. See Figure 18-9.

*Figure 18-9 Changing the Scanner Name*

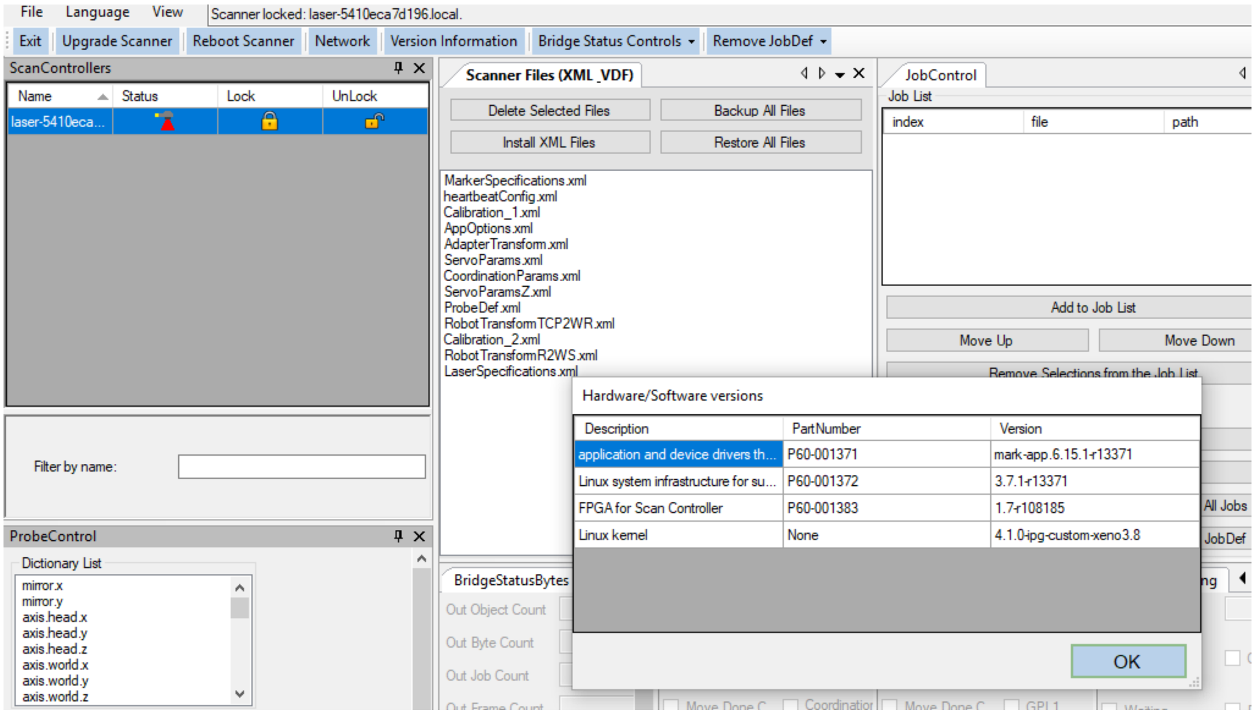


## 18.6 Viewing Scan Controller Firmware Versions

The following procedure outlines how users can view Scan Controller firmware version information.

1. Connect to the desired scanner in the Scan Controller Utility.
2. Click “Version Information” in the Actions Bar.
    a. A pop-up will then appear with the appear with the appropriate information. SeeFigure 18-10.

*Figure 18-10 Viewing Scan Controller Firmware Version Information*



# 19 Appendix – Bug Reporting Utility

## 19.1 Overview

The purpose of the Bug Reporting Utility is to provide users with an easy means of providing IPG support personnel with the necessary information for troubleshooting software bugs and issues.

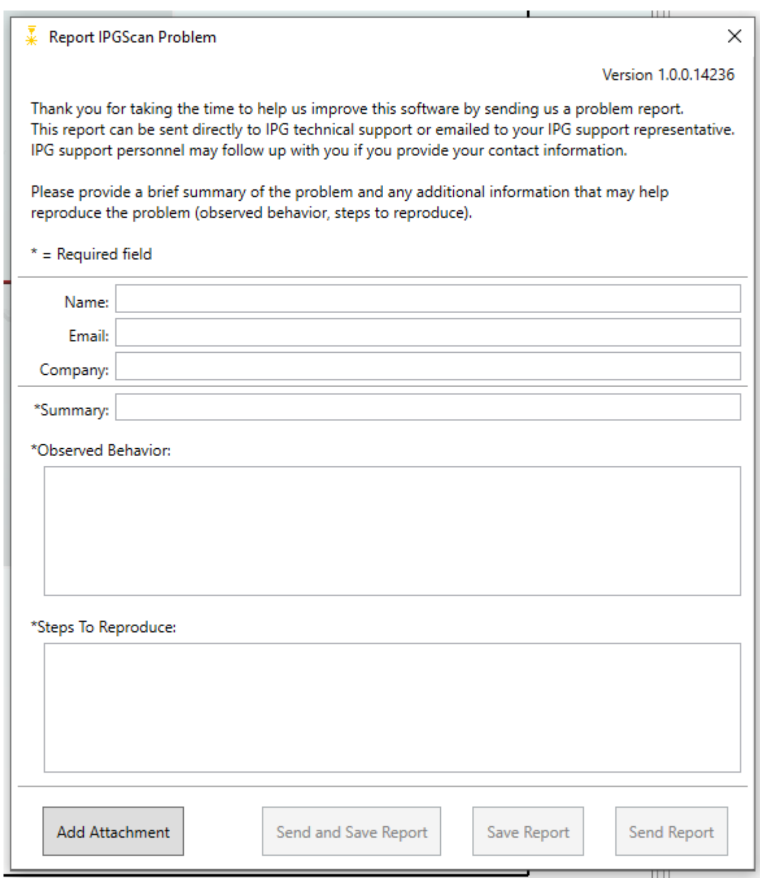
| | |
|---|---|
| **IMPORTANT** | **The Bug Reporting Utility is only intended for supporting IPG developed scanning software (IPGScan and the Scan Controller Utility).**<br><br>**This software is not related to any laser support.**<br><br>**This software is not intended for software feature requests.**<br><br>**If Immediate support is required, please contact IPG Support.** |

## 19.2 Submitting a Software Bug

To submit a bug, please refer to the following procedure.

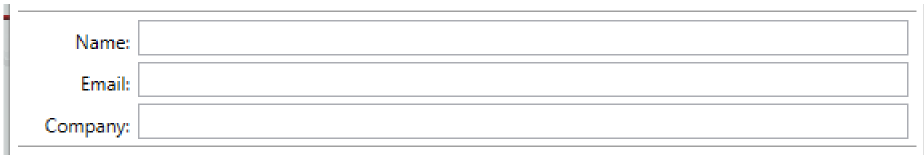
1. Launch the Bug Reporting Utility (also called "Report IPGScan Problem") from the Start Menu or click "Help" and "Report a Problem" in IPGScan. See Figure 19-1.

*Figure 19-1 The Bug Reporting Utility*



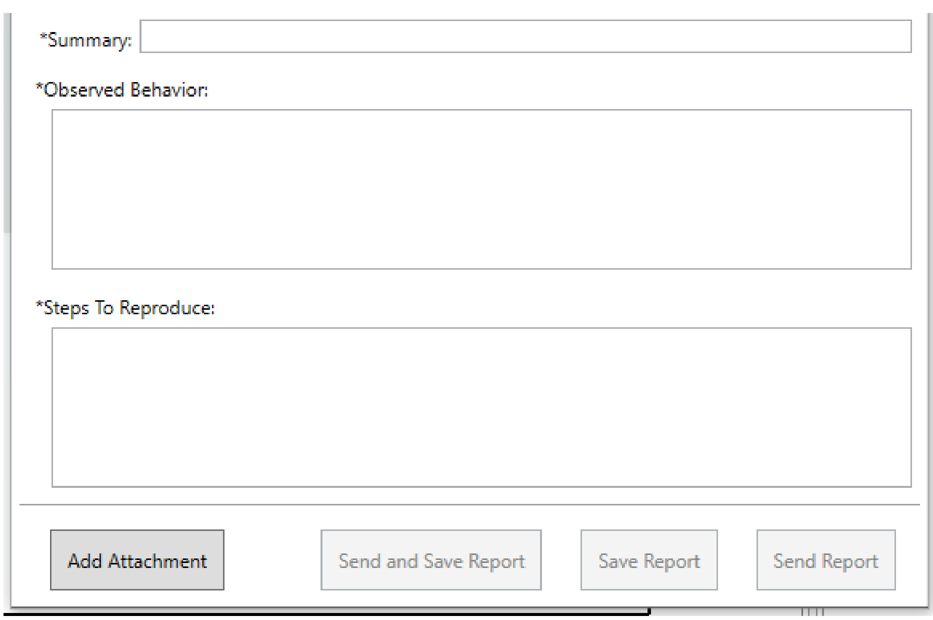
2. Provide an email address, name, and company name. See Figure 19-2.

*Figure 19-2 Email, Name, and Company Name*




3. Provide a summary in the “Summary” field.
    a. An example of a good summary might be, “Unhandled Exception Error Occurs After Clicking Show Projection Volume.”
4. In the “Details” field, provide the following information (see Figure 19-3):
    a. **Observed Behavior** – This should detail what happens when the bug occurs or what the IPG support personnel should be looking for.
        i. Good Example – When I click on the “Show Projection Volume” button, I get a pop-up window that details an unhandled exception error has occurred. Furthermore, the image of the scanner disappears in the IPGScan Canvas.
    b. **Steps to Reproduce** – This should be step by step instructions that help support personnel recreate the bug that is occurring.
        i. Good Example
            1. Open IPGScan Job “Examplebug”
            2. Connect to scanner “laser-5410eca7d196”
            3. Click the job title in the Job Tree and set “Show Scanner Graphic” to True
            4. Click “Show Projection Volume” in the Tool Bar. At this point, the unhandled exception error should occur and the scan head display will disappear from the canvas.

*Figure 19-3 Summary and Details Field*



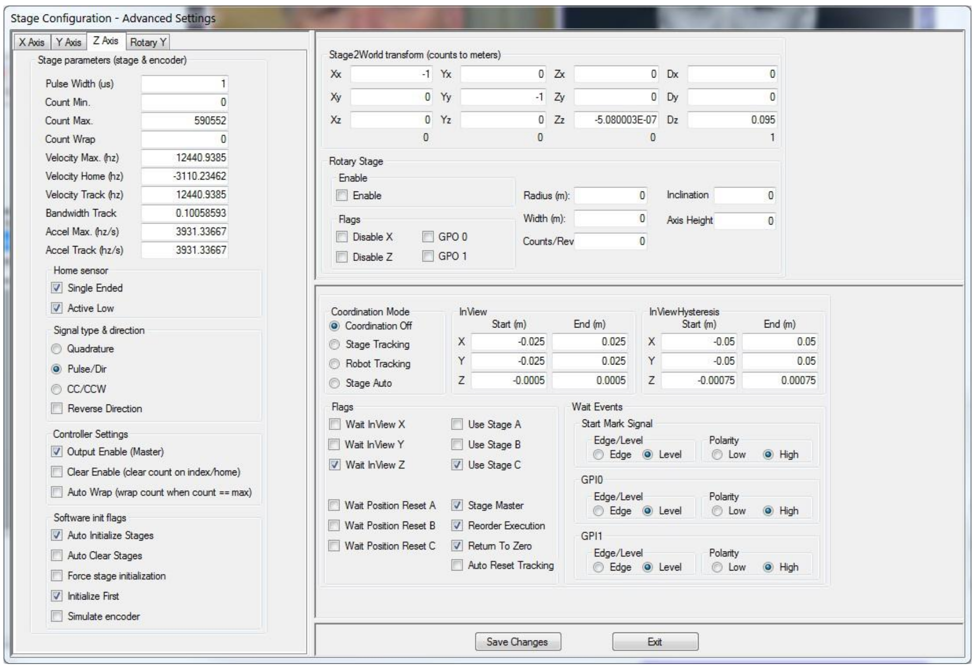
5. Attach any additional documents that may be useful to IPG for debugging purposes by clicking the “Add Attachment” button (i.e. videos or pictures of the issue occurring).
6. Once all fields have been populated, click “Send Report” or “Send and Save Report.” Please note, it can take a couple of minutes for the report to send.
    a. The report can also be saved to a zip folder by clicking “Save Report.” The report will not be sent to IPG when this is done. It is then up to the user to send the report to the proper IPG personnel.
7. Click “Ok” to confirm the report sent successfully.

# 20 Appendix - Stage Configuration Utility

In order to configure a scanner for use with motors/encoders, please refer to the following procedure.

1. Open the Stage Configuration Utility.
2. Lock onto the desired scanner.
3. Once the scanner is connected, click on "File" → "Advanced Settings". This willopen the Advanced Configuration Window, Figure 20-1. Below highlights the important parts of this window.

*Figure 20-1 Advanced Configuration Window*




a. **Stage2World Transform**: defines the transformation matrix for thesoftware. For On-The-Fly the only parameters that should not be zero are Xx, Yy and Zz. The three non-zero values are the meters per encoder count.

b. **Rotary Stage**: This is not applicable for encoder applications. All fields should be left as shown in Figure 20-1.

c. **Stage Parameters:**

   i. **Pulse Width**: not applicable for encoder input.

   ii. **Count Min/Max**: defines the process area in pulse units.

   iii. **Count Wrap**: Distance in encoder pulses before the software "Wraps" the weld.

   iv. **Velocity Max**: not applicable for encoder input.

   v. **Velocity Home**: when using the simulate encoder, Velocity Home defines the encoder velocity in pulses/sec.

   vi. **Velocity Track**: not applicable for encoder input.

   vii. **Accel Max / Accel Track**: not applicable for encoder input.

   viii. **Home Sensor**: If the encoder reset signal is single ended, SingleEnded flag should be checked, otherwise encoder reset is assumed differential. Active Low: if single ended, this flag indicated if the sensor is activelow.

ix. **Signal Type & Direction**: Encoder input works in quadrature mode by definitions so Quadrature must be selected. Reverse direction is not applicable for encoder input.

x. **Controller Settings**: Output enable flag must be off so the controller will read encoder signals. Clear Enable is optional; as it clears positioncount when the encoder reset pulse is detected. Auto Wrap is optional; as it wraps the position count if the encoder pulse count is equal to Count Max.

xi. **Software Init Flags**: The only applicable flag is Simulate encoder which ensures that a simulated encoder signal (count settings taken from Stage2WorldTransform) is initiated when the software starts. Other flags not applicable for encoder input.

d. **Coordination Mode:**

i. **Coordination Off**: When checked, coordination with external motionis disabled.

ii. **Stage Tracking**: When checked, enabled coordination with external motion control. This setting must be selected when using On-The-Fly.

iii. **Robot Tracking**: used with robot motion. Not applicable for encoder input.

e. **InView & InViewHysteresis:** Controls the size of the volume in whichthe scanner mirrors are allowed to move when Stage Tracking isenabled.

**IMPORTANT**

**For most 2D applications, InView X and Y should be set from -1/4 of the lens's optical field to +1/4 of the lens's optical field. The InViewHysteresis should be set to -1/2 of the lens's optical field to +1/2 of the lens's optical field. For the Z setting, the same rule applies, however the focal depth (not the lens's optical field) is used.**

f. **Flags:**

i. **Wait in View & Use Stage**: enables or disabled each axis for encoder input. For most applications Wait in View should be checked anytime Use Stage is checked.

ii. **Wait Position Reset**: for a moving target, waits until mirrors resettheir position before outputting the next vector. Must be on for On-The-Fly.

iii. **Stage Master**: informs the software whether to control external motion or not. Must be off for On-The-Fly since an encoder is controlling the soft- ware (software must be controlled externally).

iv. **Split Long Vectors**: if enabled, vectors that are longer than the area specified by the InView, will be split into multiple vectors that will fit the InView window. Must be checked for On-The-Fly.

v. **Return to Zero**: Not applicable for On-The-Fly.

vi. **Auto Reset Tracking**: automatically resets encoder tracking whenlaser is enabled. Should be off when using IPGScan.

g. **Wait Events**: are not related to On-The-Fly specifically, but are included for convenience. Wait Event Flags are used to set up the behavior of the general I/O's for the scanner controller, including external start.

h. **Polarity**: defines if the signal is active high or active low. This system is active high by default.

i. **Edge/Level**: defines how the signal becomes active. Default is Level, where signal is active anytime input voltage exceeds threshold level. Edge option is where the signal becomes active anytime a transition from low to high is detected.

**IMPORTANT** **Threshold is determined by the External Interface being used. See the Scanner Series User Manual and External Interface Board User Guide for more information.**